Functional Specification

Music Analysis Tool

Patrick Ferry

Computer Applications & Software Engineering
Dublin City University
30/11/2018

Contents

1 Introduction

1.1 Overview

The system to be designed shall be a music analysis tool. It will take a song as an input and then output details about the song. The file type the system shall prefer will be a WAV and the system shall only take instrumental songs. The details that are returned will come at different levels beginning with simpler information such as the Beats per minute (BPM), key signature, and form. This will be done by trying to find all of the notes by analysing the soundwave. It will then try to differentiate between instruments that are playing in the song using the textures of the sounds. The system will try and identify any sounds that are drum hits and these sounds will be passed to machine learning model that is trained to try identify different drums. The system shall then return a notated version of the song, for each instrument in the form of sheet music, or alternatively a Musical Instrument Digital Interface file (MIDI) file.

The back-end of the system shall be hosted on a server and will be accessible to use through a basic web application front-end. The front-end shall be a simple user interface and shall just be a means for the user to upload the file and select any options. The part of the system that is used for basic analysis shall also be available to download as a standalone that you can run from command line.

1.2 Business Context

The system will possibly be used in the education industry, specifically for music theory. It could be used as a tool for musicians who are looking to learn new songs for which they cannot find any sheet music for online. Music theorists may also be able to use the system to help with their studies, as they will not have to worry about transcribing the song into sheet music.

Producers will be able to avail of this tool in the form of education also. It will help a beginner producer study songs that they enjoy and would like to recreate, as this a common way to improve as a producer. Producers with more experience would be able to use the tool to extract parts of the song that they like e.g. drum beat or melody. The producer would then be able to interpolate these extracts into their own songs.

This educational use of the software could be marketed through a subscription to the web application, or possibly as a once off payment. A free trial could be offered before users pay. Another possible use for the system may be in data science. The system would be able to act on a library of songs to find their notes and info. This could then discover new things about songs and the similarities or differences between them and be useful for recommendation systems.

1.3 Glossary

back-end the data-access of a software.

data science an interdisciplinary field that uses scientific methods processes algorithms and systems to extract knowledge and insights from data.

DAW A digital audio workstation is an application software used for editing and producing audio files..

discrete Fourier Transform function that can be used to turn a sound wave over time into the frequencies that make it up.

form Structure of a musical compostition.

frequency the number of a soundwave completes it's cycle; from this we can derive the note.

front-end the presentation layer of a software.

interpolate in music generally refers to a music track reproducing a portion of another song in some other way.

machine learning is an application of artificial intelligence..

manuscript paper The paper with the five line staffs for notating music.

metadata data that gives information on other data.

overfitting when a model trained too closely to a particular set of data.

overtones any frequency greater than the fundamental frequency of a sound..

percussion An instrument is an instrument that is sounded by being struck.

polyphonic A song with multiple melodies playing at the same time.

producer someone who oversees and manages the creation of song.

timbre The perceived sound quality or tone color of a note or sound.

WAV Wave file type; commonly used for storing audio bitstreams.

Acronyms

BPM Beats per minute.

CSV comma-separated values file.

MIDI Musical Instrument Digital Interface file.

2 General Description

2.1 Product/System Functions

The system shall take a song as an input, chosen by the user, in the format of a WAV file. The system will take the song and first analyse the soundwave to find some of the notes and the rhythm. This shall be enough to determine the BPM and key signature. This information will be returned to the user. After this point the system shall run a version of onset detection and will mark any points in the song that a note or beat maybe occur. Onset detection is the technique of trying to identify the beginning of a note or sound.

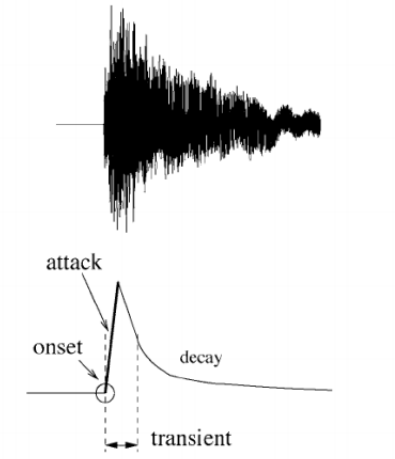

Figure 1: Onset detection example

This marked version of the song will then be analysed to look at the timbre of any of the marked notes. This will help differentiate between various instruments or noises. The system will use the help of overtones and other aspects of the sound when inspecting the timbre. Once the system has identified how many instruments it thinks the song has it will split the song into a separate version of the song or track for each of these. Any sounds thought to be drums will be marked again and put into another track solely for drums. The tracks that contain melodic or harmonic instruments will be filtered down to the range of the particular sound. Each note in these tracks will then be analysed using discrete Fourier Transform functions. These functions allow the system pull a frequency apart into the smaller frequencies that make it up.

The system will have a machine learning model for classifying drums. This will be trained on tagged samples of different types of drums. These samples will range from the sound of a type of percussion on it's own to sounds with other noise or songs in the background. The drum track that was created earlier will then chopped into each separate drum hit and our model will try to identify each drum hit and these will be marked. All of the tracks will then be passed to a module of the system that is intended to either output all of the tracks on sheet music or as a MIDI file.

2.2 User Characteristics and Objectives

Users would be expected to have some knowledge of music theory although it should not be neccessary, it would be able to help for more complex operations. The user just needs to select a song they would like to analyse then, using the web app interface or command line, select the file for that song. They will then select some arguments for how many instruments they want to analyse. They will then be returned with download links or files containing the MIDI or PDF file(s) that contains all the information for whichever instruments they selected.

2.3 Operational Scenarios

2.3.1 User wants sheet music for piano piece

User Objective & Description

The user hears a piano piece that they like online. The user wants to know how to play this piece but cannot find any sheet music for it online.

Primary Actor

The main actor in this scenario is an adult amateur pianist.

User Action

The user downloads the piece that they like and makes sure it is in WAV format. They then go to the webapp and drag and drop the file into the interface. The user then selects that they would like the song returned to them as sheet music, in the settings displayed on the interface. The user knows that there is just one instrument in the piece so they also so they let the tool know that through the interface and specify piano. The user clicks quick scan. The tool returns some basic information about the track to the user. The user clicks the analyse button and after a few moments a download link appears to them. The user downloads the file.

Comments

The user in this situation was able to identify that there was only one instrument in the piece, and by letting the tool know this, the user helped speed up the

process. If the user had been wrong and there was multiple instruments the user would have received inaccurate or cluttered sheet music back.

2.3.2 User wants metadata for collection of songs

User Objective & Description

The user is looking to remix some songs that they have in their music library. They do not want to have to transpose the tracks or change the tempo they have and therefore are looking for songs in the same key signature and BPM. They will need the metadata of the songs.

Primary Actor

Amateur DJ, with some experience with the tool and command line interfaces

User Action

The user will download the basic analysis part of the tool to be used on the command line. They will add it to the bin on their machine and run it from the command line. They locate the folder containing all the songs that they would like to analyse and run the tool on all of them. The tool will output a comma-separated values file (CSV) file containing all the information the tracks. They will then be able to filter this to find the tracks that match in key signature and BPM

Comments

This would be a more awkward operation and was mainly made possible because the user had some experience using their operating system and the tool.

2.3.3 User wants to use song's drum beat but with different samples

User Objective & Description

The user hears a drum beat in a song that they like but they are unsure of how to recreate it.

Primary Actor

A music producer

User Action

The user downloads the song containing the drum beat that they would like to recreate. They convert the file to WAV format. The user uploads the file using the web interface and selects the option for a MIDI file. They then select that they would only like the drums. The user clicks analyse and a download link appears. The user downloads the MIDI file and opens it in their DAW.

Comments

The user will have to have a software capable of opening MIDI files.

2.4 Constraints

1. It will be harder to analyse songs with more complexity e.g. polyphonic, too many instruments. This will have an effect on how accurate the results will be because if the song is polyphonic with two very similar instruments it may be picked up as just one single instrument. The plan will be to build up the tool in stages of handling complexity starting with just a melody, then a melody and eventually work up to a song with a melody, bass line and drums.

2. The quality of the audio file will also have an effect on how accurate the result will be. Inputs may have to be mastered to a certain standard to ensure better results.

3. The tool will not be built for many people to use it at once as it will be hosted on a small enough server for the purpose of this project.

4. Copyright may be an issue if the tool is used for anything more than education.

3 Functional Requirements

3.1 Requirements

3.1.1 Information returned to the user

Description

Information about the song is returned to the user. This will depend on the options chosen by the user. As a default it shall be a small output on the interface for the basic information and then a download link for the desired notation format will be displayed to the user.

Criticality

This is very important as without this the user will receive no information or feedback on the file they have chosen meaning the tool basically has zero functionality.

Technical Issues

Any file that is to be downloaded should be named uniquely to insure the right file is provided. The input from the options on the interface will all have to be compatible.

Dependencies

This will depend on the song being successfully notated into the files that are to be downloaded.

3.1.2 Sheet Music Notation

Description

This will be when the system will create the file that user can download. This shall output the sheet music in the form of pdf.

Criticality

This is very important because this is how the system will display the results to the user and if the system cannot correctly transcribe the notes for the users then it will be deemed useless.

Technical Issues

If the notation isn't split up correctly into the different instruments then the file will be close to illegible. It may be difficult to identify certain musical features that are usually notated in sheet music e.g. triplets.

Dependencies

This will heavily depend on the system having the correct information about the musical notes, their lengths, and the BPM of the song. The tool shall use a library to help create the manuscript paper format and put the correct musical notes on it.

3.1.3 MIDI File Creation

Description

This will be when the system will create the file that user can download. This shall output the music in the form of MIDI files.

Criticality

This is important as it what the user sees but it will also be very helpful in the case of testing of the system.

Technical Issues

If the song isn't split into the right number of tracks then a lot of files may be generated unneccessarily.

Dependencies

This, once again, will heavily depend on the system having the correct information about the musical notes, their lengths, and the BPM of the song. The tool shall use a library to help create the MIDI files.

3.1.4 Getting musical notes

Description

This will be when the system takes all the tracks and identifies the musical notes using discrete Fourier Transform functions.

Criticality

This is very important because this is how the tool will find the pitch and length of the musical notes.

Technical Issues

The system will need to know how long the notes, to improve the quality of the output, and this may be awkward to implement.

Dependencies

This will depend on the tracks being filtered correctly and that only the main notes have been marked by the timbre detector.

3.1.5 Identifying Percussion

Description

The tool will try and identify what type of drums are being hit using machine learning.

Criticality

The tonality of a percussion hit will not be as important as the type of hit it is, as that is how the percussion will notated.

Technical Issues

Training the model correctly to identify drums with other noise possibly happening.

Dependencies

This will depend on the tracks being split and correctly identified earlier in the pipeline.

3.1.6 Filtering

Description

The tool will filter down each track to try and keep only the needed musical notes or percussion hits.

Criticality

This will greatly help the discrete Fourier Transform functions identify the correct notes and will therefore increase the accuracy.

Technical Issues

There may be some issue filtering down the tracks correctly if there is multiple tracks in the same range.

Dependencies

This will depend on the timbres of different sounds being identified. It would also be helpful for this if there wasn't too many instruments contained within the same range, this would make it more effective.

3.1.7 Timbre Detection

Description

This will be used to try mark notes per possible instrument in a song and then using their timbre and split them into separate tracks.

Criticality

This how the system will know what how many tracks to split the song into. Without this the system would just output everything into one messy output.

Technical Issues

The timbres may be hard to find as their may be many sounds. Some filtering earlier will help fix that.

Dependencies

This depends on the clarity of the notes and on how well the on-set detection in the tool will be.

3.1.8 On Set Detection

Description

Onset detection is the detection of the beginning of a sound or note.

Criticality

This is how all of the notes or hits will be marked for the later functions to analyse so it will be critical in ensuring that we have found all of the notes.

Technical Issues

Onset detection may be difficult because it still an active research area and will take a lot of research to get correct. This will probably define how complex the songs that can be successfully analysed will be.

Dependencies

The clarity of the song requested to be analysed will be a big issue here.

3.1.9 Basic Analysis

Description

The song will be scanned over to return it's BPM and key signature. This will pick up on any musical phrases so the tool can learn the form of the song.

Criticality

The BPM is will be crucial to notating the music correctly into any of the possible outputs. The key and form may offer a heuristic technique to figuring out notes that cannot be heard clearly.

Technical Issues

The tool may have issues doing basic analysis on audio files with bad sound quality.

Dependencies

The input will have to be a valid song that our user selects.

3.1.10 User inputs a song and selects options

Description

User will select a song that they would like to analyse and will drag and drop it into the web application interface.

Criticality

This is how the tool will get it's input and is the beginning of the pipeline.

Technical Issues

Ensuring that interface is user friendly and not over complicated.

Dependencies

This depends on the user selecting the right kind of input.

3.1.11 Validating file

Description

This is when the system checks the file to make sure it won't cause any issues. e.g. too long, wrong format, etc.

Criticality

This will stop the system from crashing due to errors that should not be occurring if given the right output.

Technical Issues

It will need to be ensured that this will not allow any file that may cause a problem.

Dependencies

This relies on the file transfers working correctly.

3.1.12 Training classification model

Description

This is how the model will be taught to recognise different types of percussion e.g. snare, hi-hats, toms, kicks

Criticality

The tonality of the drums does not matter so much, but more so for their rhythm and the sound that they make. This model shall find this for us meaning it is crucial to the accuracy of the drums and it also allows them to be notated.

Technical Issues

It may be difficult to get the accuracy as high as the system may require. It may be difficult to find enough drum samples or snippets to train our model on

to get as wide of a range as possible. The scope of the drums may need to be limited.

Dependencies

This will heavily rely on a good, labeled dataset of drum samples. This will also rely being able to identify the main attributes in a sound wave for any of the samples.

3.1.13 Testing classification model

Description

This is when model will be tested by running it on a dataset that is labeled and measuring the accuracy.

Criticality

This will let us know if the model needs adjustments to improve the accuracy.

Technical Issues

It will need to be ensured that model has new data to test on so that it isn't being tested on previous training data and can therefore avoid overfitting.

Dependencies

This relies on having enough samples to train properly and still have some other data left to test on.

3.1.14 Testing notation accuracy

Description

This is when the whole system will be tested for it's accuracy. It will be used on songs, of different complexities, of which the notations have already been obtained, and it will test against these to find it's accuracy.

Criticality

This will be a test of the overall system and will be very helpful in the development and improvement of the system.

Technical Issues

A system will need to be looked into for comparing sheet music against other sheet music.

Dependencies

This relies on the whole system working correctly so that it may be tested. This relies on finding a diverse collection of songs that are already notated.

4 System Architecture

The tool will be hosted on a cloud server and will be accessible using the web application through a browser on a pc. The basic analysis part will be able to used offline as a standalone as it will not require any machine learning.

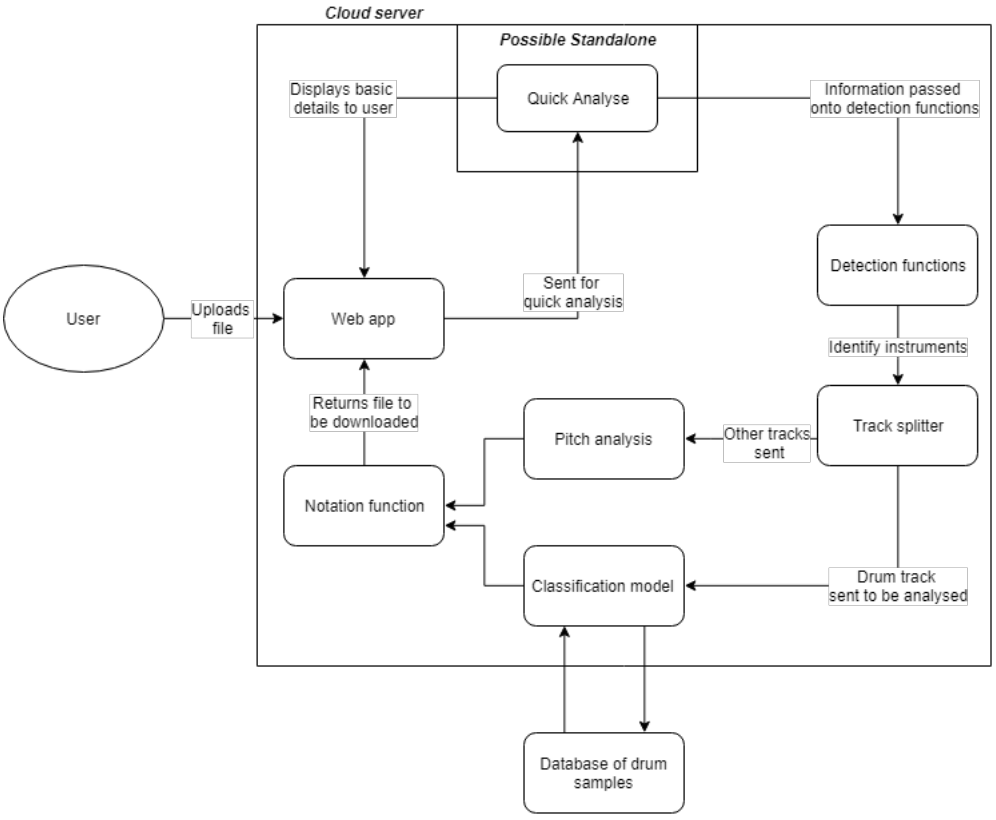

Figure 2: Basic diagram showing high level system architecture

5 High-Level Design

Below is a simple illustration of the system in the form of a context diagram. It shows the relationship between user, the web application, and the cloud server.

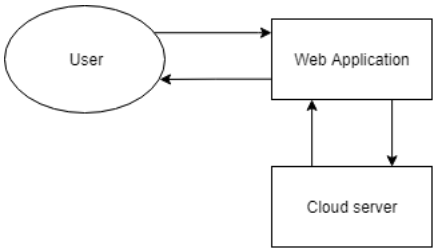

Figure 3: Context diagram

Below is a more in-depth diagram showing the relationship between different modules in the system.

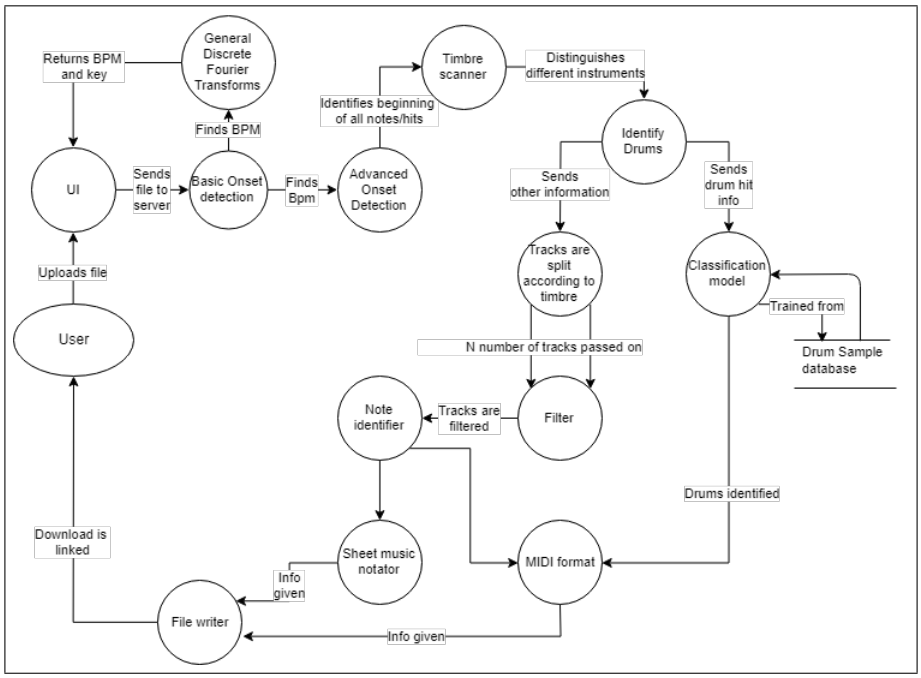

Figure 4: Data flow diagram

6 Preliminary Schedule

6.1 Deliverables and Gantt Chart

This project will be split into four deliverables

1. System will be able to correctly notate a song with one instrument
2. System will be able to correctly notate a song with instruments in different ranges e.g. lead and bass line
3. System will be able to correctly notate a song with polyphonic melodies and a bass line
4. System will be able to do all of the above and also correctly identify and notate percussion

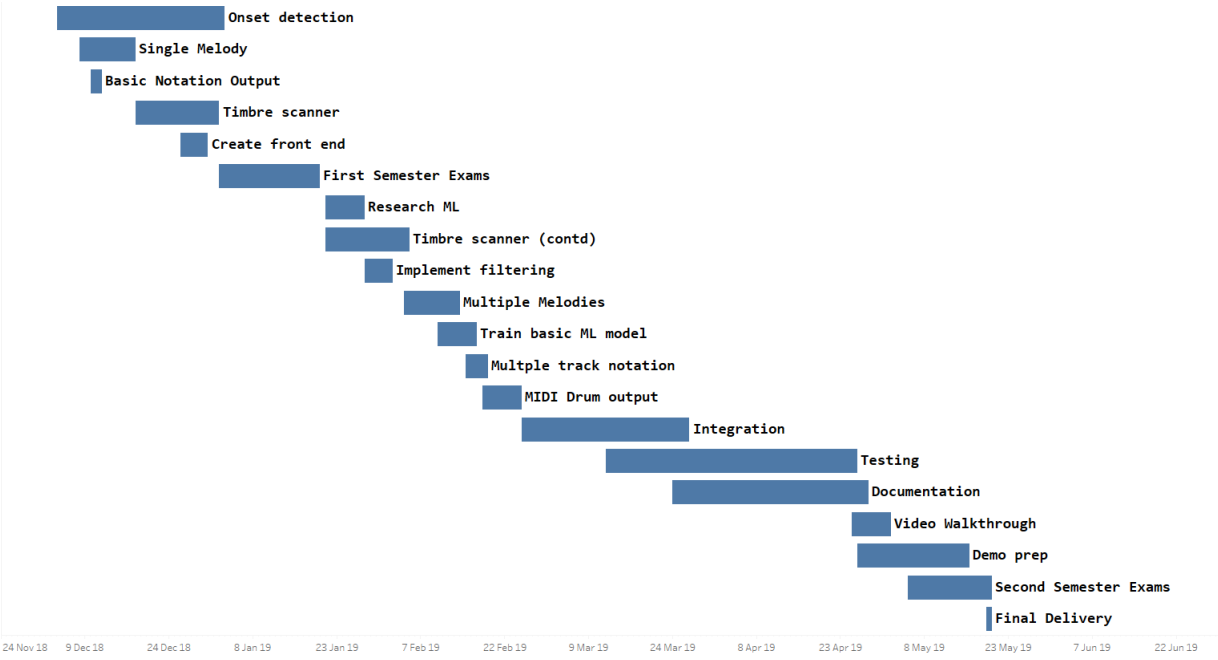

Figure 5: Gantt Chart

7 Appendices

Onset detection
http://www.iro.umontreal.ca/~pift6080/H09/documents/presentations/xavier_bello_tutorial.pdf

Overtones
http://www.phys.uconn.edu/~gibson/Notes/Section4_2/Sec4_2.htm

Classification
https://loop.dcu.ie/pluginfile.php/2291869/mod_resource/content/3/4-Classification-handouts.pdf