# Modulo7 : A full stack Music Information Retrieval and Querying Engine using Music Theoretic Principles

by

Arunav Sanyal

A thesis submitted to The Johns Hopkins University in conformity with the requirements for the degree of Master of Science.

Baltimore, Maryland

December, 2015

# Abstract

Music Information Retrieval (MIR) is an interdisciplinary science of extracting non trivial information and statistics from music data sources. In today's digital age, music is stored in a variety of digitized formats - e.g midi, musicxml, mp3, digitized sheet music etc. Music Information Retrieval Software aim at extracting features from one or more of these source. MIR research helps in solving problems like automatic music classification, recommendation engine design etc. Users can then query the acquired statistics to acquire relevant information.

The author proposes and implements a new Music Information Retrieval and Query Engine called Modulo7. Unlike other MIR software which deal with low level audio features, Modulo7 operates on the principles of music theory and a symbolic representation of music. Modulo7 is a full stack deployment, with server components that parse various sources of music data into its own efficient internal representation and a client component that allows consumers to query the system with sql like queries which satisfies certain music theory criteria (and as a consequence Modulo7 has a

ABSTRACT

custom relational algebra with its basic building blocks based on music theory).

Primary Reader: Dr David Yarowsky

Secondary Reader: Dr Yanif Ahmad

# Acknowledgments

I would like to thank Dr David Yarowsky for giving me the opportunity to work on this project. His detailed insights have immensely helpful to me to power through my work. I would like to thank Dr Yanif Ahmad for his crucial help in the systems aspects of my query engine and the implementation of the server side components.

# Dedication

This thesis is dedicated to my family and to all the music lovers in the world.

# Contents

# Chapter 1

# Introduction

Why does a person like a particular song? What are the inherent aspects of a song that pleases a person's musical taste? Is it the complexity of a song, the beat the song or just a particular melodic pattern ? More so if a person likes a song, can we predict if he/she will like a similar song?

Music has been created since the dawn of civilization and these questions have plagued mankind just as long. In response to this, man has created elaborate systems of formal study for music and classification techniques in almost every ethnic community since antiquity. Two notable examples are the western system of solfege and classical music theory and the Indian system of raagas. These elaborate systems are based on very simple fundamental building blocks of melody and harmony and simple rules that govern the interplay of these building blocks. However very complex pieces of

music can be created with these simple rules depending on the skill and virtuosity of artists. Composers use these rules and concepts to create novel music for mass consumption.

In the modern era industry and academia have attempted to address the problem of music recommendation and music classification. The industry has predominantly favored approaches that look at user preferences and history. For example Amazon Music recommendation works on users shopping history. Pandora on the other hand hires an army of musicologists to ascertain how a song is similar to another song and creates software that leverages this adhoc generated data. These approaches are either expensive in the human labor needed or in the amount of data processed that is input from a large number of users. More recently, companies like Echo Nest has extensively extracted features from music sources and mined cultural information on the web but leave it at consumers how best to leverage the data. Hence symbolic MIR is not traditionally used in industry and music theory is an after thought.

Academia on the other hand attempts to solve very particular problems in MIR. Typical examples would be cover song detection, processing information via signal processing, audio feature extraction, optical music recognition etc. In most cases the applications are of a very specific domain and does not fully scale with bulk music data. Generic frameworks like the jMIR[1] (which also happens to be a major inspi-

ration for Modulo7) suite for automatic music classification exists, which is meant to facilitate research in MIR with a machine learning focus. However academia is disconnected with industry and no full scale MIR engines can satisfy the scale of industry applications.

This work is attempt to bridge both communities. Modulo7 is a full stack deployment of Music Information Retrieval Software, providing both a server architecture and a sql like client to query based on music theory criteria. Modulo7 is a big data and information retrieval framework to explore the possibilities of exploring music theoretical aspects of music sources. Modulo7 does not attempt to solve very complex music theoretic problems (e.g study orchestral music to identify counter point information). Rather Modulo7 acts a framework on which such analysis can be built upon. Most importantly, Modulo7 addresses the issue of scale and allows a fast comparison between songs on certain music theoretic criteria. Modulo7 also acts to address deficiencies in existing software, such as filling up incompleteness in music sources. Certain problem statement of this sort would be Key estimation, Tempo estimation etc.

Modulo7 only includes a unique yet novel indexing scheme. This indexing involves on lookups based on .This indexing scheme allows for fast lookups for certain types of queries (e.g find all songs that in the key of CMaj)

# Chapter 2

# Literature Review

Music Information Retrieval is an active and vibrant community. Both academia and industry diligently pursue it albeit with different goals in mind. While academia's primary aim is to explore particular problems (e.g cover song detection, estimating chords from chroma vectors[2] ) etc. Whereas Industry is primarily interested in solving problems like song recommendation and similarity searches. The following sections outlines the software efforts and research problems tackled by MIR community in general.

## 2.1 Current MIR Software

Both Industry and Academia have created an extensive set of software for solving these problems. The author presents an overview of such software and the problems

they attempt to address. The following software packages were investigated

### 2.1.1 jMIR

jMIR,[1] or Java Music Information Retrieval tool set is a collection of java code, GUI, API and CLI tools for the purpose of feature extraction from variety of music sources (in particular audio, midi) and mine cultural information from the web. jMIR extracts and exhaustive set of features that can be used in machine learning tasks. The primary use of jMIR is automatic music classification and feature extraction and not similarity computations per se (which is one of Modulo7's core goals). Moreover jMIR does not scale to myriad sources of music in existence. Unlike Modulo7 jMIR also relies on faithful recordings and does not attempt to fill up missing information (like key signature not being encoded etc). Nevertheless its one of the best Open Source MIR software in existence especially for MIR research.

### 2.1.2 Marsyas

marsyas[3] (Music Analysis, Retrieval and Synthesis for Audio Signals) is a software stack for audio processing with specific emphasis on Music Information Retrieval and music signal extraction. Marsyas is a heavily developed and a widely developed state of the art framework for audio processing but also has a steep learning curve. Modulo7 has different goals (multiple format support, music similarity etc).

### 2.1.3 SIMILIE

SIMILIE[4] is a set of tools for music similarity measures used for single melodies and features multiple ways to construct vector space models for melodies. The techniques used in SIMILIE are novel. Modulo7 uses a subset of these similarity measures as basis for an extended and improved model of similarities based on polyphonic music and harmonic elements. Moreover SIMILIE needs its own file format (called .mcsv) for analysis. Although the software package gives a converter for different sources, its not as variegated as Modulo7's format support is.

### 2.1.4 Echo Nest APIs

Echo Nest is a company that specializes in big data music intelligence. Echo Nest power many music platforms like last.fm, Spotify etc. In particular Echo Nest provides APIs for extraction of audio features, acquiring artists similar to a particular artist etc. Echo Nest API is used for some sub tasks in Modulo7 (which is discussed in the Software Architecture Chapter).

Echo Nest also maintains the worlds biggest music database as well as data mined from them along with extracted audio features, web mined information, user preference etc).

### 2.1.5 Humdrum

Humdrum[5] is a set of tools for computer based automation and assistance in music research. Humdrum has the capability for solving very complex questions using music theoretic concepts. Humdrum supports its own file format for analysis. Humdrum is specifically designed for musicologists for automating tasks that they otherwise would have required manual analysis and not gathering statistic, music classification or music similarity analysis as an end goal.

### 2.1.6 Gamera

Gamera[6] is Optical Symbol Recognition Open Source software based on supervised and hybrid learning approaches for training. Gamera is designed with the particular aim of symbol recognition of old documents. Gamera also supports creating of new plugins for custom tasks. For the purpose of Music Information Retrieval, gamera can be used to solve the problem of Optical Music Recogntion (OMR) since sheet music images are also a format for music source.

### 2.1.7 Audiveris

Audiveris is an Open source software for Optical Music Recognition. Unlike Gamera, Audiveris can be directly consumed as a service for the purpose of OMR. Audiveris is used as service in many leading Notation Platforms like Musescore etc. As such,

Audiveris is used as a subcomponent of Modulo7's architecture for OMR.

## 2.2 Music Representation Formats

Modulo7 parses multiple formats for music. But there are many other sources that are worth mentioning.

**GUIDO :** GUIDO musical notation format is a computer notation format that is made to logically represent symbolic musical information that is easily readable by both humans and computers and can be stored as a text file. This example has been taken from wikipedia which shows the ease with which GUIDO represents a musical phrase.

## 2.3 Typical problems of MIR

On top of the generic software created by researchers and industry experts, researchers have tackled specific problems in Music Cognition, Classification, Cover song identification, Query by Humming Systems etc. Only certain approaches have been incorporated in Modulo7 which help completing metadata information (e.g. if the key signature of a song is not present, Modulo7 estimates it using ). Broadly speaking though, the problem statement falls in the following broad categories :-

### 2.3.1 Music Classification / Genre Identification

The problem of music classification is to assign a tag (also called a genre to a song) which broadly categorizes it according to some criteria. While the genre definitions for songs are often vague, it helps in giving information about which songs are relevant. Companies like Pandora and Microsoft assign genres to songs via musicologists[7] which means highly trained people manually classify music. Such approaches are expensive in terms of human labor and prone to error. Automatic Music Classification takes a different approach using different algorithms and machine learning approaches like jMIR[1] does to classify music.

### 2.3.2 Music Similarity Analysis

The problem of music similarity analysis lies at the heart of a large number other applications like Song Identification, Query by humming systems etc. Most literature have addressed the problem of melodic similarity[8] and not on generic polyphonic similarity. There are many systems and music databases in existence for the purpose of music similarity analysis.

### 2.3.3 Automated Musicological Research

In many cases musicological research is pursued manually by applying rules and music theoretic criteria. An example would be applying counterpoint analysis techniques

given in a treatise[9] to music sheet manually. This is labor intensive and the research community tries to address automated analysis of music. A significant effort is done by the Humdrum community.[5] in automated musicological research.

### 2.3.4 Audio processing and feature extraction

Most music is represented in Audio format rather than symbolic format, as consumption of music is primarily for the layman or the musically uninitiated. One such task would be music transcription(also known as melody extraction[?]).

### 2.3.5 Intelligent archiving and Retrieval

# Chapter 3

# Basics of Music Theory

Music theory is defined as the systematic study of the structure, complexity and possibilties of what can be expressed musically. More formally its the academic discipline of studying the basic building blocks of music and the interplay of these blocks to produce complex scores (pieces of music). Modulo7 is built on top of western theoretic principles and hence only western music theory is explored. Also music theory is an extremely complicated subject and hence only the basics and relevant portions to the modulo7 implementation are discussed here.

Traditionally music theory is used for providing directives to a performer to play a particular song/score.

This section is primarily meant for people with a weak or lack of understanding

of western music theory. The following section talks about the basic building blocks of music theory:-

## 3.1 Building Blocks

Music is built on fundamental quantities (much like matter is built on fundamental quantities like atoms and molecules). The following are the core concepts in order of atomicity (i.e successive blocks build on the preceding ones)

**Pitch/Note:** A pitch is a deterministic frequency of sound played by a musical voice (instrument or human). In western music theory, certain deterministic pitches are encoded as Notes. For example the note A4 is equal to 440 Hz. In other words Notes are symbolic representations of certain pitches. With certain notable exceptions, most music is played on these set frequencies.

Each note is characterized by two entities. First is the note type and the second is the octave. An octave can be considered as a range of 12 notes. There are 8 octaves numbered 0 to 7 which are played by traditional instruments or vocal ranges. Then is the note type. Notes are categorized into 7 major notes (called A, B, C, D, E, F, G) and 5 minor notes (also called as accidentals). They can be characterized by increasing or decreasing the frequency of the notes by a certain amount

(called sharps(#) and flats(b) respectively). For example the accidental lying in between (A and B is called A# or Bb). Similarly accidentals lie in between C, D; D, E; F, G and G, A. (Note that there are no accidentals in between B and C and E and F).

**Semitone and Tone:** A semitone is an increment or a decrement between two notes. For instance there is one semitone in between A and A#. Similarly there are 3 semitones in between A and C. A tone is an increment in between two major notes. Another characterization of a tone is two semitones.

**Beat/Tick:** A beat or tick is a rhythmic pulse in a song. Beats in sequence is used to maintain a steady pulse on which the rhythmic foundations of a song is based.

**Pitch duration:** A pitch duration is a relative time interval the pitch persists on a musical instrument. For example a whole note will persist twice as longer as a half note.

**Attack/Velocity:** The intensity or force with which a pitch is played. This parameter influences the loudness of the note and in general the dynamics of the song (covered in a subsequent section)

**Chord:** A chord is a set of notes being stacked together (being played together at or

almost at the same time). Chords are the basic building blocks of a concept called harmony (which will be discussed further on.). Traditionally a chord is constructed by stacking together notes played on a single instrument, but a chord can be constructed by different instruments simultaneously playing different notes.

**Rests:** Rests are pauses in between notes (with no sound being played at that point of time) for a fixed duration.

**Melody:** A melody is a succession of notes and rests which sound pleasing. There are many rules about what makes a melody sound good which we will get to in the subsequent reading.

**Harmony:** A harmony is a succession of chords (also known as a chord progression) along with the principles that govern the relationships between different chords.

**Voice:** A voice is an interplay of notes, chords and stops by a single instrument/vocalist. The reader can think of a voice as a hybrid or generalization of the melody and harmony concepts.

**Register:** For a given voice, the register of a voice is the range of notes that the singer of that voice can comfortably sing or a musical instrument sounds nice in.

**Range:** For a given voice, the range of a voice is the range between the maximum and minimum notes that a singer can sing or a musical instrument can play.

**Score/Song:** A score or a song is an interplay of voices. It is the final product of music that is delivered to the audience. Songs are of different types based on cultural context and complexity (for example an orchestra is a large number of voices being coordinated by a conductor. In contrast a folk song might be played by a single person on a guitar or a duet between a vocalist and an instrumentalist).

**Interval:** An interval is the relative semitone distance between any two notes. Intervals are categorized as melodic(semi tone distance between successive notes in a melody) and harmonic intervals (semi tone distance between notes within a chord).

## 3.2 General Concepts in Music Theory

On top of the building blocks of music, there are certain generic ideas or concepts on which music is based. The following sections describe them :-

**Polyphony/Homophony:** A homophonic song involves exactly one voice in the song. An example would be a single person singing a tune. A polyphonic song is one which involves two or more voices transposed with one another. An example of

polyphonic music would be a Western classical orchestra of a rock band performing a chorus.

**Phrase:** A musical phrase is a subset of the song that has a complete musical sense of its own. One could think of phrases as musical sentences, whereas a voice could be considered a paragraph. As a side effect a musical phrase can be played independently and still be considered as a song albeit an incomplete one.

**Meter:** The meter of a song is an expression of the rhythmic structure of a song. In context of western classical music, its a representation of the patterns of accents heard in the recurrence of measures of stressed and unstressed beats. Meters dictate the rhythm or tempo in which a song is played.

**Key/Tonality:** Tonality or key of a song is a musical system in which pitches or chords are arranged so as to include a hierarchy of relation between musical pitches, stabilities and attractions between various pitches. For example if the song is in the key of C, C is the most stable pitch in that song and other pitches like B have a tendency to go towards C (also called resolution of a phrase) to inculcate a sense of completeness. Moreover other pitches in relation to this pitches have various degrees of stability.

**Scale:** A scale of a song is an ordered set of notes starting from a fundamental frequency or pitch. If viewed ascendingly or descendingly (increasing/decreasing frequency of the pitches respectively) on this ordering, a scale describes a relationship between successive notes and their semitone distances from each other. A scale restricts the set of notes being played once the fundamental pitch is determined.

**Key Signature:** A key signature is a key along with a scale defined for a song (or in other words the fundamental pitch of the scale of the song is the same as the key of the song). A key signature is an expression of coherence for a song as well as a well defined set of notes that can be played for this piece, and as a result a song does not have notes that are outside of this key signature.

**Chromatic Music:** Chromatic music is any music that does not have a well defined key signature. Alternatively chromatic music can be categorized as music which is in the chromatic scale (chromatic scale is a scale in which all semitones in western music is present). Chromatic music is more difficult to analyze due to its lack of structure.

**Melodic Contour:** Melodic contour is the "shape" of melody. A melody with pitches going monotonically upward in frequency is called an ascending contour. Similarly a melody going monotonically downwards in frequency is called a descending contour. There are many other kinds of contour in music theory.

**Dynamics:** Dynamics is a coarse idea which indicate the variety of relative loudness between notes, speed of notes being played across phrases and other such ideas.

**Counterpoint:** Counterpoint is a musical phenomenon of two or more independent voices being interleaved to produce a rich and more interesting piece of music. Counterpoint pieces sound more interesting than the sum of their parts. Counterpoint is the basic fundamental on top of which orchestral pieces are built.

# Chapter 4

# Mathematics of Modulo7

The following sections describe the mathematical concepts used and implemented in Modulo7.

## 4.1 Preprocessing Steps

It might be that the input sources require certain preprocessing for any mathematical model to work. The following sub sections describe certain preprocessing operations that can be done in order to prepare input data to be transcribed into a vector space model.

**Tonality Alignment :** In order to compare two songs in different keys, the songs must be transposed to one key. This transposition shifts every note by a certain

interval (same as the intervalic distance between the keys of the input songs.) This is analogous to correcting a global offset such that similarity measures based on string representations of music can be applied. Mathematically consider Songs $S_1$ and $S_2$ with their respective keys being $K_1$ and $K_2$. Define intervalic distance between keys as $K_{intervalic} =$

## 4.2 Vector Space Models of Music

In traditional text based information retrieval retrieval systems, documents are indexed and a vector space representation of documents are created. Typical approaches for counting term frequencies or some weighting scheme like Term Frequency-Inverse Document Frequency Approach (TF-IDF). Analogous to text based IR, Music data can also be expressed as a vector space based on the approach taken. Some of these approaches are taken from the SIMILIE[?] but generalized for polyphonic music. Many approaches are novel based on the author's music theoretic studies.

### 4.2.1 Vector Space Models for Monophonic Music

Certain vector space models are with respect to a single voice. This allows vector space models to be represented as arrays with simple methods of computation that can be applied on top of it. First define pitch as a real number/string depending on context such that at given instant time $t_i$ either frequency $p_i$ is being played or its

note representation $p_i$ is being played. With this simple definition of pitch and onset time we can define our vector space models as follows

**Pitch Vector:** A voice can be expressed as a sequence of pitches $n_i = (p_i, t_i)$ where $p_i$ is the pitch and/or the set of pitches at instant of time $t_i$. The symbolic representation of music essentially a discretized version of these values from music sources and hence a vector representation can be made. A voice V can be represented as a vector

$$P =< n_1, n_2, ...n_n > \tag{4.1}$$

A similar vector representation could be when the time information is eschewed in favor on only the pitches. This vector is called the raw pitch vector and is denoted as the follows :-

$$R =< p_1, p_2, ..., p_n > \tag{4.2}$$

**Pitch Interval Vector:** Another way to look at elements is the interval spacing between elements. This is same as the interval concept in the music theory chapter. Mathematically an interval is defined as $\Delta p_i = p_i - p_{i-1}$. And thus an pitch interval vector is defined as

$$PI =< \Delta p_1, \Delta p_2, ..., \Delta p_n > \tag{4.3}$$

**Rhythmically weighted Pitch Interval Vector:** In order to include the rhythmic information in the pitch interval Vector, define rhythmically weighted pitch as $rp_i = \Delta p_i \times t_i$. Now the rhythmically weighted pitch vector can be represented as

$$RPI =< rp_1, rp_2, ...rp_n > \tag{4.4}$$

**Normalized Tonal Histogram Vector:** The tonal histogram is a vector or map of 12 distinct intervals present in western music theory and the number of time. Each position in the vector corresponds to the total number of times that interval has occurred in a voice. Mathematically define $\Delta P^{voice_j} = \sum_{i=1}^{len(voice)} p_i^{voice_j}$. Define interval fraction as : $\Delta p_i^f = \frac{\Delta p_i}{\Delta P^{voice}}$. Thus we can define the normalized tonal histogram vector as

$$NTH =< \Delta p_1^f, \Delta p_2^f, ..., \Delta p_n^f > \tag{4.5}$$

**Normalized Tonal Duration Histogram Vector:** The tonal duration histogram is a vector or map of 12 distinct intervals present in western music theory. Each position in the vector corresponds to the cumulative duration for which that interval has occurred in a voice. Mathematically define $\Delta T^{voice_j} = \sum_{i=1}^{len(voice)} t_i^{voice_j}$. Define durational interval fraction as : $\Delta t_i^f = \frac{\Delta t_i}{\Delta T^{voice}}$. Thus we can define the normalized tonal duration histogram vector as

$$NTDH =< \Delta t_1^f, \Delta t_2^f, ..., \Delta t_n^f > \tag{4.6}$$

### 4.2.2 Vector Space Models for Polyphonic Music

While vector space models per voices are useful in single melody similarity, its not enough to ascertain similarities or compute statistics about songs when songs have more than one voice. As such certain extensions are proposed to the voice vector representation that allow for similarity computations for polyphonic music.

**Normalized Song Tonal Histogram Vector:** Consider the normalized tonal histogram in section. We can define the cumulative intervals across all voices as $\Delta P_{song} = \sum_{i=1}^{num(voices)} \Delta P^{voice_j}$. Define interval fraction over song as $\Delta p_i^{fs} = \frac{\Delta p_i}{\Delta P^{song}}$ Now define the normalized tonal histogram vector for the song as the follows:-

$$NSTH =< \Delta p_1^{fs}, \Delta p_2^{fs}, ..., \Delta p_n^{fs} > \tag{4.7}$$

## 4.3 Similarity Measures

Similarity is defined in Modulo7 as a function which takes as input two voices or songs and outputs a value between 0 to 1 where 0 stands for least similar and 1 stands for most similar. Similarity measures are a cornerstone of recommendations

and many recommender engines are based on rank similarity measures for different criteria. Mathematically :-

$$Sim_{song}(S_1, S_2) \in (0, 1) \tag{4.8}$$

$$Sim_{voice}(V_1, V_2) \in (0, 1) \tag{4.9}$$

### 4.3.1 Similarity Measures for Monophonic Music

Similarity measures are different concepts for monophonic and polyphonic music as it stems from comparing different vector representations. For the following sections assume vectors of equal length. In a further section 4.3.3 we extend standard similarity measures to vectors of unequal length.

**Edit Distance on Raw Pitch Vector Representation:** Consider the raw pitch vector in equation 4.2. This vector is essentially a vector of tokens or equivalently a string. Hence standard edit distance algorithms in normal text IR can be applied to it (e.g Leveinstein Distance, WagnerFischer algorithm etc[10]).

### 4.3.2 Similarity Measures for Polyphonic Music

In order to incorporate vector space models to polyphonic similarity, monophonic measures can be extended in order to accomodate for polyphony. Another approach

would be to apply measures.

Generic maximal voice similarity An approach would be to take pairwise voice similarities between two voices of a song, and then representing the max of these pairwise computed similarities. This model is especially useful in cases where comparing a melody against a song which contains a similar melody. Mathematically

$$GMVS(S_1, S_2, VSim) = arg_{max}(VSim(V_i, V_j)) \; s.t \; V_i \in S_1 \; and \; V_j \in S_2 \qquad (4.10)$$

### 4.3.3 Similarity of vectors of unequal length

Its almost certain that two voices will never have the same length. Hence its important at this point to ascertain how to map similarity measures to unequal length voices. Consider two voices $\vec{A}$ and $\vec{B}$. Without loss of generality assume $len(A) > len(B)$. Let i be an index running from 0 to len(A) - len(B) and $C_i$ be defined as the sub component of a melody (also called submelody) which contains voice elements $A[i] \; to \; A[i + len(B)]$. Then similarity measure $S_{unequal}(S_{original}, A, B)$ can be defined as the following

$$S_{unequal}(S_{original}, A, B) = max_i(C_i, B) \text{ where } i \in \{0, len(A) - len(B)\} \qquad (4.11)$$

This procedure is a basic modification of the concept of Horizontal Shifting.[?] Here we keep shifting the smaller melody along the longer melody successively for faster

computation. This technique is best used when one melody's vector space representation is completely contained in another (for an example a phrase of a voice compared with a voice itself).

### 4.3.4 Meta Data based similarity

All the similarity measures considered so far is based on similarity on voices and sets of voices in a a song. However there are other global properties of a song such as the key signature or the time signature of a song. This global information can give us context about a song's particular characteristics (for example songs in Minor scale are generally sadder than songs in the Major scale). Hence estimates can be more quickly derived by comparing meta data features rather than voices (whose computation). These similarity measures can be used for a additional purposes(for example completing incomplete meta data).

### 4.3.5 Tonal Similarity

Often pieces of one key are similar to pieces on a different key, simply based on the fact that the keys themselves are similar.

## 4.4 Criteria Analysis

While Modulo7's primary goal is on comparing similarities between pieces, often its better to ascertain whether a certain piece satisfies a certain music theoretic predicate. Some example problems of such sorts are if the piece has a species 1 counterpoint (i.e. the voices move with the exact same speed) or if the piece has voices in the STAB criteria (with exactly 4 voices and their ranges being in particular range of high and low notes). This allows a consumer to build complex queries based on pieces satisfying selectivity requirements on top of similarity measures or alternatively if certain pieces just satisfies a one or more criteria. Following are the criteria implemented in Modulo7

### 4.4.1 Simple criteria

Simple criteria are based on simple global level properties of the song.

**Polyphonic Criteria:** Its a simple criteria which decides whether a piece of music is polyphonic or not. This is decided on the basis of the number of voices in the song. Mathematically

$$PC(Song) = \begin{cases} true & num_{voices}(Song) = 1 \\ false & otherwise \end{cases}$$

**Key Signature Equality Criteria:** Its a simple criteria that checks if a song is in a particular key or not. Mathematically

$$KSC(Song, desiredKeySig) = \begin{cases} true & Key_Signature(Song) = desiredKeySig \\ false & otherwise \end{cases}$$

# Chapter 5

# Software architecture and Methodology

The following sections present the software architecture of Modulo7.

## 5.1 Server Side architecture

Modulo7 is designed with the purpose of scalability. A block diagram of the components of the server side architecture is presented below :-

1. Source Converter : Converts music sources (e.g. music XML, midi etc) into modulo7's binary representation.

2. Music Theory Models : The model is a description of music theoretic criteria that can be applied on top of a song. Examples would be melodic contour, tonal

histogram etc.

3. Distributed Storage Mechanism : The modulo7 internal representation is a conversion to create a song representation with all the meta data of the song (Key, Scale, etc) along with the sequences of note events stored as lists. This representation is then serialized and stored in and Hadoop Distributed File System. This allows for fault tolerance and a distributed deployment of the input data.

4. Lyrics Indexer : A distributed index of songs lyrics. This acts as a base on which standard techniques for similarity analysis might be applied. Alternatively it can provide a framework on which custom models (e.g. semantic intent of the song, correlation between music theory models and lyrics might also be applied).

5. Lyrics similarity models : A set of similarity models that can be applied to an index.

6. Query Engine : An SQL like interface to a client that allows you to gather and ascertain useful information (based on music theoretic criteria).

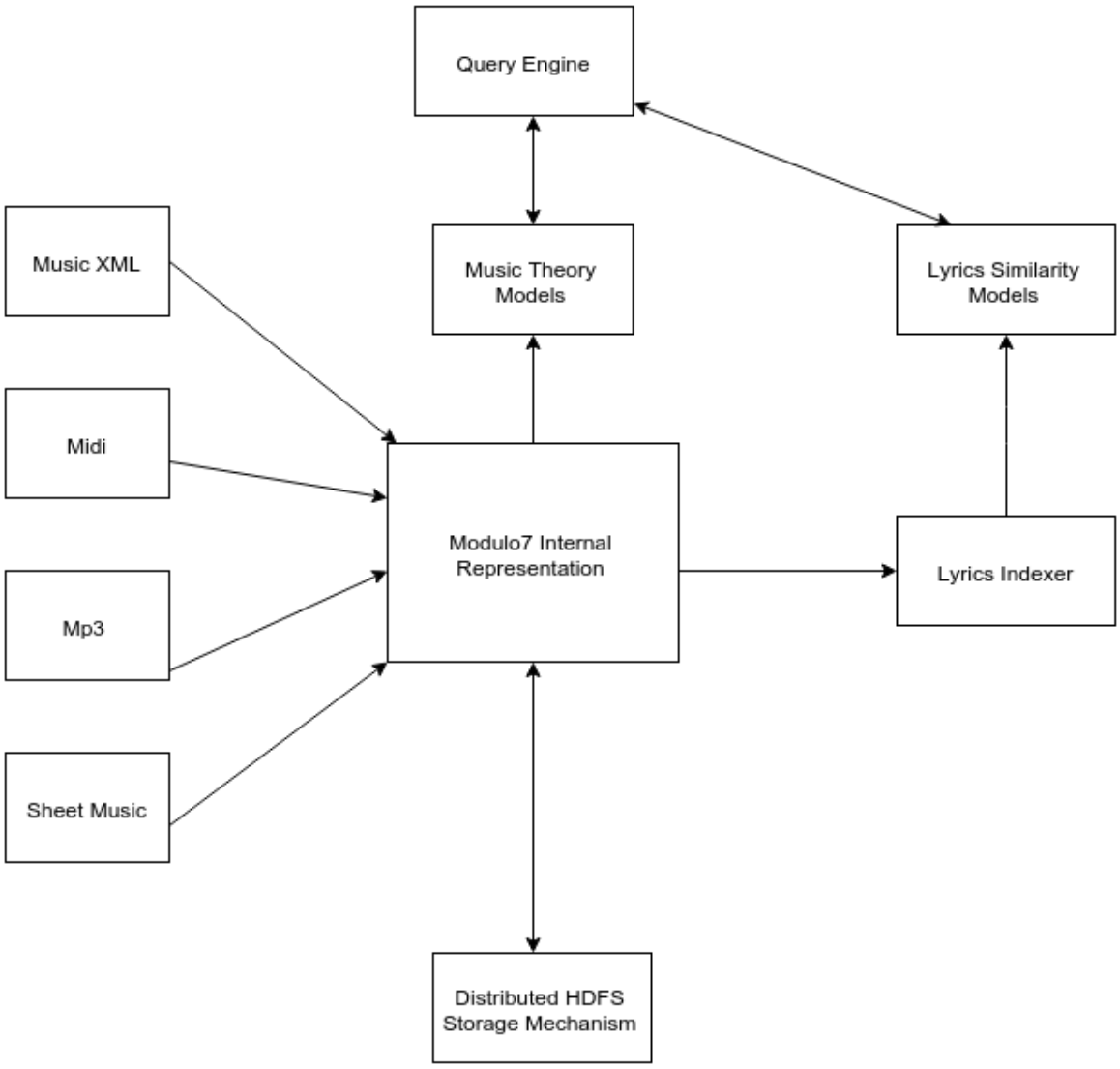


**Figure 5.1:** Modulo7 architectural design

## 5.2 Client architecture

The server exposes a sql like interface as well as a consumable API. Some sample queries would be :-

1. select midi files from database where $melodic\_complexity > somethreshold$
2. select * from database where $artist = led\_zepplin\ and\ harmonic\_movement > harmonic\_movement(stairway\_to\_heaven)$
3. select $num\_voices\ from\ Database\ where\ songName = someSong.midi$

An API will also be exposed to the client along a remote invocation procedure. The API would primarily target single sources for specifics. Some example API would be :-

1. int getNumVoices(String midiFilePath)
2. double melodicContourMovement(String pngSheetFilePath)
3. double compareAverageAttack(String musicXMLFile)

This API can be consumed for specific song analysis. As design this API will not work on a bulk of files like its sql counterpart.

Moreover the client also exposes a highly customized search engine based on the custom vector space representation of features extracted by Modulo7.

## 5.3 Song sources

At the heart of Modulo7's design is its song sources adaptors (or converters) into its own internal binary format. Each music source is a different representation and while certain sources ascribe what how music should be played (e.g musicxml, sheet music), other formats ascribe what is actually being played (e.g midi, mp3). There are many other music sources in existence (e.g guitar tablature, GUIDO format, humdrum format), but for the purposes of breadth and ubiquity, these four sources have been targeted as input for Modulo7. Note that acquiring features from each format is a domain specific challenge and inaccuracies are inherent because of that. The following subsections describe the individual formats in detail and the challenges encountered in parsing them.

### 5.3.1 Midi format

MIDI (short for Musical Instrument Digital Interface), is a technical specification for encoding of events on a midi enabled instrument and a protocol for interfacing and communicating between various midi enabled instruments. Typically any midi enabled electronic instrument when played, relays to its internal circuitry a message. Examples of such messages could be a particular note is being hit on a keyboard, a note is being hit off after being hit on, tempo based messages on the number of ticks per second etc. While MIDI is a technical specification for encoding music the

score is being played, Modulo7 treats it as a symbolic representation of music. Midi was also a simple and popular encoding format for music and gaming industry in the nineteen ninties.

A symbolic representation is a codification of music which acts a higher level of abstraction (individual notes or chords being played) as compared to lower level representations like audio files (which codify information like waveforms). Modulo7's internal representation is also a symbolic representation. Symbolic representations are easier to manipulate when applying a music theoretic criteria.

Midi is one of the easier formats to parse for musical specifications. Moreover there is a big volunteer community of midi encoders. As such acquiring and parsing non trivial amounts of midi data is not a very challenging task.

### 5.3.2 Western Sheet Music

Sheet music is one of the oldest forms of music in existence. Its a hand written or printed form of music that uses a specific script (a set of musical symbols on a manuscript paper) to ascribe music. Music Composers from Medieval and Modern periods of the western world use western sheet scripting to codify their work while performers play from these sources. A vast body of older work and particularly orchestral work is codified in sheet music.

Like midi, sheet music is also symbolic in nature. However unlike midi, its an ex-

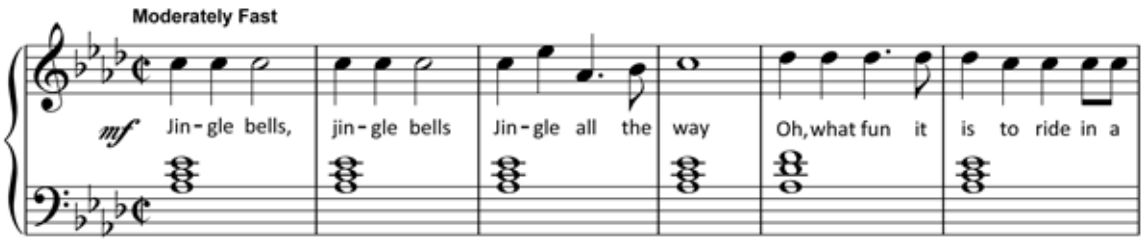


**Figure 5.2:** Jingle bells melody sheet music representation

pression of how a score should be played, rather than what is being played. Modulo7 converts digitized versions of these sheet music (e.g sheet music stored .tiff, .png. jpeg etc formats)

A very simple example of sheet music for describing a melody is shown below.

Parsing digitized sheet music is an extremely challenging task. It requires a solid understanding on Computer Vision and even the state of the art software in existence today cant handle all scores (especially a poorly digitized formats). Given the amount of domain knowledge required, Modulo7 uses a third party library called Audiveris .

### 5.3.3 Music XML format

Music XML format is a standard open format for exchanging digital sheet music. A music XML format is unusual as its a format that is easy to parse for computers and easy for humans to understand it. MusicXML formats are heavily used by music notation applications. Music XML format is a symbolic format and can be considered

a modernization of the Sheet music format. Its disadvantage however is unlike sheet music, a performer cant read the piece and play it on the spot directly.

Just like Western Sheet music and midi, music XML is a symbolic format as well. Music XML is also a transcription format which specifies how a score should be played.

### 5.3.4 MP3 format

For the sake of completeness, Modulo7 also supports an audio format called mp3. Its an audio encoding format that uses lossy compression to encode audio data. Mp3 gives a reasonably good approximation to other digital audio formats of music storage with a significant savings in space for storage. Its one of the defacto standards of digital music compression and transfer and playback on most digital audio players.

## 5.4 Modulo7 Internal Representation

Modulo7 consists of converters that convert data into Modulo7's internal representation. This representation can be thought of a document representation on which similarity measures described in Chapter 4 can be applied to. Moreover the internal representation can be thought of as an indexed meta data structure for any source of song from which relevant information can be acquired. Hence Modulo7 indexing schematic is a symbolic representation of music much like music xml and sheet

music. The converters are responsible for converting different music sources to this representation format. Its important to note that depending on there source one or more of the subcomponents of the internal representation may be missing or wrong. Modulo7 indexes songs based on each of these criteria and on top of these boolean queries can be formulated. The components are broadly categorized as the following:-

**Song Metadata:** The metadata aspects in a song e.g. The name of the song/ the composer/performer's name, Key Signature of the Song, Meter of the Song etc. These are global properties of the song.

**Voices in a song:** Similar to the Voices in Music theory, Voices in Modulo7 represent the same symbolic data as is present in the sources from which the information is parsed.

**Lyrics of a song:** The textual representation (along with delimiters for line breaks) for the lyrics of a song. Lyrics can live independently as separate entities (if the input to Modulo7 is a text file containing the lyrics and no other information). However midi/musicxml and sheet music have optional lyrics elements present in their transcriptions and Modulo7 transcribes from those.

In most cases though lyrics exists as a separate entity from songs. In such cases,

Modulo7 separately indexes lyrics. In certain datasets, the lyrics representation is different (for example the million song dataset has a representation format as a bag of words with counts of the words occuring for each format[11]). Modulo7 accomodates such formats as well.

## 5.5 Methodology

This section contains the methodology followed in the information retrieval phase and then the indexing steps taken after the domain specific conversion is completed by Modulo7's adapters

1. Given a root directory, Modulo7 recursively parses all the sheet music image files, mp3, midi and music xml files. Depending on the file type individual parser modules are invoked and an internal representation is created in memory and serialized to disk (depending on user preference)

2. Modulo7 then indexes all the objects created on specific meta data (such as key signature, time signature and artist of a song). Moreover it also creates a lucene index on lyrics extracted. It stores all these indices in memory.

3. Modulo7 then exposes a prompt to the consumer which contains a set of standard querying options along with a SQL like querying interface. Consumer can then choose the option they like and query the constructed database.

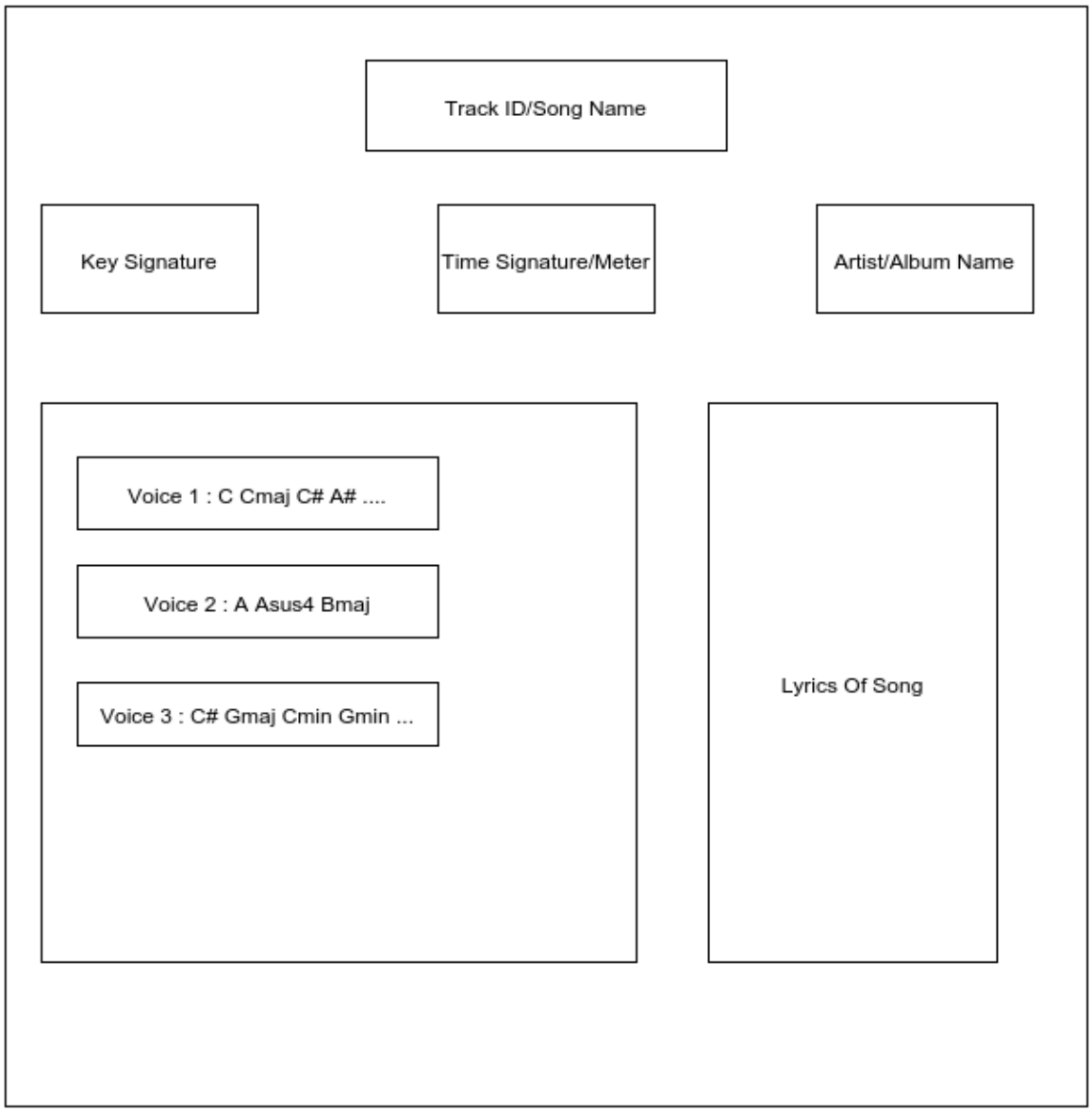


**Figure 5.3:** Modulo7 internal representation

### 5.5.1 Modulo7 standard query set

Modulo7 exposes a standard set of querying features to the consumer. These queries are useful to extract simple information from the uploaded dataset to Modulo7.

### 5.5.2 Modulo7 SQL Algebra Specifications

# Chapter 6

# Experimental Evaluation

For the purposes of evaluating Modulo7, test cases have been designed into two formats. One category of testing is micro testing, for validating correctness and precision recall for small sets of data. This ensures verifiability of algorithms and similarity measures on small datasets as well as novel explorations of data. Most MIR research is done on small scale datasets and hence falls in the purview of micro testing. The other format is macro testing which involves large datasets such as the million song dataset.[11]

A few assumptions that are made in testing are as follows :-

1. In order to estimate ground truth values, the author assumed ground truth values presented in datasets used / or subjective judgements to which songs are similar to each other. These subjective judgements are procured from existing

literature.

2. If the song metadata (such as keysignature, timesignature, total duration of song) is not encoded, its estimated by the parsers. This estimation id done by existing algorithms in literature. However if metadata is encoded, its assumed to be correct.

3. Most tests are against file formats of the similar types (for example midi is tested against other symbolic files). This is due to the inherent complexity of symbolic decoding of audio formats like mp3. Also its easier to compare symbolic data against other symbolic data.

4. In the event of parsing data, there can be legal issues (e.g. the song can be copyrighted). For that reason custom parsers to build alternate research datasets (e.g the million song dataset has already derived features that Modulo7 intended to derive for Mp3 files.[11])

5. All evaluations are done against research datasets which are published in academia or exposed as public datasets in industry.

## 6.1 Results of Index Compression

The Modulo7 representation can be thought of an indexed meta data version of the song. True to all indexed data, Modulo7 represents the song in a much smaller size

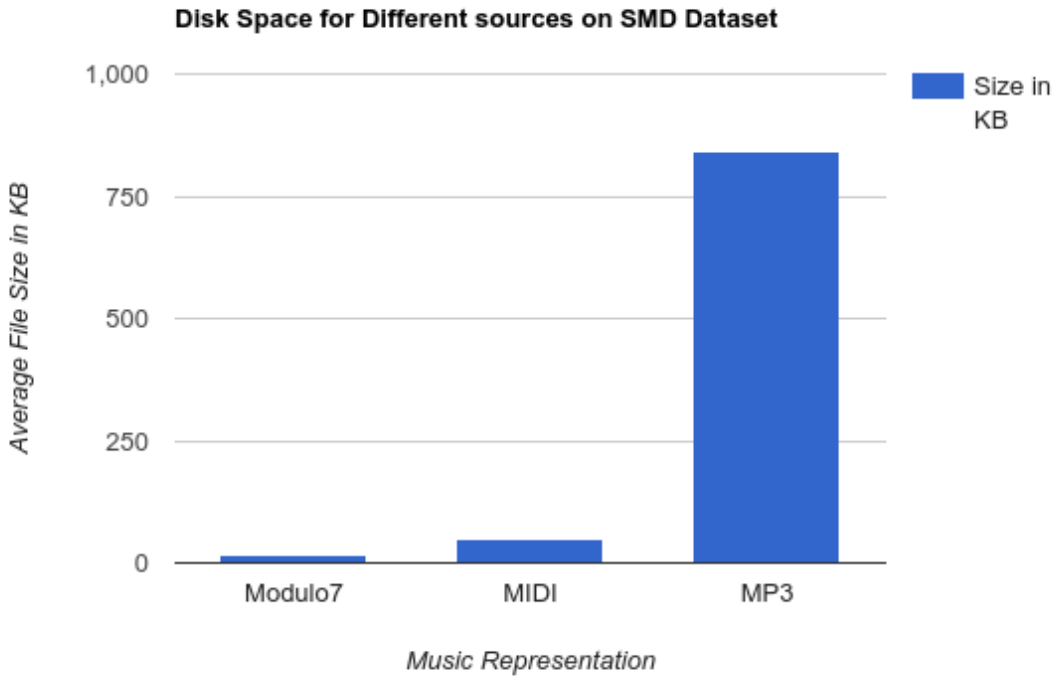


**Figure 6.1:** Modulo7 architectural design

than the original source. The following chart demonstrates the average compression of indexed data as compared to source files on the Saarland Music Data (SMD) Dataset:[12]- As expected Modulo7's serialized format expresses a song in less disk space than its source formats while keeping the symbolic information intact. The results are positive as there is a 4 time decrease in size of expressing symbolic information as compared to midi files.

A similar transformation was also done on the wikifonia dataset

## 6.2 Results on similarity measures

# Appendix A

# Third Party Libraries Used

Modulo7 is a significant software engineering effort.

# Appendix B

# Algorithms is use in Modulo7

There are certain algorithms in literature that are directly implemented in Modulo7. These algorithms facilitate the smooth functioning of Modulo7's indexing in face of incomplete metadata. Some notable algorithms that have been used are briefly described in the following subsections.

## B.1 KK Tonality Profiles - Key Estimation

Many music sources have the key signature inscribed in it. For example a midi file might have the key signature bytes transcribed.

# Bibliography

[1] C. McKay, *Automatic Music Classification with jMIR*. Montreal: McGill University, 2010.

[2] M. D. P. Adam M. Stark, "Real-time chord recognition for live performance."

[3] P. G. Tzanetakis, "Marsyas a framework for audio analysis," *Organized Sound*, vol. 4(3).

[4] D. M. Klaus Frieler, "The simile algorthms for melodic similarity."

[5] "The humdrum toolkit: Reference manual. menlo park, california: Center for computer assisted research in the humanities, 552 pages, isbn 0-936943-10-6." p. 552 pages.

[6] I. F. Karl MacMillan, Micheal Droettbroom, "Gamera: Optical music recogntion in a new shell."

[7] D. M. N. Scaringella, G. Zoia, "Automatic genre classification of music content: a survey."

BIBLIOGRAPHY

[8] *MELODIC SIMILARITY: APPROACHES AND APPLICATIONS.*

[9] L. Cherubini, *A Treatise On Counterpoint Fugue.* Novello, Ewer And Co.,, 2010.

[10] G. Navarro, "A guided tour to approximate string matching."

[11] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[12] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, "Saarland music data (SMD)," in *Late-Breaking and Demo Session of the 12th International Conference on Music Information Retrieval (ISMIR)*, Miami, USA, 2011.