International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:1, 2014

Knowledge Based Chords Manipulation in MES

V. Hepsiba Mabel, K. Alagarsamy, Justus S.

***Abstract*—**Chord formation in western music notations is an intelligent art form which is learnt over the years by a musician to acquire it. Still it is a question of creativity that brings the perfect chord sequence that matches music score. This work focuses on the process of forming chords using a custom-designed knowledgebase (KB) of Music Expert System. An optimal Chord-Set for a given music score is arrived by using the chord-pool in the KB and the finding the chord match using Jusic Distance (JD). Conceptual Graph based knowledge representation model is followed for knowledge storage and retrieval in the knowledgebase.

***Keywords*—**Knowledge, Music, Representation, Knowledgebase, Chord-Set.

I. Introduction

A chord is a combination of relevant musical notes arranged to produce pleasing harmony along the melody line. Musicians have to be creative and artistic in forming a chord for a piece of music score. This skill is attained over the years of regular practice and performance. However, a chord fixed for a phrase of music is often criticized or argued for its authenticity and correctness in the musician's creativity [1]. A formed chord for a music score which is acceptable for one musician may not be accepted nor truly acknowledged by another musician. Nevertheless, this is not a matter of forming a correct or an incorrect chord, rather a more optimum chord for the given music score. Still musicians perform and publish their masterpieces mindless of these criticisms.

This work has studied the universal problem of optimal chord formation and has attempted to propose a knowledgebase system which would help a musician choose optimal acceptable chords among the Chord set delivered by the Music Expert System (MES) that works on our designed knowledgebase. A framework is designed to address this issue, and it has the following steps: 1) Initiate the knowledgebase with a rich set of chords, 2) Represent the given music score in the CFR Model (a knowledge representation model using conceptual graph), 3) Retrieve Chordset from the knowledgebase for the music score, bar-wise. The musician is given choices to choose an optimal chord from the generated Chordset. Optimality is assigned to each chord in the Chordset.

The paper is organized as follows: Section II presents the music terms and a little understanding on Chords. Section III presents the knowledge representation model in MES. This includes the KB design. Section IV describes the Chord formation using knowledgebase, and the algorithms involved to calculate the chords and retrieval of K-balls, and a few experiments. Section V presents the discussions and conclusion of the work along with future directions.

V. Hepsiba Mabel is with the Madurai Kamaraj University, Madurai, TN, India (phone: +91-9840454781; e-mail: hepsi.mku@gmail.com).

Dr. K. Alagarsamy is with Madurai Kamaraj University, Madurai, TN, India (e-mail: alagar_samy54@gmail.com).

Dr. S. Justus is with the School of Computing Sciences and Engineering, VIT University, Chennai, TN, India (phone: +91-9789145400; e-mail: justus.s@vit.ac.in).

II. Music Theory & Terms

A. Major and Minor Scales

Given a music score, it will fit into any one of these two categories: a Major or a Minor Scale. Scale is defined as a series of seven distinct music notes, producing a universally accepted, well-defined basis for all music or masterpiece or song [2]. Major Scale has notes separated by whole tones except for the 3rd and 4th and 7th and 8th. (Every note carries ½ tone weights – the sequence from C to D is C, C# and D, ½ + ½ = 1, whole tone). Minor Scale has notes separated by whole tones except for the 2nd and 3rd and 5th and 6th. An example of a Major Scale and Minor Scale is shown below:

'C' Major Scale: C – D – E – F – G – A – B
'C' Minor Scale: C – D – D# – F – G – G# – A#.

B. Music Score

Music Score is a melody line, or a tune, composed and arranged sequentially based on the Scale (major or minor) which is denoted as Key Signature, Time Signature, and Clef Signature [3]. Consider Fig. 1.

Fig. 1 Example Music Score

This music score is written in D Major Scale, with 4/4 time count, and has 4 bars. 4/4 means, each bar contains a sum total of 4 counts. Let us not go detail into the notations and its measures as it is not relevant to this work.

C. Chords

A given music score is usually accompanied by the set of relevant notes in order to produce enriched music. A Chord is a combination of three or more notes that blend harmoniously when sounded together [3]. Consider Fig. 2: For the given melody score in the first line, relevant chords are given in the second line, which when played on a piano or organ will sound rich.

Fig. 2 Music Score with Chords

This series of chords are given by a musician according to his skill level. This set of chords may vary with different musicians.

D. Other Notations

The notations that are used to write the music score are: Minims takes 2 counts, Crochet takes 1, Quaver takes ½. Semi-Quaver takes ¼. The five lines are together called stave, and named from down to up: E, G, B, D, F. Refer [4], [5] for more music theory.

III. Knowledge Representation in MES

Music Expert System is a simple tool designed to simulate a musicians artistic skills of composing, harmonizing and orchestrating. Though these modules are still in the infant stage, we are working on the Chord Formation module in the harmonizing part. Earlier works on MES can be found in [6], [7]. The work has improved over the years, and now we are having a designed knowledgebase for music ontology, with which the intelligence and acceptance level of the chords formed are evaluated. This section details how the scores are represented in the knowledge base, and how knowledgebase is helpful in chord formation.

A. Conceptual Graphs

A piece of knowledge can be expressed in the form of a concept, which when connected conceptually forms a conceptual graph (CG).

Conceptual graph, according to John F. Sowa, [6] is "A finite, connected, undirected, bipartite graph with nodes of one type called concepts and nodes of the other type". Conceptual graphs address in terms of concepts and its attributes. A concept can be an entity, event or an action. Every concept has its own attributes and is instantiated with instances. The formation of conceptual graphs is as follows:

- Every arc a of G is a pair (r, c) consisting of a conceptual relation r and a concept c in G. The arc a is said to belong to r; if it has link r to c; but it does not belong to c.
- A conceptual graph G may have concepts that are not linked to any conceptual relation; but every arc that belongs to any conceptual relation r in G must link r to exactly one concept c in G

Now every music note that is written on stave is not independent. They depend on the stave, key signatures, time signatures in order to be semantic in the given music score. The same note may carry a different meaning when it is written for a different key signature. Hence it is preferred to define a music note as a concept, and since they are related to each other, they are conceptual related [8].

B. Concept Function Relation Model

This model is designed to represent this unit of knowledge, termed as K-Ball. It is a combination of the concept (c) and relation (r). Concept part is modeled as - attributes or properties of the concept (identifiers, values, etc), - functions/actions (procedures, member functions that processes the attributes) – Access Quantifiers (values, referents, relational identifiers etc).

Relation part is modeled as – set of arcs or links that are conceptually connected to this concept. This includes relations to instances of the concepts of same type or other non-similar instances of concepts. Fig. 4 represents the concept-function-relation model (CFR) Mode.

Consider the following example.

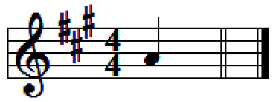

Fig. 3 Sample Music Note for Representation

Fig. 3 represents a single music note on the stave, with a clef sig., time sig., and key sig. The music note represents 'A3' on a piano.

Its semantic in this context is "A3 is a tonic note in a 3# major scale, with a 4/4 time sig., and on the treble clef". This same A3 can take a different meaning in different key, time, and clef sigs. Hence it is authentic to consider the notations are concepts (taking a higher level of intelligence).

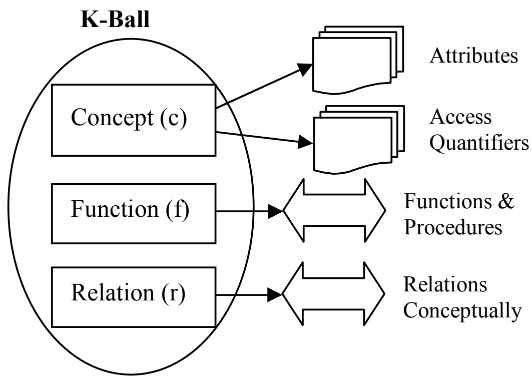

Fig. 4 CFR Model

C. Music Representation as a KBall

The concept and relation together in CFR model of Fig. 3, is a Knowledge Ball representing a music note here. The shape of the ball is given here for a knowledge unit, because it is represented in an Object-Relational Modeling platform and the knowledge units are stored as objects, giving a perception of a collection of connected K-Balls.

Table I gives the different concepts involved in this representation model. These objects are defined in Object-Relational Databases.

TABLE I
CLASS_TYPE DEFINITIONS IN ORDB

Create type Note_t(
Note_name varchar(2);
Note_id varchar(2);
Staff_sig varchar(6);
Duration number;
Position varchar(4);
Expression varchar (10);
Operations:
setvalues() integer;
getvalues() integer;
Relationship:
note_t Pre_note;
note_t Post_note;
Staff_t Staff_link **inverse**
Staff_t::Arc_set;
);
	Create type Key_t (
	Key_name varchar(3);
	Key_sharps[12] varchar(2);
	Key_flats[12] varchar(2);
	);
	
	Create type Key_Scale_t(
	Key_name varchar(3)
	inverse key_t::key_name;
	Scale_notes[12] varchar(2);
	);
	
	Create type sigtime_t (
	Numerator number;
	Denominator number;
	);

Create type Staff_t(
Concept_ID number;
Concept_type varchar(15);
Referent_type varchar(4);
Staff_order int;
Clef_sig varchar(2);
Key_type key_t;
Signature sigtime_t;
Key_Scale key_scale_t;
Operation:
set_values() integer;
get_values() integer;
Relationship:
Linked_staff Staff_t
Arc_set Note_t **inverse**
Note_t::Staff_link
);
	Create type Chord_t(
	Chord_name varchar(12);
	note[7] varchar(2);
	Relationship:
	Related_Chords Chord_t[]
	inverse this *Chord_t;
	};
	
	Create type Bar_t(
	Bar_Id int
	note[] Note_t;
	};

Presented in Table I are the primary classtypes for representing each concept as K-Ball. These classtypes are the ‘nodes’ and the relationships are the ‘edges’ of the conceptual graph that is stored and retrieved. Given a music score, it has to be represented as conceptual graph, which holds the objects of the defined classtypes. The typical representation of the music score as K-Ball is shown in Fig. 5.

The instances of objects created out of this model are stored in our knowledgebase that was designed over a relational table. The relational platform will serve as just a container to hold these K-Balls.

IV. CHORD-SET IN KNOWLEDGEBASE

In addition to these representation models and CG, a well-informed Knowledgebase is needed to form an effective Chord set for a given music score. This set up will help the MES to formulate chords in a better fashion, getting closer to a musicians preference.

A. Chord Formation Rules and Initiating the KB

A knowledgebase is a collection type supported in ORDB. This again is programmed based on the standard rule of the music theory. Given a Major or Minor, it has four primary related chords, and 20+ secondary chords [5]. For a note in a given major, it may have more than 30+ chords. For a complete list of related chords see [9].

A chord is designed in such as follows: For a given Major ‘M’, its scale starts from its note itself and proceeds to the seventh note. Based on this scale the chords are formed.

M chord: 1-3-5, M_7: 1-3-5-7
M_4: 1-4-5 M_6: 1-3-5-6
M_{dim}: 1-2 ½ -3 ½ M_9: 1-3-5-(2) and so on

where ‘M’ stands for any Scale from A to G.

```
Procedure Chord_Formation(All Chords)
{
    New Chord_t: chordsObj[];
    New Note_t: allNotes[]

    ChordsObj= findMajorChords(allNotes[]);
    ChordsObj= findMinorChords(allNotes[]);
    ChordsObj= findMajor7Chords(allNotes[]);
    ChordsObj= findMinor7Chords(allNotes[]);
    ChordsObj= findMajor6Chords(allNotes[]);
    ChordsObj= findMinor6Chords(allNotes[]);
    ChordsObj= findMajor4Chords(allNotes[]);
    ChordsObj= findMinor4Chords(allNotes[]);
    ChordsObj= findMajor2Chords(allNotes[]);
    ChordsObj= findMinor2Chords(allNotes[]);
    ChordsObj= findMajor9Chords(allNotes[]);
    ChordsObj= findMinor9Chords(allNotes[]);
    ChordsObj= findDimChords(allNotes[]);

    Return Bag(ChordsObj);
}
```

Fig. 5 Chords Formation: Algorithm

The procedure to find the possible chord formation for all combinations of chords in a given Major is in Fig. 5.

The algorithm given in Fig. 5 can be represented as follows.

$$\beta(ChordObj) = \left\{ \bigcup_{i=2,4,6,9}^{11} f(majors_i), \bigcup_{j=24,6,9}^{11} f(minors_j), \bigcup f(dim) \right\}$$

where β is a `bag_t` collection_type, a set of all the chord Objects. This set could be extended by appending new chords into this β(`chordObj`) chord set.

The knowledgebase is loaded with these collections of chords `chord_t`, before using them for forming the ‘Chord Set’ for a music score.

In the algorithm shown in Fig. 6, we have used only 13 simple chords for this study. There are still more complicated chords for each `key_t`, and a few accidental chords that could be included in any `key_t`. Hence there is a provision in this algorithm to append more chords to it, which will get added to the Chords pool in the knowledgebase.

B. Initiating Conceptual Relations

The initiated knowledgebase of Chords are now independent and they need to be connected using conceptual relation. Each `ChordsObj` has the chord information and the relations to other chords. This conceptual relation among chords is the key idea to retrieve related `ChordObj` objects for a given `Key_t` in one read operation itself, effecting the anticipatory fetching of related chords in a music score. Look at the definition of `Chord_t` in Table I, it has the notes for that chord and the related_chords for this `*chord`. The conceptual relations for each of the Chords in `Bag(ChordsObj)` are introduced as given in the algorithm in Fig. 6.

The procedure `createConceptualRelation()` is invoked every time a new music score is given as input to MES. It acts on the rich set of Chords in the KB. The procedure creates conceptual relation instantly for the given

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:1, 2014

score and loads the entire collection of Chords to the memory in one fetch/read operation request from the KB.

```
Procedure createConceptualRelation(Key_t)
{
  New Array scale_notes [] = create_scale(Key_t);
  Chord_t tonic_chord= Create_root(scale_notes[0]);
  tonic_chord →Related_Chords[1…5] = Call
  majorKeyRelation(*Chord_t →Related_Chords,
  scale_notes[]);
  tonic_chord →Related_Chords[6…10] = Call
  minorKeyRelation((*Chord_t →Related_Chords,
  scale_notes[]);
  tonic_chord →Related_Chords[11…14] =
  create_relations(scale_notes[],Rule[1_6,1_4, 1_2, 1_9]);
}
Procedure majorKeyRelation(*Chord_t →Related_Chords)
{
  Return  create_relations(*Chord_t →Related_Chords,
  scale_notes[], Rule[1_7, 4, 5, 5_7, 6_m]);
}
Procedure minorKeyRelation(*Chord_t →Related_Chords)
{
  Return  create_relations(*Chord_t →Related_Chords,
  scale_notes[],Rule[6_m, 6_m7, 2_m, 3_m, 3_m7]);
}
Procedure creat_relation(*Chord_t →Related_Chords,
scale_notes[], rule[]);
{
  For(i=0;i<=length[rule];i++)
  {
     *Chord_t →Related_Chords[]=scale_notes[rule[i]];
  }
  Return *Chord_t →Related_Chords;
}
```

Fig. 6 Initiating Conceptual Relations among Chords

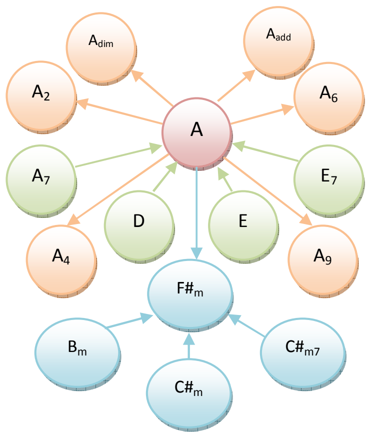

Fig. 7 Set of Family-Chords for Key_t in Amajor

A typical output of the algorithm will generate a set of chords associated with the `Key_t` in A_{major} is shown in Fig. 7. From this set of family-chords, it is easier to assign chords to the `Bag_t` of music score.

C. Chordset Formation from Knowledgebase

Now, the Knowledgebase (KB) is divided into two parts: PART-A: to store K-Ball of chords formed using algorithm in Fig. 6; and PART-B: to store the CG of the music scores.

In Part-A: Using the algorithm given in Fig. 6, the knowledgebase in MES is initiated with the chords in 12 majors and 12 minors totaling to more than 300+ chords, all represented as sets of K-Balls and KB is initiated for the MES.

In Part-B: The CG similar to that shown in Fig. 8, for a music score is stored. The `Bag_t` has a of `Note_t` that are structured and compartmentalized into equally weighted 'Bars'. It would be meaningful to term a Bar as a concept (K-Ball) than a single `note_t`. It's because Chords are usually formulated for Bars and not on notes.

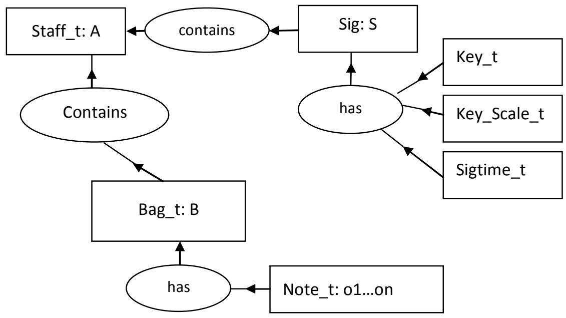

Fig. 8 Model of CG for a Music Score

The steps to process the K-Balls of Bars is shown below.

1. Initiate Part-A of KB with possible chords
2. Decode a music score to CG in Part: B of KB
3. Process the `Bag_t` in CG as follows:
   a. Pick Bar b_x in `Bag_t` BT
   b. Name notes in bx as $n_1, n_2 \ldots n_t$
   c. Find the Jusic Distance(JD) of each note n_i from the set of family chords
   d. Sum the occurrences of JDs for each n_i and choose the top ranked chords for the bar b_x
   e. The chords with maximum and equal JDs will be suggested for the Chord-Set.

Consider this bar b_x in Fig. 9.

Fig. 9 Sample Bar b_x of the music score

The Jusic Distance (JD) for bar b_x in Fig. 9 (within the rectangle) is given below:

JD(b1) = {(4,A), (4,E)}

TABLE II
CALCULATE THE JUSIC DISTANCE

Family Chords	n_1	n_2	n_3	n_4	n_5	n_6	1	3	5	7	3
A	3	5	5	6	7	1	1	1	2	0	4
D	2	2	2	3	4	5	0	1	1	0	2
E	1	1	1	2	3	4	3	1	0	0	4
E_7	1	1	1	2	3	4	3	1	0	0	0
$F^{\#}m$	7	7	7	1	2	3	1	1	0	3	0

The maximum number of occurrences of related notes for a Chord is summed up at the last column of Table II. We infer

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:1, 2014

from the JD(b_1) matrix, that 4 is the maximum value, and so A and E are the optimum Chords for the Bar b1. Though E7 and F#m has a maximum value, the 7th note for E7 is missing and 7th note occurrence for F#m is not required. Hence the value 0 for both the chords, and are not considered. The family Chords in a Key_t can be extended further to accommodate more choices, in applicable. Similarly, Jusic Distance is calculated for the subsequent bars in sequence and the relevant Chord Set are formed for each bar. Here, A and E chords are the formed Chord Set for bar b1, and the musician is given a choice to choose between these two chords in the Set.

D. Experiments on Chordset Formation

With the knowledgebase, rich in Chords and Music Scores, the above mentioned procedures and calculations, experiments with two music scores were conducted. The two music scores taken for this study are: "*Oh Love that will not let me go*", a hymn in the arrangement of David Phelps and "*El Shadai*" in the melody of Michael Card and arrangement of Simon Prakash. Let us rename the scores for easier identity. The first shall be (OLMG) and the later shall be (ELS).

The music scores are given as *.nwc file to the decoder, where the entire score is decoded and a relevant CG is constructed. This representation in the KB is taken for further processing. The details of the K-Balls generated are given in Table III. With this experimental setup, we executed the two music scores with the initiated knowledgebase. MES is able to present the Chord-Set for the scores, bar-by-bar. Sample output of 13 bars for OLMG is given in Fig. 10 (a). Sample output of 16 bars for ELS is given in Fig. 10 (b). The Bar progression is in the x-axis, and the y-axis takes the Chords from the Part-A of KB. Chords formed against each bar b_x is shown is colored boxes.

TABLE III
METRICS COLLECTED FROM CG CONSTRUCT

Metrics	OLMG	ELS
Staff_t	2	2
Note_t	275	346
Bars_t	46	62
No of links (bet Notes_t with Chord_t)	24	30
No. of Key Shifts	4	2
Avg No. of conceptual relations	14	17
Avg No of Chord Set per Bar_t	6	3

These outputs from MES shows that the Chord-Set formed based on the algorithms and JDs we have proposed, were most optimal – in the sense, the selection of Chords by three musicians showed correlation between MES and their expertise. Those results were not discussed in this paper due to space limits.

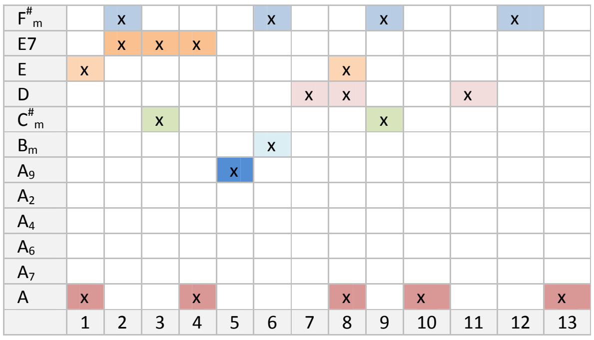

Fig. 10 (a) Sample output of 13 bars – OLMG

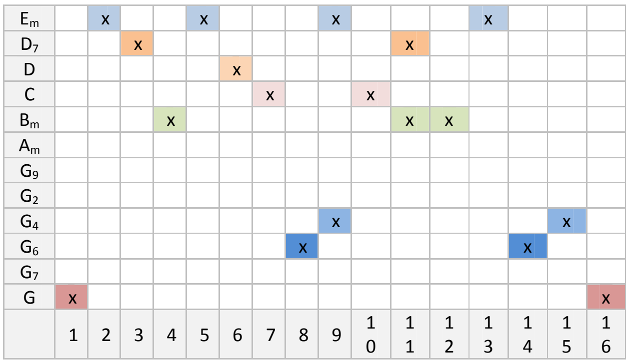

Fig. 10 (b) Sample output of 16 bars – ELS

The two music scores used for this study have been given choices for chords for the first 16 bars, and we have found these choices are appealing to musicians both theoretically and practically. However, when it comes to comparison between the existing text-based music software tools like note-worthy composer (NWC) and Cakewalk (CW), we could appreciate the performance of MES for its better throughput and response times. The same music scores are fed in NWC and CW, and its performance is evaluated as follows shown in Figs. 11 and 12.

Observing the experimental results from Figs. 11 and 12, it is well understood that the MES performance in terms of throughput and response time are quite significant in claiming that the knowledge models that we have proposed for the knowledgebase are better than some of the existing music processing tools. The transaction throughput is obviously high when compared the text based processing in NWC and CW.

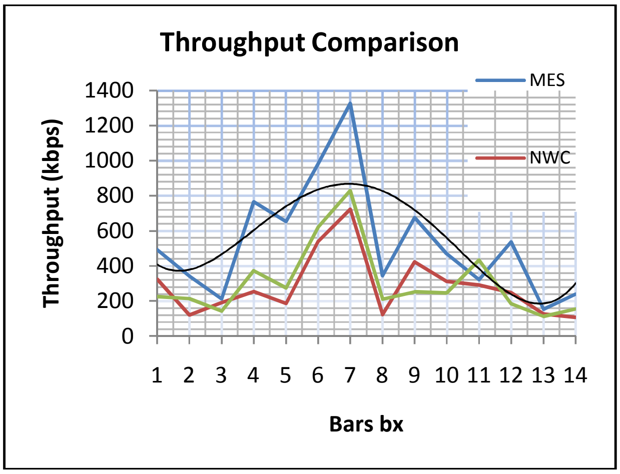

Fig 11 Throughput Comparison between music tools

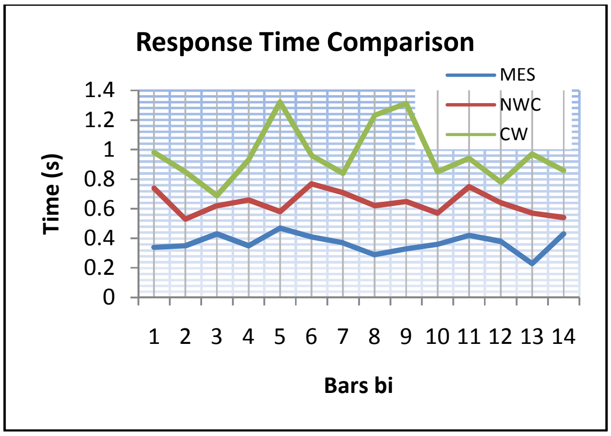

Fig 12 Response Comparison between music tools

The `bag_t` collection objects used in MES has increased the possibility of pre-fetching the entire collection of bars (bi), thus computing the chords for the bars; whereas, the other two tools don't have these collection objects and no intelligent processing of music notations. This will consequently reduce the response time of the KB of MES, compared to NWC and CW. Since the retrieval of notations is from the `Bag_t` collection, the response time in wasting for disk access is not required, thus better response time.

The knowledge representation models for musical notes and knowledgebase designed for this study is customizable and scalable as the research works extends to house more chords into the Part-A of knowledgebase and more music scores in the Part-B of knowledgebase. Also the CG constructs and the procedures to form the Chord-Sets are also extensible to house more CGs and improve efficiency as the research progresses.

We understand that the chords chosen for the study were just a few, a subset of many new chords available. Those chords could be considered for future works and more complexity added to the music score. This may lead to interesting options of chords derived and appended in the Chord-set for each music score.

Moreover, choosing optimal chords may also be extended to support parallel harmonic notations in line with melodic line. That is, a two part or three part harmony lines could be added to a melodic music score.

However this work lead to a discussion on the implications on how this framework on chord formation using knowledgebase could be deployed as a tool that would support musicians in choosing their preference of chords from the Chord-Set. Musical composition requires intelligent chord and sets of chord formation for better results [10].

V. Conclusions

The knowledge representation models for musical notes and knowledgebase designed for this study is customizable and scalable as the research works extends to house more chords into the Part-A of knowledgebase and more music scores in the Part-B of knowledgebase. Also the CG constructs and the procedures to form the Chord-Sets are also extensible to house more CGs and improve efficiency as the research progresses.

We understand that the chords chosen for the study were just a few, a subset of many new chords available [11]. Those chords could be considered for future works and more complexity added to the music score. This may lead to interesting options of chords derived and appended in the Chord-set for each music score.

Moreover, choosing optimal chords may also be extended to support parallel harmonic notations in line with melodic line. That is, a two part or three part harmony lines could be added to a melodic music score. These works are planned for future, which would add a little more contribution to the requirements of music art and creativity. Intelligence and smartness of a system compliments creativity of a music score presentation in more pleasing harmonic progressions.

Acknowledgments

The Authors wish to thank the Musicians and Artists for giving critical comments and reviews on this work. Also the anonymous reviewers are gratefully acknowledged. The authors also thank the management of Entapp solutions for providing their multimedia lab for conducting the experiments.

References

[1] Alan Belkin, "A Practical Guide to Musical Composition", 2008 (url: http://franklin.duxbury.k12.ma.us/Music Downloads/Documents And Parts/Curriculum/Current Curriculum/Electives/Electives UBD/Music Theory/Music Theory Class Structure/Level2/10.Form/FullForm.pdf) accessed Feb 2013)

[2] Theory. "Key Signatures and the Circle of Fifths," ImprovHQ. http://www.improvhq.com/?id=keysignatures (accessed January, 2013).

[3] Kevin Meixner, "The Basics of Reading Music", (url: http://readsheetmusic.info/readingmusic.shtml)

[4] Turek, Ralph. Theory for Today's Musician, New York: McGraw Hill, 2007.

[5] Winston Vaughn Arblaster, "Music Theory and Arranging Techniques for Church Musician", Master's Thesis, University of Oregon, September 2010.

[6] John F. Sowa, Conceptual Graphs, Draft Proposed ISO Standard for Conceptual Graphs, 2001. http://www.jfsowa.com/cgcgstand.htm.

[7] Justus S., Iyakutti. K., "Measuring the Storage and Retrieval of Knowledge Units: An Empirical Study", Contemporary Computing,

Communications in Computer and Information Sciences, Vol. 40, pp. 358-369, 2009.
[8] Justus S., Iyakutti. K., "An Empirical Validation of the Suite of Metrics for Object-Relational Data Modelling", Intl. J. of Intelligent Information and Database Systems, vol. 5, No. 1, pp/ 49-80, 2011.
[9] http://music.shanemcdonald.org/chordmap/.
[10] A. Gartland-Jones and P. Copley, “The Suitability of Genetic Algorithms for Musical Composition”, Contemporary Music Review, 22(3):43–55, 2003.
[11] Clendinning, Jane Piper and Elizabeth West Marvin, “The Musician's Guide to Theory and Analysis”, New York: W.W. Norton &Company, Inc., 2005.

V. Hepsiba Mabel is a PhD research scholar at the Department of Computer Science, Madurai Kamaraj University, Madurai, TN, India. Her areas of research include Knowledge Engineering, Software Design Models and Interactive Multimedia.

Prior her fulltime research, she was working as Assistant Professor in a constituent college of the same University. She has publications in the areas of process improvement in software design and development and in Knowledge Management in reputed conferences and journals. Hepsiba is a member of ISTE and IAENG.

Dr. K. Alagarsamy is a Doctorate in Computer Science from Madurai Kamaraj University, Madurai, TN, India. Her areas of research include Knowledge Engineering, Knowledge Management, Process Improvement and software engineering

He is full-time Professor in the Department of Computer Science of the same university. He has several publications in the areas of process improvement in software design and development in Knowledge based systems in reputed conferences and journals. He is a member of board of studies in various colleges and technical universities.

Dr. S. Justus received his doctorate degree from Madurai Kamaraj University, Madurai, India. His research specializations include Object-relational data modeling, knowledge engineering and Big Data.

He has been into academic research and has published several of his research work results in International Journals and Conferences – including SwSTE in Israel and DASMA in Germany. He has also practitioner’s experience while working with Software development companies in the fields on performance evaluation and validation.

He is a member of IEEE, ISTE, IAENG and ACM professional associations. He has served as research & project coordinator for PG studies at Engineering Institutes. Presently he is working as Associate Professor at VIT University, Chennai.