

MusicXML Analyzer

Ein Analysewerkzeug für die computergestützte
Identifikation von Melodie-Patterns

*Manuel Burghardt, Lukas Lamm, David Lechler,
Matthias Schneider und Tobias Semmelmann*

Universität Regensburg

Institut für Information und Medien, Sprache und Kultur

Ansprechpartner: manuel.burghardt@ur.de

Zusammenfassung

Der Beitrag beschreibt Aktivitäten aus einem aktuellen Digital Humanities-Projekt, das im Schnittfeld von Informations-, Musik- und Kulturwissenschaft angesiedelt ist. Dabei soll eine Sammlung von ca. 50.000 handschriftlichen Liedblättern mit deutschsprachiger Volksmusik digitalisiert und maschinenlesbar in *MusicXML* kodiert werden, um schließlich über ein Informationssystem quantitative Analysen des Materials zu erlauben. Wir stellen einen ersten webbasierten Prototypen vor, der es erlaubt, Musikstücke im MusicXML-Format statistisch auszuwerten und nach konkreten Melodie-Patterns zu suchen. Das Tool ist zudem in der Lage, virtuelle Partituren und Audioausgaben auf Basis des MusicXML-Markups zu erstellen und direkt im Webbrowser verfügbar zu machen.

Abstract

This article describes a recent digital humanities project by a group of researchers from various disciplines, including musicology, cultural studies, and information science. The project is aiming to digitize a collection of approx. 50,000 handwritten sheets of German folk music. In addition, the songs will be encoded in *MusicXML* format, which makes the data accessible for quantitative analyses. In this article we describe a web-based prototype that can be used to analyze any MusicXML encoded song with regard to quantitative properties (e.g. most frequent notes and intervals), but that also allows researchers to query a corpus of songs for specific melodic patterns. The tool is also designed to render virtual scores and create audio output from the MusicXML markup directly in a user's web browser.

Einleitung

Konzepte wie *distant reading* (Moretti, 2007) und *macroanalysis* (Jockers, 2013) beschreiben Ansätze in der Literaturwissenschaft, die es erlauben, auf quantitativer Ebene neue Erkenntnisse zu größeren Zusammenhängen zwischen einzelnen Texten und Autoren zu erlangen. Dabei sollen generische Muster identifizierbar gemacht werden, welche durch *close reading* einzelner Texte nicht unmittelbar sichtbar sind. Es scheint somit naheliegend, quantitative Verfahren, welche größere Datensammlungen zunächst aus der *Distanz* analysieren, auch auf andere Medientypen, wie etwa Audiodaten zu übertragen (für eine beispielhafte Studie vgl. etwa Viglianti, 2007), um so durch *distant hearing* besonders markante oder häufig wiederkehrende Melodie-Patterns in großen Liedkorpora zu entdecken.

Music Information Retrieval

Das Indexieren und Durchsuchen von Audiodaten fällt gemeinhin in das Feld des *Music Information Retrieval* (MIR). Bestehende MIR-Tools lassen sich einerseits in Tools für die Verarbeitung von Audiodaten (akustische Aufnahmen) und andererseits in Tools für die Verarbeitung von notierter Musik (Transkriptionen / Notenblätter) klassifizieren (vgl. Typke, Wiering, & Veltkamp, 2005). Über einen Vergleich von 17 bestehenden Tools identifizieren Typke et al., (2005) die folgenden zentralen Komponenten für MIR-Systeme, die jeweils auf unterschiedliche Arten umgesetzt werden können:

- *Input*: Formulierung einer Query, entweder im Audioformat (vgl. etwa „Query by Humming“, Ghias et al., 1995) oder als formalisierte, maschinenlesbare Suchanfrage
- *Matching*: Abgleich der Query mit der Musikdatenbank (exakt vs. angenähert, monophon vs. polyphon)
- *Features*: Parameter, welche bei der Formulierung der Query gesetzt werden können (Tondauer, Tonhöhe, Intervalle, u.a.)

Konkrete Ansätze zur quantitativen Analyse von Musikdaten mithilfe von MIR-Systemen finden sich zahlreich in der Literatur, und werden dort meist unter dem Gesichtspunkt der *melodic similarity* beschrieben (Grachten, Arcos, & de Mántaras, 2002; Grachten, Arcos, & Mántaras, 2004; Miura & Shioya, 2003; Müllensiefen & Frieler, 2004a, 2004b).

Ziele

In diesem Beitrag präsentieren wir ein prototypisches Tool das es erlaubt, beliebige Musikdaten, die im standardisierten *MusicXML*-Format¹ vorliegen, zu analysieren und nach Melodie-Patterns zu durchsuchen. Durch seine Flexibilität hinsichtlich der analysierbaren Daten füllt das Tool damit zum einen eine Lücke in der Landschaft bestehender MIR-Tools, die häufig für eine abgeschlossene Sammlung von Musik konzipiert sind (vgl. Typke et al., 2005). Zum anderen liefert das Tool einen ersten Prototyp für die Analyse von mehreren tausend handschriftlichen Liedblättern, die in einem aktuell laufenden Projekt digitalisiert und im MusicXML-Format kodiert werden. Gleichzeitig illustriert das Tool die Vorteile einer standardisierten, XML-basierten Kodierung von Musikdaten, welche es nicht nur erlaubt die Daten automatisch auswertbar und durchsuchbar zu machen, sondern im Umfeld webbasierter Technologien zudem eine grafische Ausgabe in Form virtueller Partituren sowie eine akustische Wiedergabe der Daten in einer integrierten Analyseumgebung ermöglicht.

Projektkontext: Regensburger Volksmusik-Portal

Das in diesem Beitrag präsentierte Analysewerkzeug ist im Kontext eines laufenden Digital Humanities-Projekts, im Schnittfeld von Informationswissenschaft, Musikwissenschaft und Kulturwissenschaften, angesiedelt. Wesentliche Ziele dieses interdisziplinären Projekts sind die Digitalisierung, sowie auch die computergestützte Analyse eines Korpus mit über 50.000 Liedblättern, die zum Bestand des Regensburger Volksmusik-Portals (RVP)² zählen. Die einzelnen Liedblätter sind jeweils handschriftlich verfasste Dokumente, welche sowohl die monophonen Melodien als auch die Liedtexte zu Volksliedern aus dem deutschsprachigen Raum im 19. – 20. Jhd. enthalten (vgl. Abbildung 1). In einer vorhergehenden Projektphase wurden die Metadaten (Sangesort, Incipet, etc.) des Bestands bereits größtenteils in ein digitales Archivsystem überführt.

¹ Mehr Informationen zum MusicXML-Standard finden sich auf der offiziellen Webseite <http://www.musicxml.com/>. Hinweis: Sämtliche URLs die in diesem Beitrag erwähnt werden wurden zuletzt am 12.6.2015 auf Verfügbarkeit geprüft.

² <http://www.uni-regensburg.de/bibliothek/projekte/rvp/index.html>

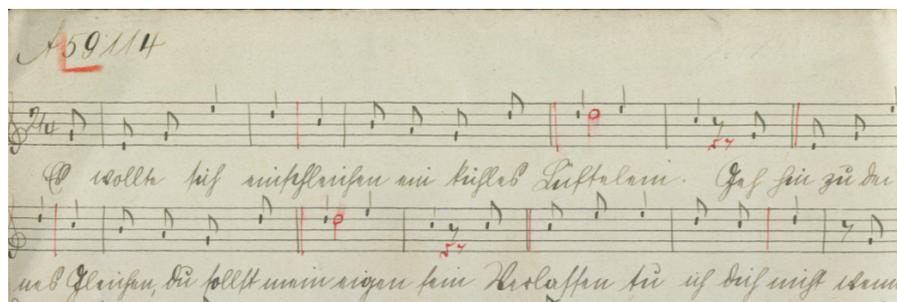


Abbildung 1: Teilausschnitt eines Liedblattes aus der RVP-Sammlung.

Im Rahmen des Projekts soll u.a. diese bestehende Liedblattsammlung digitalisiert und über ein Webportal öffentlich verfügbar gemacht werden. Dabei sollen die Daten nicht nur gescannt werden, sondern so repräsentiert werden, dass quantitative Analysen des Materials möglich werden, also z.B. die Identifikation typischer Melodie-Fragmente in bestimmten geografischen Bereichen oder zeitlichen Epochen. Aktuell wird geprüft, inwieweit mit automatischen *Optical Music Recognition* (OMR)-Verfahren (Bainbridge & Bell, 2001; Raphael & Wang, 2011; Rebelo, Capela, & Cardoso, 2010) die handschriftlichen Notenblätter in das etablierte MusicXML-Format überführt werden können. Da die ersten Evaluationsergebnisse zu solch automatischen Verfahren bislang wenig aussichtsreich für die vorliegende Datenbasis scheinen, wird alternativ angedacht, eine webbasierte Crowdsourcing-Plattform nach dem Vorbild bestehender Transkriptionsprojekte, wie etwa „What’s the Score“³, umzusetzen.

MusicXML Analyzer

MusicXML Analyzer unterstützt Kultur- und Musikwissenschaftler dabei, Musik einerseits statistisch auszuwerten und andererseits wiederkehrende Melodie-Patterns in unterschiedlichen Musikstücken aufzufinden. Vor dem Hintergrund der besonderen Anforderungen an die *Humanist-Computer Interaction* (Burghardt & Wolff, 2015) wird beim Interface-Design, d.h. beim Upload von Dokumenten, der Eingabe von Suchmustern sowie der Darstellung der Auswertung und der Ergebnisse, besonderes Augenmerk auf *Usability* und *User Experience* gelegt. Die Ergebnisse der Analysen werden direkt

³ <http://www.bodleian.ox.ac.uk/bodley/finding-resources/special/projects/whats-the-score>

im Browser visualisiert und können darüber hinaus auch als Audioausgabe abgespielt werden. Daneben ist aber auch der Export der Analysen als PDF oder als CSV-Datei (*Comma Separated Values*) zur weiteren Verarbeitung der Daten möglich.

Verfügbarkeit

Eine erste, funktionsfähige Version des *MusicXML Analyzer* ist unter folgender Adresse verfügbar, und kann direkt im Browser ausprobiert werden⁴:

- Demo: <http://music-xml-analyzer.herokuapp.com/>

Gleichzeitig ist das Tool mit allen Komponenten als *Open Source Software* auf dem Code Repository *GitHub* verfügbar und kann bei Bedarf beliebig angepasst und weiterentwickelt werden:

- GitHub Repository:
<https://github.com/freakimkaefig/Music-XML-Analyzer>

Um einen ersten Eindruck über die wesentlichen Funktionen des Tools zu gewinnen, wurde zudem ein kurzes Demonstrationsvideo erstellt:

- Video: <https://www.wevideo.com/view/417278189>

Funktionsübersicht

Die Funktionen des Systems lassen sich in drei Teilbereiche aufgliedern, die im Wesentlichen den typischen Analyse-Workflow widerspiegeln (vgl. Abbildung 2). Im ersten Schritt erfolgt der Upload von Dateien im MusicXML-Format, daraufhin folgt die automatische, statistische Auswertung. Den letzten Schritt stellt die Suche nach Mustern in den hochgeladenen Dateien dar.

⁴ Hinweis: Für die aktuell verwendete Serverumgebung gelten einige technische Einschränkungen: Bei längerer Inaktivität der Anwendung begibt sich die Laufzeitumgebung automatisch in einen Ruhemodus, welcher das erneute „Aufwecken“ der Anwendung erfordert. Das führt dazu, dass der Initialzugriff auf die Anwendung unter Umständen einige Sekunden dauern kann.

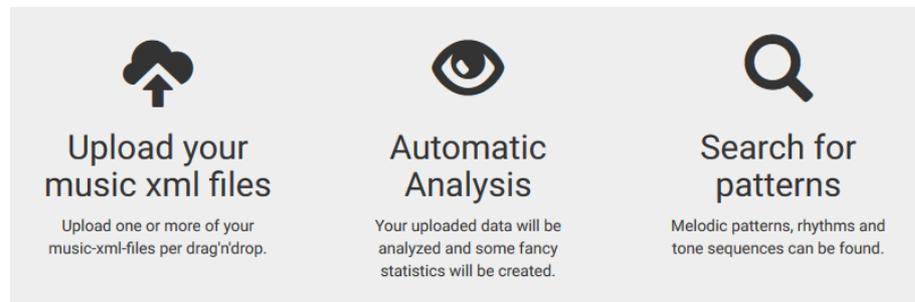


Abbildung 2: Startscreen der Anwendung, auf welchem der grundlegende Tool-Workflow (Datei-Upload > Analyse > Mustersuche) dargestellt wird.

Upload

Bevor die Analyse von Musikstücken beginnen kann, müssen zunächst Dateien im MusicXML-Format hochgeladen werden. Der Upload beliebig vieler MusicXML-Dateien kann über einen intuitiven Drag-and-Drop-Dialog bewerkstelligt werden. Die Anwendung gibt jeweils Rückmeldung über den aktuellen Upload-Status, da der Prozess bei mehreren Dateien und einer langsamen Internetverbindung einige Sekunden in Anspruch nehmen kann.

Analysekomponente

Nach erfolgreichem Upload erfolgt die Analyse der MusicXML-Dateien. Zum Parsing der MusicXML-Daten kommt eine Kombination aus XPath-Ausdrücken und PHP-Skripten zum Einsatz, die serverseitig ausgeführt werden und die Ergebnisse der Analyse in einer relationalen SQL-Datenbank speichern. Diese Parsing-Ergebnisse stellen einerseits die Grundlagen für die spätere Mustersuche dar, und werden andererseits (in Teilen) in einem Dashboard visualisiert (vgl. Abbildung 3).



Abbildung 3: Ausschnitt aus dem Analyse-Dashboard für ein Korpus aus MusicXML-kodierten Dokumenten.

Die folgenden Informationen sind über das Dashboard sowohl für das gesamte Korpus als auch für jeweils einzelne Lieder des Korpus verfügbar:

- Anzahl aller Einzelnoten, Pausen und Takte
- Verwendete Instrumente (sofern in MusicXML angegeben)
- Häufigkeitsverteilung von Notenwerten, Intervallen, Tonarten, Notenlängen und Takten

Alle im Dashboard dargestellten Analysen können zudem als CSV-Datei heruntergeladen werden.

Such-Komponente

Die Such-Komponente erlaubt es schließlich das Korpus aus MusicXML-Dokumenten nach bestimmten Melodie-Patterns zu durchsuchen. Dabei ist die Suche nach reinen Rhythmus-Patterns, reinen Tonabfolgen sowie einer Kombination aus Rhythmus- und Ton-Patterns möglich (vgl. Tabelle 1):

<i>Parameter</i>	<i>Beispielhafte Suchanfrage</i>
Tonabfolge-Pattern (<i>sound sequence</i>)	Finde alle Songs in denen die Notenfolge C - C - D - G (jeweils 4. Oktave) vorkommt, ungeachtet der Notenlänge.
Rhythmus-Pattern (<i>rhythm</i>)	Finde alle Songs in denen eine Sequenz aus folgenden Notenlängen vorkommt, ungeachtet der Tonhöhe: Achtelnote - Achtelnote - Sechzehntelnote - Sechzehntelnote.
Melodie-Pattern (<i>melody</i>), d.h. Kombination aus Tonwert und Rhythmus	Finde alle Songs in denen die Notenfolge C, Achtelnote - C, Achtelnote - D, Sechzehntelnote - G, Sechzehntelnote (jeweils 4. Oktave) vorkommt.

Tabelle 1: Suchparameter und beispielhafte Suchanfragen.

Die Eingabe der Suchmuster ist entweder durch Klicken mit der Maus über eine interaktive Notenzeile möglich, oder über die Auswahl von Noten- und Taktwerten über ein Menü (vgl. Abbildung 4). Die ausgewählten Melodie-Patterns werden jeweils direkt auf der Notenzeile dargestellt und können zudem im Browser abgespielt werden, um auch einen direkten, akustischen Eindruck der formulierten Suchanfrage zu erhalten.

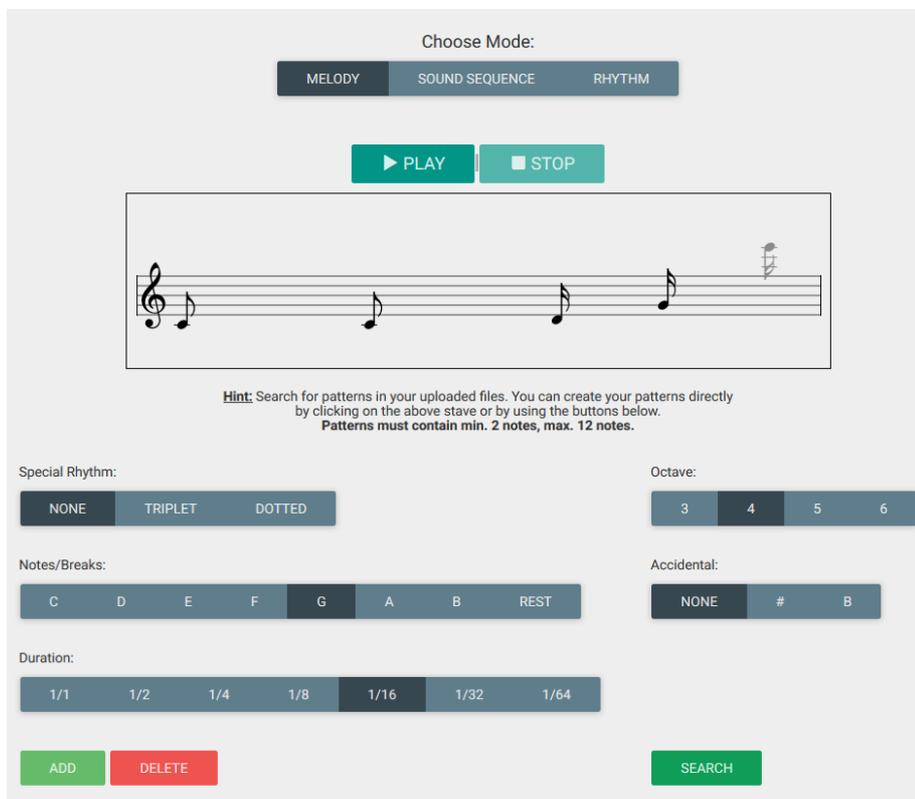


Abbildung 4: Interface für die Formulierung von Suchanfragen zur Identifikation von Tonfolge-, Rhythmus- oder Melodie-Patterns.

Nach dem Abschicken der Suchanfrage werden in einer Ergebnisliste all diejenigen Lieder angezeigt, in denen das Such-Pattern vorkommt, zusammen mit der jeweiligen Häufigkeit des Vorkommens. Mit einem Klick auf eines der aufgelisteten Lieder wird automatisch die entsprechende Partitur anhand der MusicXML-Daten im Browser rekonstruiert und die gesuchte Sequenz darin farbig hervorgehoben. Auch hier ist es wiederum möglich, die gesamte Partitur im Browser abzuspielen oder diese als PDF herunterzuladen (vgl. Abbildung 5).

Lee Actor (2003) - Prelude to a Tragedy
(ActorPreludeSample.xml)

▶ PLAY

■ STOP

About the finding:

Part name (Instrument):
Clarinets in Bb

Part ID: P5

Voice: 1

Key: C major

Measures: 35 - 38

Abbildung 5: Virtuelle Partitur-Darstellung des Trefferdokuments und farbige Hervorhebung der gesuchten Notensequenz.

Implementierung

Programmbibliotheken und Webtechnologien

MusicXML Analyzer wurde ausschließlich mithilfe von Webtechnologien wie HTML, CSS, JavaScript (JS) und PHP umgesetzt. Für die Implementierung der einzelnen Komponenten wurde, soweit möglich, auf bestehende Frameworks und Programmbibliotheken zurückgegriffen (vgl. Tabelle 2).

Funktion	Name	URL
Allgemeines PHP Framework	<i>Laravel</i>	http://laravel.com/
Allgemeines JavaScript Framework	<i>jQuery</i>	https://jquery.com/
Allgemeines CSS Framework	<i>Bootstrap</i>	http://getbootstrap.com/
JS-Bibliothek für die Visualisierungen im Dashboard	<i>D3.js</i>	http://d3js.org/

JS-Bibliothek für die Erzeugung der virtuellen Partituren	<i>Vexflow</i>	http://www.vexflow.com/
JS-Bibliothek für die Erzeugung der Audioausgabe	<i>Midi.js</i>	http://mudcu.be/midi-js/
JS-Bibliothek für die Visualisierung von Statusnachrichten	<i>Typed.js</i>	http://www.mattdboldt.com/demos/typed-js/
JS-Bibliothek für den Upload von Dateien	<i>Dropzone.js</i>	http://www.dropzonejs.com/
JS-Bibliothek für den Export von PDFs	<i>jsPDF</i>	https://parall.ax/products/jspdf

Tabelle 2: Überblick zu den verwendeten Frameworks und Programmbibliotheken.

Systemarchitektur und Implementierungsdetails

Abbildung 6 zeigt den Aufbau des verwendeten Laravel-Frameworks im Zusammenspiel mit den jeweiligen JavaScript-Komponenten. Die Laravel-Komponente kümmert sich dabei um die serverseitige Logik der Anwendung, wie etwa die Auslieferung von HTML-Seiten über eine integrierte *templating engine* oder die Kommunikation mit der Datenbank zur persistenten Speicherung. In Abbildung 6 wird auch der Aufbau nach dem Model-View-Controller-Pattern (MVC) deutlich, demzufolge die grafische Repräsentation (*view*) von der Logik (*controller*) und der Datenschicht (*model*) getrennt ist. Die JavaScript-Komponente der Anwendung ist ebenso nach dem MVC-Pattern aufgebaut, und kümmert sich hauptsächlich um die Interaktivität der Anwendung. Beispielsweise werden die Suchmuster in JavaScript zunächst clientseitig zwischengespeichert und erst beim Absenden der Suchanfrage an den Server geleitet. Weitere Aufgabengebiete sind die Visualisierung der Analysedaten und Partitur-Ausschnitte, sowie die Audio-Wiedergabe einzelner Sequenzen.

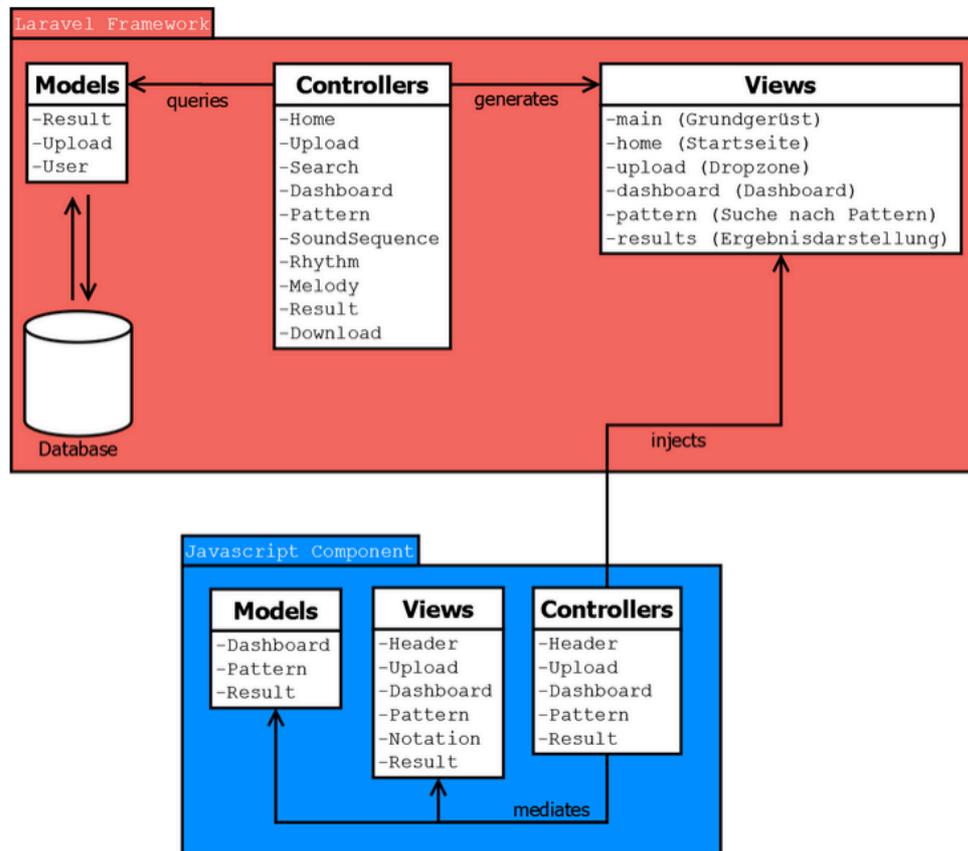


Abbildung 6: Überblick zur Systemarchitektur auf Basis des PHP Laravel-Frameworks.

Datenspeicherung und Benutzerverwaltung

Bei der Konzeption wurde bewusst auf eine aufwendige Nutzerverwaltung mit Registrierung und Anmeldung verzichtet. Um dem Nutzer die Möglichkeit zu bieten, seine Arbeit ohne zusätzliche Speicherung immer aktuell zu halten werden alle Analysen automatisch in einer Datenbank gespeichert. Den Nutzern wird bei der ersten Verwendung des Tools automatisch eine ID zugewiesen, die gleichzeitig als Cookie im Browser hinterlegt wird. Auf den Server geladene Dateien werden entsprechend mit dem Nutzer referenziert. Nach dem gleichen Schema werden einem Upload-Objekt nach erfolgreicher Analyse dessen Ergebnisse zugeordnet.

Ausblick

MusicXML Analyzer hat aktuell den Status eines voll funktionsfähigen Prototypen, der in dieser Form direkt in der verfügbaren Liveversion oder als lokale Webapplikation für eigene Musikanalysen verwendet werden kann. Gleichzeitig wird die Anwendung während des eingangs beschriebenen Projekts stetig weiterentwickelt. Parallel dazu sind Weiterentwicklungen und Anpassungen des Tools durch andere Entwickler jederzeit möglich, da das Tool auf *GitHub* unter der freien *MIT Lizenz* als *Open Source Software* veröffentlicht wurde.

Ein unmittelbar nächster Schritt für die Erweiterung von *MusicXML Analyzer* im Rahmen des laufenden Volksmusikprojekts ist die Verknüpfung der Datenbank mit den bereits digital vorliegenden Metadaten (Ort, Datum, u.a.), sodass etwa eine gezielte Suche nach Melodie-Patterns in bestimmten Regionen oder Zeiträumen möglich ist. Während im vorliegenden Prototyp zunächst die explizite Suche nach Patterns im Vordergrund steht, so soll parallel auch eine Analysekomponente erstellt werden, die in den MusicXML-kodierten Daten automatisch melodische Ähnlichkeiten und Muster erkennt (vgl. hierzu auch die eingangs zitierten Studien im Bereich *melodic similarity*). Neben der reinen Analyse von Melodie-Patterns ist auf lange Sicht zudem die Kombination mit den digitalisierten Liedtexten ein wichtiges Desideratum, welches es beispielsweise erlaubt zu untersuchen ob bestimmte Wörter mit bestimmten melodischen Charakteristika korrelieren. Eine beispielhafte Fragestellung könnte etwa folgendermaßen lauten: Kommt das Wort „Krieg“ wegen der negativen Konnotation häufiger in Kombination mit einer Molltonart als mit einer Durtonart vor?

Literatur

Bainbridge, D., & Bell, T. (2001). The challenge of optical music recognition. *Computers and the Humanities* (35), 95–121.

Burghardt, M., & Wolff, C. (2015). Humanist-Computer Interaction: Herausforderungen für die Digital Humanities aus Perspektive der Medieninformatik. *Book of Abstracts Workshop "Informatik und die Digital Humanities"*, Leipzig.

- Grachten, M., Arcos, J. L., & de Mántaras, R. L. (2002). A comparison of different approaches to melodic similarity. *Proceedings of the 2nd ICMAI*, p. 1-12.
- Grachten, M., Arcos, J. L., & Mántaras, R. L. De. (2004). Melodic Similarity: Looking for a Good Abstraction Level. *Proceedings of the 5th ISMIR*, p. 210-215.
- Ghias, A., Logan, J., Chamberlin, D., & Smith, B.C. (1995). Query By Humming – Musical Information Retrieval in an Audio Database. *Electronic Proceedings of the ACM Multimedia 95*.
- Jockers, M. L. (2013). *Macroanalysis: Digital Methods and Literary History (Topics in the Digital Humanities)*. University of Illinois Press.
- Miura, T., & Shioya, I. (2003). Similarity among melodies for music information retrieval. *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM '03)*, p. 61-68.
- Moretti, F. (2007). *Graphs, Maps, Trees: Abstract Models for Literary History*. London: Verso.
- Müllensiefen, D., & Frieler, K. (2004a). Melodic Similarity: Approaches and Applications. *Proceedings of the 8th International Conference on Music Perception & Cognition*, p. 283–289.
- Müllensiefen, D., & Frieler, K. (2004b). Optimizing Measures Of Melodic Similarity For The Exploration Of A Large Folk Song Database. *Proceedings of the 5th ISMIR*, p. 274–280.
- Raphael, C., & Wang, J. (2011). New Approaches to Optical Music Recognition. *Proceedings of the 12th ISMIR*, p. 305–310.
- Rebelo, a., Capela, G., & Cardoso, J. S. (2010). Optical recognition of music symbols. *International Journal on Document Analysis and Recognition (13)*, 19–31.
- Typke, R., Wiering, F., & Veltkamp, R. C. (2005). A survey of music information retrieval systems. *Proceedings of the 6th ISMIR*, p. 153–160.
- Viglianti, R. (2007). MusicXML: An XML Based Approach to Musicological Analysis. *Proceedings of the 18th Digital Humanities conference*, p. 235–237.