Journal of Engineering and Applied Sciences 13 (14): 5814-5819, 2018
ISSN: 1816-949X

Preliminary Results on the Implementation of Knowledge based Expert System

V. Hepsiba Mabel and Justus Selwyn
School of Computing Sciences and Engineering, VIT University, Chennai, India

Abstarct: To present the implementation and its preliminary results of the knowledgebase for music expert system. The knowledgebase is divided into two parts where one will have all the chords and the other will have the input music score stored as conceptual graphs. The processes involved in deriving a set of music chords for the score is presented. The implementation of Jusic Distance algorithm and the empirical results are explained with necessary screenshots and resulting chordset. Musicians and composers will be the beneficiaries of this system. Extended experiments were conducted with two mega music scores and their results are also presented with finer details. The future direction of these results is to build an exclusive system, a graph database storage for the music scores, so that, a music score can be retrieved along with their set of chords. Additional staffs in the music score may be included along with improvement in performance of the knowledgebase.

Key words: Knowledge management systems, knowledgebase, expert system, music, necessary screenshots, database storage

INTRODUCTION

With the extension of our earlier works in building a knowledgebase for Music Expert System (MES), we have obtained some preliminary results during the implementation process.

The Knowledge Base (KB) in music expert system is divided into UNIT-A which has Knowledge Units (KU) of the chords and UNIT-B has the Conceptual Graph (CG) (Sowa, 2001) of the music scores. UNIT-A is populated with 5 sharp-majors and minors and 5 flat-majors and minors totaling to more than 250 chords, all represented as knowledge units connected conceptually (Meixner, 2012). UNIT-B has the CG for a piece of music that will be the input to the MES for processing.

MATERIALS AND METHODS

MES architecture: The MES architecture is simplified in Fig. 1. It has a preprocessing unit where the input music score is in the format of MIDI type-0. Then the part A of the KB is populated with the relevant major chord and its family chords. The input MIDI file is converted to a conceptual graph with a bag-t (Justus and Iyakutti, 2010) of collection of notes, durations, rests, bars and additional music information.

In this study, we have implemented this MES architecture with a simple music score and a simple set of chords. Though, UNIT-A is complete with majors and minors and its chords, the UNIT-B was not a complete CG.

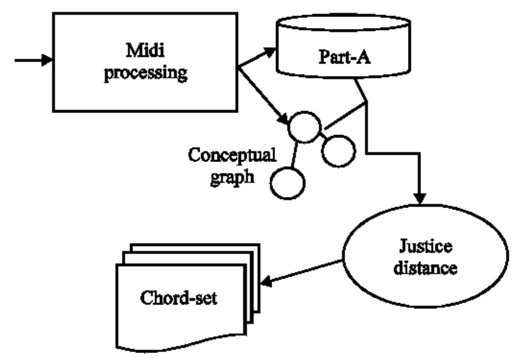

Fig. 1: MES architecture

Jusic distance algorithm is proposed and implemented which gives a set of chords from which a musician can pick the more apt one.

Populating family chords: To begin with the research, first the knowledgebase has to be initiated with a collection of Family Chords for a given Major/Minor μ. Family Chords (RCCC., 2001) are derived as:

$$\mu(\text{family\_Chords}) = \mu+\Delta\mu$$

Where:
Δ = 4 V 5 V 6 m V 2 m V 3 m V 3 m 7 V 3
'm' = The 'minor chords'

For example if μ = D major, then its family chords are derived as shown in Fig. 2.

Corresponding Author: V. Hepsiba Mabel, School of Computing Sciences and Engineering, VIT University, Chennai, India

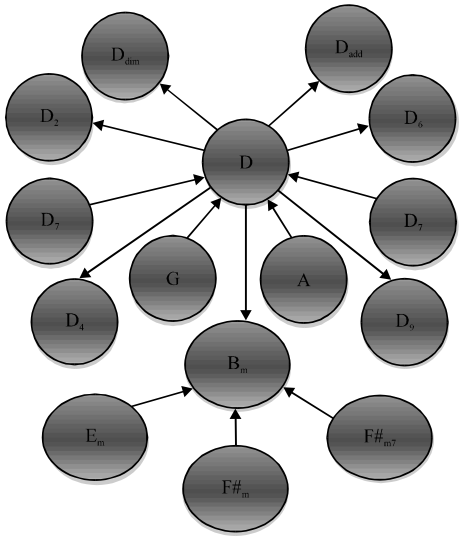

Fig. 2: Family chord-set for D-major

s_id	name	scale	related_chords	no_of_sharps	no_of_flats
1	C	C-D-E-F-G-A-B	C-F-G-G7-Am-Dm-Em-Em7-E-E7	0	0
2	D	D-E-F#-G-A-B-C#	D-G-A-A7-Bm-Em-F#m-F#m7-F#-F#7	2	0
3	E	E-F#-G#-A-B-C#-D#	E-A-B-B7-C#m-F#m-G#m-G#m7-G#-G#7	4	0
4	F	F-G-A-Bb-C-D-E	F-Bb-C-C7-Dm-Gm-Am-Am7-A-A7	0	1
5	G	G-A-B-C-D-E-F#	G-C-D-D-Em-Am-Bm-Bm7-B-B7	1	0
6	A	A-B-C#-D-E-F#-G#	A-D-E-E7-F#m-Bm-C#m-C#m7-C#-C#7	3	0
7	B	B-C#-D#-E-F#-G#-A#	B-E-F#-G#m-C#m-D#m-D#m7-D#-D#7	5	0
8	Am	A-B-C-D-E-F-G	Am-Dm-Em-Em7-E-E7-C-F-G-G7	0	0
9	Bm	B-C#-D-E-F#-G-A	Bm-Em-F#m-F#m7-F#-F#7-D-G-A-A7	2	0
10	C#m	C#-D#-E-F#-G#-A#-C	C#m-F#m-G#m-G#m7-G#-G#7-E-A-B-B7	5	0
11	Dm	D-E-F-G-A-Bb-C	Dm-Gm-Am-Am7-A7-F-Bb-C-C7	0	1
12	Em	E-F#-G-A-B-C-D	Em-Am-Bm-Bm7-B-B7-G-C-D-D7	1	0
13	F#m	F#-G#-A-B-C#-D-E	F#m-Bm-C#m-C#m7-C#-C#7-A-D-E-E7	3	0
14	G#m	G#-A#-B-C#-D#-E-F#	G#m-C#m-D#m-D#m7-D#-D#7-B-E-F#-F#7	5	0

Fig. 3: UNIT-A of the knowledgebase

Optimal chord formation: The Knowledge Base (KB) is divided in two UNITS: UNIT-A: which has the chords and its family chords both major and minor chords and PART-B which has the graph of the melody line of the music.

In UNIT-A, exploitation the algorithmic program given in Fig. 3, the KB is populated with the chords in 20+majors and minors with 200+chords (RCCC., 2001), drawn as collection of KUs and computer memory unit is populated for the MES. In UNIT-B is a collection of structured objects defined as Note-t that are placed into equal measures of 'Bars'. Every note is processed in Jusic Distance algorithm.

RESULTS AND DISCUSSION

Implementing the Jusic Distance algorithm: The Jusic Distance (JD) of a every notation N is defined as the number of hops of the given note N from the root note R_N.

Fig. 4: The Justice Distance (JD)

Table 1: Sample bar JD calculation

Chords	nt_1	nt_2	nt_3	nt_4	nt_5	nt_6	cm1	cm3	cm5	cm7	Σ
A	5	5	5	6	7	1	1	0	3	1	**5**
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
Bm	4	4	4	5	6	7	0	0	0	1	1
C#m	3	3	3	4	5	6	0	3	1	0	4

nt = note, cm = chord-member, # = sharp (musical term), E_7 = E-seventh, F#m = F-sharp-minor, Bm = B-minor, C#m = C-sharp-minor

Algorithm 1; Implementing the Jusic Distance algorithm:

```
 Procedure Jusic_Distance
 Input: N, R_N
 Output: Jusic-Distance
 BEGIN
      JD = 1
      BEGINE Loop Until JD<8
         IF (N = = RN-Scale-List[JD] THEN
                         Jusic-Distance = JD
                        BREAK
  ELSE
      Increment JD One Step
END Loop
 END
```

The musical notations are processed as bar-wise as follows:

- UNIT-A populated with all chords
- Conver a music piece into readable file and create objects for those notations in UNIT-B
- UNIT-B is processed as following these steps
  - Consider a bar Bx
  - Notes in Bx are read independently
  - JD is calculated using the algorithm
  - JD for every note is calculated and ranked in descending order
  - The top two/three ranked chords are given as suggestions

For example take this highlighted portion b1 Fig. 4. The JD for the notations in the highlighted bar b1 is given in Table 1.

The occurances of JD value 1, 3, 5, 7 and the sum of these occurances are attached to the set of chords. The chord that has the highest rank is the suggested chord for a given bar Bx. Table 1 gives the calculated JD values for each note in b1 and two chords, A chord with 5 rank and E chord with 4 rank are suggested for the below bar. Each note in the score is denoted as nt and the chord-members of a triad is denoted as cm.

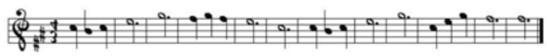

Fig. 5: Working music score

Table 2: Jusic distance calculation for bars 1-3

Family chords	B1 nt1	B1 nt2	B1 nt3	B1 JD	B2 nt1	B2 JD	B3 nt1	B3 JD
D	5	6	5	**2**	3	**1**	1	**1**
G	2	3	2	1	7	0	5	1
A	1	2	1	**2**	6	0	4	0
A7	1	2	1	**2**	6	0	4	0
Bm	7	1	7	1	5	**1**	3	**1**
Em	4	5	4	1	2	0	7	0
F#m	3	4	3	2	1	**1**	6	0
F#m7	3	4	3	2	1	1	6	0
F#	0	4	0	0	1	1	6	0
F#7	0	4	0	0	1	0	6	0
	JD(b1) = {(2, D|A|A7)}				JD(b2) = {(1, D|F#m)}		JD(b3) = {(1,D|Bm)}	

nt = note, B1 = bar1, B2 = bar2, B3 = bar3, JD = Jusic Distance, # = Sharp (musical term), D-DF#A, G-GBD, A, AC#E, A7-AC#EG, Bm-BDF#, Em-EGB, F#m-F#AC#, F#m7-F#AC#E, F#-F#A#C#, F#7-F#A#C#E

Table 3: Jusic distance calculation for bars 4-6

Family Chords	B4 nt1	B4 nt2	B4 nt3	B4 JD	B5 nt1	B5 JD	B6 nt1	B6 JD
D	2	1	2	1	3	1	5	**1**
G	6	5	6	1	7	0	2	0
A	5	4	5	**2**	6	0	1	**1**
A7	5	4	5	**2**	6	0	1	1
Bm	4	3	4	1	5	**1**	7	0
Em	1	7	1	2	2	0	4	0
F#m	7	6	7	0	1	**1**	3	1
F#m7	7	6	7	2	1	1	3	1
F#	7	6	7	0	1	1	0	0
F#7	7	6	7	2	1	1	0	0
	JD(b4) = {(2, A|Em)}				JD(b5) = {(1, Bm|F#m)}		JD(b6) = {(1, D|A)}	

nt = note, B4 = bar4, B5 = bar5, B6 = bar6, JD = Jusic Distance, # = Sharp (musical term), D-DF#A, G-GBD, A, AC#E, A7-AC#EG, Bm-BDF#, Em-EGB, F#m-F#AC#, F#m7-F#AC#E, F#-F#A#C#, F#7-F#A#C#E

The value 0 for some of the chords are not considered because the given note in the given major/minor does not fall into that chord's category. Likewise, Jusic Distances are calculated for the subsequent bars in the order of their presence in the music score and also the optimal chord is derived for each note in a bar.

Consider the sample music score in Fig. 5. Applying the above stated Jusic Distance Algorithm, the results are given in Table 2 and 3. Each note is denoted as nt, each bar in the score is denoted as B_i where i = 1, 2, 3, 4, 5, 6 and JD is the calculated Jusic Distance.

Empirical validation of jusic distance: Having the chords and the scores on the Part-A of the KB, the given procedures and calculations are validated empirically with one sample music score. The music score given as input is shown in Fig. 6 and its relevant txt format for parsing is shown in Algorithm 2.

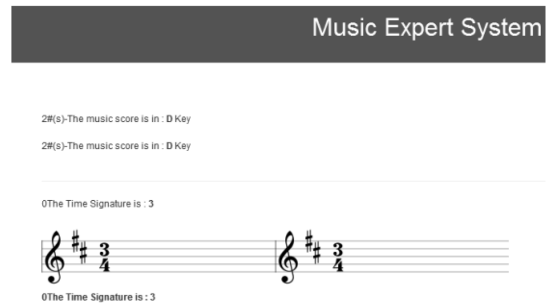

Fig. 6: Header extraction for the score

Algorithm 2; Converted music score in txt format:

<Absolute>0</Absolute>
<NoteOn channel = "1" Note = "67" velocity = "110"/>
</Event>
<Event>
<Absolute>160</Absolute>
<Note on channel = "1" Note = "67"> velocity = "0"/>
</Event>
<Event>
<Absolute>192<Absolute>
<NoteOn channel = "1" Note = "67" velocity = "110"/>
</Event>
<Event>
<Absolute>352<Absolute>
<Note on channel = "1" Note = "67"> velocity = "0"/>
</Event>
<Event>
<Absolute>384<Absolute>
<Note on channel = "1" Note = "74"> velocity = "110"/>
</Event>
<Event>
<Absolute>544<Absolute>
<Note on channel = "1" Note = "74"> velocity = "0"/>
</Event>
<Event>
<Absolute>576<Absolute>
<Note on channel = "1" Note = "74"> velocity = "110"/>
</Event>
<Event>
<Absolute>832<Absolute>
<Note on channel = "1" Note = "71"> velocity = "0"/>
</Event>

Figure 3 gives the initiated family chords in the knowledgebase and the corresponding scale for each major. The content of the UNIT-A will be collection of chords related to the key-signature of the given music score. In this example, the D major and its relevant chords are taken from the UNIT-A of the knowledgebase.

The Jusic Distance algorithm presented in study is applied for the music score given in Fig. 4 is implemented in PHP (WAMP). The MES sequence of operation is as follows:

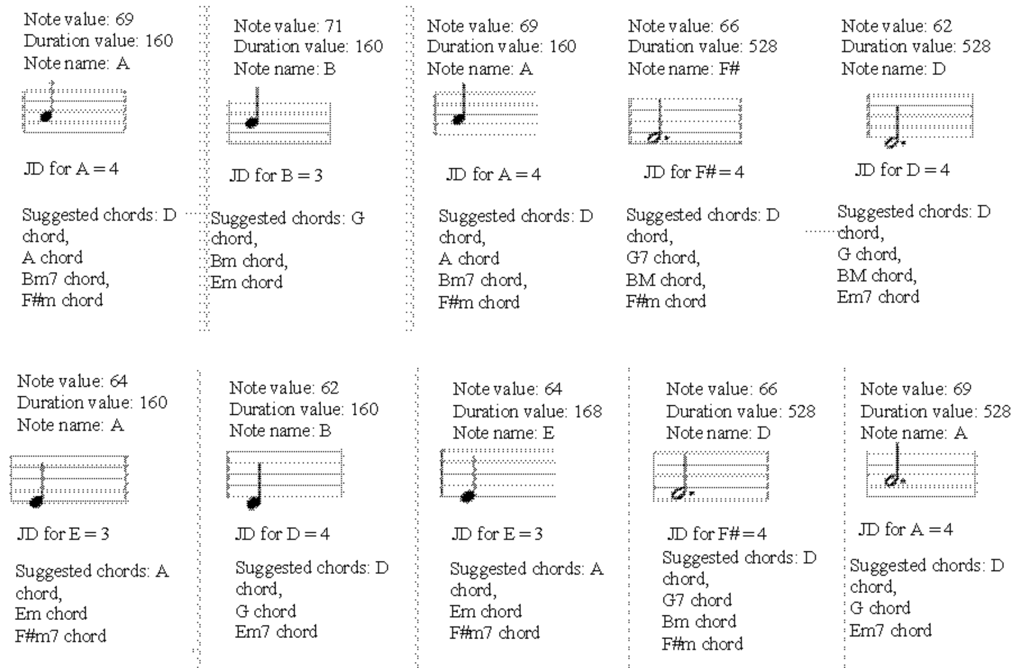

Fig. 7: Chord-set for the score

Algorithm 3; Jusic Distance algorithm:

Input: a midi file of the music score

Step 1: The midi file is converted to XML format for easy processing

Step 2: The XML file is preprocessed to remove the header information like name-of-the-clef, key-signature, time-signature. Once the key-signature is determined, the Key's corresponding chords are fetched as single object from the KB

Step 3: In the XML file, every note is represented as an XML element with two entries. One entry gives the start-time of the note (mentioned as 0) and the second entry gives the time duration of how long the note will be played. The difference between these values will help us determine the value of the note (whole, half, quarter, etc.)

Step 4: The name of the note is determined by the <NoteOn Note> value These are internationally standard values, hence mapping these values to the musical note will give the name of the note

Step 5: Jusic Distance (JD) is calculated for every note. The distance value determines the order in which the chords should be preferred

Step 6: Every note is processed bar-wise based on the time-signature and a set of chords are suggested for every note

Output: Chord-set for every note

In our empirical study, the first 6 bars of the music score is presented here for understanding the working of the MES. These optimal sets of chords are given as suggestion to the musicians or composers for finding the completeness in their music works. These chord suggestions will assist them in the improvisations and orchestrations of the music score in the future.

In Fig. 7, we have given the output for the first 6 bars of the music score given in Fig. 4. The header information, note value, note name are all extracted by processing the XML file and the relevant chord set is derived by using the Jusic Distance procedure given in study C.

Table 4: Metrics collected from CG construct

Metrics	YMA	ITY
Stave	1	1
Notations	465	620
No. of bars	76	120
No of relations bet notes and chords	42	68
Key transpose	3	3
CR average	24	47
Chords/bar average	2	4

YMA = You are My All in all, ITY = I Thank You

Extended experiments and results: Both the chords and family chords and the music scores that are given as input are stored in the knowledgebase and using these two parts of the KB, these experiments were conducted. Among a variety of music scores, two we choses for this study: "You are my all in all", cannon in D arrangement and "I Thank You" in the melody and arrangement of Solomon Rajkumar. The scores are renamed as (YMA) for the first score and (ITY) for the second score.

The treble clef of the music score are given as MIDI type-0 files to the preprocessor where the MIDI is converted to XML format and parsed. Individual notes are extracted and a CG is built. Table 4 presents the details of the K-units generated through these experiments. Two music scores are executed with the initial set of chords in the KB.

MES is able to derive a set of chords for the music, note-by-note. Figure 8a gives the output of 13 bars for YMA and in Fig. 8b gives the output for 16 bars for ITY. The x-axis has the bar progression and y-axis has the

X Family
X Related minor
X Additionals

(a)

	1	2	3	4	5	6	7	8	9	10	11	12	13
$F^{\#}_{m7}$		x				x			x			x	
$F^{\#}_{m}$		x	x	x									
$F^{\#}_{7}$	x							x					
$F^{\#}$							x	x			x		
G_m			x						x				
B_m						x							
D_9					x								
D_2													
D_4													
D_6													
D_7													
D	x			x				x		x			x

(b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
E_m		x			x				x				x			
D_7			x								x					
D						x										
C							x			x						
B_m				x							x	x				
A_m																
G_9																
G_2																
G_4									x						x	
G_6								x						x		
G_7																
G	x															x

Fig. 8: a) Chords fr first 13 bars-YMA and b) Chords for first 16 bars-ITY

family chords from UNIT-A of KB. Colored boxes are the preferred set of chords for the given music scores.

We showed these chord outputs to three experts in music composition. The results MES gave are close to the experts' choices in the selection of chords. What is manually and artistically done is now achievable by a knowledgebase driven music expert system.

The discussions and the reports of the experimental study are not presented in this study but can be explained in another work of relevance. Though this is an experiment conducted with two music scores, the results of the chords derived would be preferred by the artists and composers. The input files of these music scores are given as a single treble clef staff. The bass clef is not taken because that is re-written based on the chords that are derived for the melody in the treble clef. The implementation of MES is done in WAMP as server PHP and MySQL DB.

We also evaluated the performance of MES in terms of throughput and response time. The performance of our MES in terms of throughput and response time is compared with existing softwares like note-worthy and cake-walk application tools. The comparisons of the performances of the music tools are shown in Fig. 9 and 10.

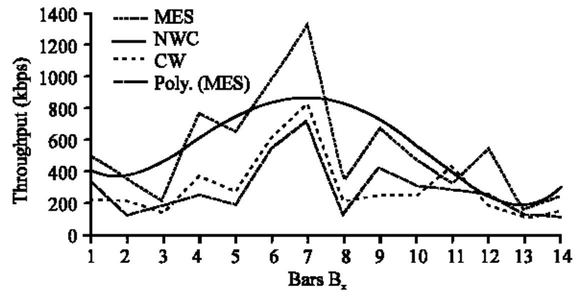

Fig. 9: Comparison of throughput

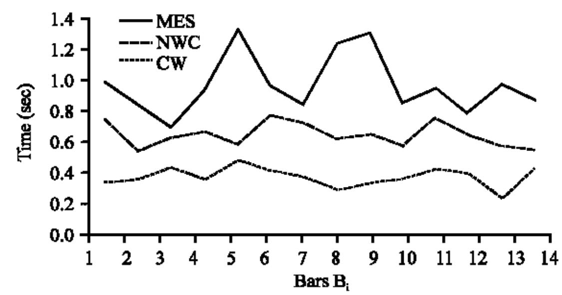

Fig. 10: Comparison of response times

The experimental results for through shown in Fig. 9, depicts a high of 1360 bytes per disc-read of a knowledge unit. Usage of prefetching the knowledgunits of each chord has increased the throughput of MES when compared to the existing music processing tools that are good in chord suggestions.

The collection type bag-t (Mariano *et al*., 2016) used in MES has the concept of pre-fetching the chords for the entire bar bi in one read operation. The other two music processing tools are working based on text processing of notations. This intelligent processing of notations in MES using pre-fetching of knowledge units has led to a very minimal response time.

The knowledge units that are modelled using object-structured and conceptual structured (Mabel and Justus, 2016) are scalable and customizable to process larger music scores in UNIT-B that will always refer to te UNIT-A of the knowledgbase.

The chord-sets that are used in this preliminary study are less when compared to the entire range of chords. The music scores that, we considered are also simple ones. However, the knowledge models (Mabel and Justus, 2016; Rao and Nayak, 2017) have more room for increasing the chord-sets and include more complicated music scores.

Currently, we are working on enhancing the Music Expert System (MES) with additional features for processing an analog MP3 file to read the frequencies and later map them to the music alphabets. Once this is possible done, then the musicians, learners, beginers can use MES to process their music scores, derive chord-set sugestions, harmonize, wrtie orchestrations for thier melody and bring out their music works.

The knowledge based models used in MES is quite simple and shows very significant results in this preliminary work. The future directions would be to include more artistic, intelligent storage and retreival of music notations and chords in terms of knowledge units/objects.

CONCLUSION

The music models used in knowledge representation proposed in this research are scalable and customizable because the main work of this research is seeking to extend to include more chords into the UNIT-A of knowledgebase and storage of music scores graphs in the UNIT-B of knowledgebase. The stored conceptual graphs should be accessible by Cypher queries which is accompanied by triggers and procedures (Chung *et al*., 2016).

Instead of one staff with a melody line, more number of staffs with bass clef, tenor clef and alto clef may be included in the MES in the future. This will add value to the research, so that, complicated music scores from composers can be processed to form a set of chords for complete orchestration that will be of rich artistic beauty.

These works on MES can also be made as distributed software product where musicians, researchers and students can make use of the features of MES for professional works or for their own research works where the outputs will be of help in their works too.

REFERENCES

Chung, K., R. Boutaba and S. Hariri, 2016. Knowledge based decision support system. Inf. Technol. Manage., 17: 1-3.

Justus, S. and K. Iyakutti, 2010. An empirical validation of the suite of metrics for object-relational data modelling. Intl. J. Intell. Inf. Database Syst., 5: 49-80.

Mabel, H. and S. Justus, 2016. Performance assessment of 3-layered knowledgebase and representation models: An empirical study. Intl. J. Knowl. Eng. Data Min., 3: 239-254.

Mariano, S., S. Mariano, Y. Awazu and Y. Awazu, 2016. Artifacts in knowledge management research: A systematic literature review and future research directions. J. Knowl. Manage., 20: 1333-1352.

Meixner, K., 2012. The basics of reading music. Institute Sheet Music, Vaasa, Finland.

RCCC., 2001. Key signatures and the circle of fifths. RubyCliff Computing Company, JULIAN, California.

Rao, S.S. and A. Nayak, 2017. Enterprise ontology model for tacit knowledge externalization in socio-technical enterprises. Interdiscip. J. Inf. Knowedge Manage., 12: 99-124.

Sowa, J.F., 2001. Conceptual graphs, draft proposed ISO standard for conceptual graphs. ISO., Geneva, Switzerland.