

RESEARCH

Open Access



iSargam: music notation representation for Indian Carnatic music

Stanly Mammen^{1*} , Ilango Krishnamurthi¹, A. Jalaja Varma² and G. Sujatha³

Abstract

Indian classical music, including its two varieties, Carnatic and Hindustani music, has a rich music tradition and enjoys a wide audience from various parts of the world. The Carnatic music which is more popular in South India still continues to be uninfluenced by other music traditions and is one of the purest forms of Indian music. Like other music traditions, Carnatic music also has developed its musicography, out of which, a notation system called Sargam is most commonly practiced. This paper deals with development of a music representation or encoding system for the Sargam notation scheme which enables easy music notation storage, publishing, and retrieval using computers. This work follows a novel idea of developing a Unicode-based encoding logic and allows storage and easy retrieval of music notation files in a computer. As opposed to many existing music representation systems for western music notation, iSargam is the only music notation encoding system developed for Indian music notation.

Keywords: Music representation, Music encoding, Music processing, Music information retrieval, Indian music, Computer music

1 Introduction

Music information representation systems, in general, encompass musical information contained in music notation or recordings. Since the whole purpose of music representation is computer analysis, retrieval, or synthesis, most of the representation systems focused on representation of musical notation, particularly, the Common Western Notation (CWN), because of its effectiveness in representing discrete elements of music. Thus, many systems like ENP [1], LilyPond [2], Humdrum [3], Guido [4], and Cadenza [5] used ASCII-based textual descriptions of western music notations. Similarly, music representations for non-CWN systems like Gregorian chant, Django for tabulature, GOODFEE for Braille notation etc. also developed [1]. An example of a Gregorian chant is given in Fig. 1 [6]. Another recent trend was the development of Web-viewable notation applications such as Scorch by Sibelius software and ScoreSVG [7].

The textual descriptions of music notations can be further classified as record-based, command-based, symbolic codes, and LISP-based. While systems like DARMS [8],

Guido [4] etc. used symbolic codes, systems like CMN [9] used command-based representations. A few examples of LISP-based representations are MUZACS [10], Rhythm-Editor of Patch Work [11], and CMN [9]. Also, to take advantage of XML features like structuring and portability, many popular XML-based notations like MusicXML [12], MEI [13] [14], and WEDELMUSIC [15] emerged.

2 Background and history

Indian classical music is one of the oldest music traditions in the world, and it enjoys the next position to western music in its popularity. The music tradition of India can be divided into two large traditions, namely, Hindustani and Carnatic music. The Hindustani classical music which is predominant in the northern part of the Indian subcontinent, originates from the ancient Vedic, Persian, and many folk traditions. While the Carnatic classical music, uninfluenced by non-Indian music traditions, is the purest form of Indian music and is prevalent in the southern parts of the Indian subcontinent. It is generally homophonic in nature with emphasis on vocal music. If performed on an instrument, it assumes a singing style.

Carnatic music is usually performed by an ensemble of musicians consisting of a principal performer, usually a

* Correspondence: stanly.mammen@gmail.com

¹Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India
Full list of author information is available at the end of the article



Fig. 1 Example of a Gregorian chant, an excerpt from "Antiphonale Monasticum" by Solesmes, licensed under Creative Commons [36]

vocalist, accompanied by a rhythm instrument, melodic instrument, and a monophonic drone instrument.

The tradition of using music notations for singing was practiced from Vedic times. But there was no uniformity in the notation system used from time to time. However, all the notations developed so far can be classified as script as opposed to staff notation used for western music. Different systems of notation were prevalent in each period of history in both Carnatic and Hindustani traditions. Also, many notation systems existed in parallel.

In regard to Carnatic music, there were two popular notation systems, viz, the old notation and modern notation. The old notation, as the name implies, can be found in many ancient books, including the "Sangita Sampradaya Pradishini" [16] written by the late Subbarama Diksitar, a popular music theorist. Currently, a new music notation system, namely, modern notation or Sargam notation is widely used for notating Carnatic music in various books, reviews, and in academia. The Sargam notation system uses a subset of symbols used in old notation. A Carnatic music composition notated using Sargam notation is given in Fig. 2.

The main objective of this work is to develop a unified representation system for storing Indian Carnatic music notations in computer files. The system aims at encoding music notation symbols and other associated information like title, composer name etc., while ignoring the layout specific details. Most of the encoding systems

developed so far are intended at encoding western music staff notation or a derivate of it. This work aims at developing a machine readable music notation system for Carnatic music which can support playback, printing, retrieval, and searching within the composition.

Since Unicode is currently supported by most text editors and web browsers, the iSargam file can be read directly and displayed without the need for any additional code similar to the ASCII-based text files. Additionally, they can employ plug-ins to provide different styles and layouts for printing purposes. Since most of the current programming languages support Unicode, application programmers and music analysts can easily build algorithms for the music information retrieval or computer-based analysis of the iSargam music databases.

The next section gives details about Sargam notation system, and the section following provides details of related works done elsewhere. We then describe the iSargam encoding system, explaining its approach and encoding algorithm. For increasing readability of western readers, we give comparisons with western music concepts wherever applicable. We also present some example encoding compositions as proof of our approach.

3 The Sargam notation system

Sargam notation is a music notation language for Carnatic music. Each notation starts with specification of raga, tala, and mela. Sometimes, the notes used in ascending scale and descending scale, known as arohana and avarohana, are explicitly defined in the start of the composition. This is followed by the time signature and actual music notation following it as illustrated in Fig. 2. In this section, we attempt to describe the terminologies used and the notation scheme.

3.1 Raga

Raga [17] is one of the most distinguished features of Carnatic music. The raga can be defined by a melodic scheme characterized by a definite scale or notes, order or sequence in which the notes can be used, melodic features, pauses and stresses, and tonal graces. Some ragas define the same set of music notes (swara) but are still differentiated by some other features like the order of appearance of swara, melodic punctuation, accent, intonation, and melodic phrases. The raga in Carnatic music is analogous to key signatures in western music.

| | | | |
|---|---|---------------------|---|
| JATISWARAM Raga – Sankarabharana Tala – Rupaka | | | |
| Mela 29. { s r g m p d n s s n d p m g r s | | | |
| O | I ₄ | o | I ₄ |
| Pallavi Ś; P; | ; R Ś N ; Ś DP | DP MG | ; MGM Rg m P d n (Ś;;) |
| Anupallavi P; S; | ; dp M p m ; sn S gr | G m g G m g | Rgr Ssn M p m P d n (Ś;;) |
| Charanam 1 M ; M ; M g s | ; p m g s r g ; p m ś N d p r s M g . s p p | M; M; M g - d | ; p d p p m g ; g r g s r g p p m g m p d n (Ś;;) |

Fig. 2 An excerpt from "Jatiswaram," notated in Sargam

A related field is the specification of arohana and avarohana. The arohana (meaning “ascending”) follows from the raga specification and explicitly lists the set of allowed swara syllables when the music is following an ascending flow. Similarly, the avarohana (meaning “descending”) specifies the name of the parent raga.

3.2 Tala

The term “tala” [17] refers to the rhythm system which controls and establishes the music. There are hundreds of defined rhythm styles (talas) in Indian music. The name of the tala used in the notated composition is given above the notation as in Fig. 2.

Now, we attempt to describe the tala system explaining its elements and symbols. Each rhythm pattern or tala is derived as a combination of six basic elements called “angas,” each measured with a unit known as “aksharakala.” The name, notation, and duration measurement of the six basic elements are given in Table 1. This is similar to various note types like crotchets and semiquavers in western music notation. Also, each element (anga) has a defined reckoning mode. The reckoning mode specifies the actual delivery of the rhythm.

The rhythm pattern is repeated in a cyclic manner throughout the music and hence is known as “avarta,” which means repetition. The basic rhythm used is notated at the beginning of the notation, and notes are grouped according to it as seen in Fig. 2. The grouping method used is similar to grouping notes according to time signature as in western music notation but with bars of different measure.

3.3 Mela

Mela or Melakartas [17] are parent ragas from which the other ragas evolved. There are twenty-two of them. A distinguished feature of the mela or parent ragas is that they contain all the seven notes in order. The melakartas have a numbering scheme and are identified by the number. The parent raga of the composition is so indicated by an integer number as illustrated in Fig. 2.

3.4 Notation

The notation style is primarily script-based where the note symbols are placed on a straight line. Suitable signs

Table 1 Basic elements of Indian Carnatic music rhythms (tala)

| Element | Symbol | Duration in aksharakala units |
|-----------|---|-------------------------------|
| Anudrutam | ˘ | 1 |
| Drutam | ◦ | 2 |
| Laghu | ₃ , ₄ , ₅ , ₆ , ₇ , ₉ | 3,4,5,6,7,9 respectively |
| Guru | 8 | 8 |
| Plutam | 8̇ | 12 |
| Kakapadam | + | 16 |

and symbols are also used to indicate various other musical features. A detailed review of the Sargam transliteration scheme can be found in [18].

As shown in Fig. 2, the notation part starts with tala symbols called “anga” which group the other music notes according to time duration. The symbols used in notation can be classified as music note (swara), gamaka symbols, and other symbols. The following section details the concept and notation of musical note, gamaka, and other symbols.

3.4.1 Musical note

A swara or music note usually denotes the note name indicating the pitch, duration, octave, and whether it is played with expressions (called gamaka) or not. The notes are named differently according to its pitch as shadja (sa), rishabha (ri), gandhara (ga), madhyama (ma), panchama (pa), dhaivata (dha), and nishada (ni) and is abbreviated as given in brackets. The current style of written forms is with vowels removed and expressed with single letters as shadja (S or s), rishabha (R or r), gandhara (G or g), madhyama (M or m), panchama (P or p), dhaivata (D or d), and nishada (N or n).

3.4.2 Octave or sthayi

In Indian music, there are five referred octaves, with a middle octave and two upper and two lower octaves. The first upper octave is denoted by adding a dot below the music note and the second upper octave denoted by adding two dots below the music note as shown in Fig. 3c. Similarly, the immediate lower octave is denoted by adding a dot above the music note and the next lower octave is denoted by adding two dots above the music note as shown in Fig. 3b.

3.4.3 Duration

The duration of each music note is measured in terms of aksharakala, which is the unit of measurement in the Indian tala (rhythm) system and is analogous to “beats” in western music. To denote the duration of a music note, uppercase or lowercase letters with or without comma, semicolon, or with underline or over line are used. A swara letter in lowercase indicates one aksharakala duration, and an upper case swara letter indicates two aksharakala duration. A comma placed near a music note increases its duration by one aksharakala and a

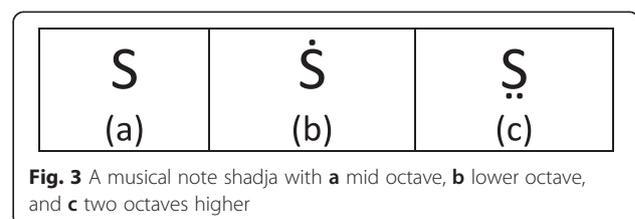


Fig. 3 A musical note shadja with **a** mid octave, **b** lower octave, and **c** two octaves higher

semicolon by two. Similarly, a single horizontal line over the swara reduces the swara duration to its half and double over or under line reduces it to its quarter.

The duration of a rest note is indicated using the necessary number of semicolon or comma symbols placed inside simple parenthesis, e.g., (;).

3.4.4 Note variety

In Carnatic music, each music note (swara) can represent more than one pitch value, usually two, according to the raga followed by the composition. Because of this characteristic, they are called “note varieties” as each swara note provides many colors to choose from.

Generally, there are no special signs or symbols to represent the variety of the note. This information is implicitly associated with the raga of the song. However, some subscripted numerals with Swara symbols are rarely used to denote sharp and flat varieties of the notes which comprise the twelve “swarasthanas” [18] [19]. For example, r_1 denotes the musical note “Sudhhari” or “Komalari” and r_2 denotes “Chatusrutiri” or “Tivrari” [18] [19].

3.4.5 Additional symbols

There are a few special symbols used in the notation scheme for denoting some musical features like articulation, ornaments etc. All of them are notated using symbols attached with the swara letter. This includes symbols for an ascending or descending glide, foreign note, stressed note, repeat symbols, and gamaka mark. A complete list of such symbols and the symbol used is given in Table 2.

Rarely, some notes which are not part of the raga specification are used, and such notes are represented by an asterisk mark over the swara symbol. The repeat symbol, usually found at the end of an avarata (measure) denotes that the portion of music should be repeated. A stressed note, denoted by letter “w” over the swara, is similar to staccato in functionality, as used in western music. The gamaka mark, represented by a tilde symbol over the swara symbol symbolizes ornamentation which is of utmost importance to Indian music. A music phrase, unlike in western music, represents a set of musical notes which has to be sung together in one breath duration

Table 2 List of additional symbols used and their notation

| Name of symbol | Symbol |
|----------------------------|---------|
| Ascending/descending glide | / and \ |
| Foreign note | * |
| Repeat avarata | (r) |
| Stressed note | w |
| Gamaka | ~ |
| Musical phrase | -- |

and is symbolized by hyphens at the start and end of the phrase.

3.5 Notation arrangement

The music notes with adjoint symbols are written on a straight line similar to tonic solfa notation in western music. The music notes are then grouped according to the rhythm structure (tala) of the composition, which is similar to the grouping of notes with time signatures in western music notation. Here, we explain the grouping mechanism in comparison with grouping in western notation for easy understanding of readers.

In western notation, notes are grouped according to the indicated time measure to form equi-measured bars as demonstrated in Fig. 4. In Carnatic music, the grouping or structuring of music notes is done according to the rhythm pattern (tala). The tala (rhythm) specification consists of a set of basic elements (angas), each with a specific duration. The basic rhythm pattern repeats over the entire composition. The music notes are grouped in such a way that the total duration of the music notes is equal to the corresponding anga duration as illustrated in Fig. 4. This shows that Indian music follows variable measured bars as opposed to equi-measured bars in western music. Once the music notes are grouped for one cycle of the rhythm pattern, it is called “avarata.” Thus, the grouping of music notes is done until the end of the composition. This is illustrated in Fig. 5.

4 Related works

Music representation systems encompass musical information in any of the three levels: sound, music notation, or data for analysis [20]. Music notations are generally an encoding of abstract representations of music. They contain instructions for performance and representation of sound. Several structured representation like the hierarchical music structures representation [8], “Music Structures” [21], “TTrees” [22], hierarchical representation of scores [23], musical events [20], musical tones [23], abstract datatype representation in [24], generative approaches like in grammars [24], Petri Nets [25], Markov Models [26], and object-oriented approaches like SmOke [27], Aspect Music [20], and graph-based approaches [28], for representing musical data.

Music representation systems can be classified as audio signal representations, resulting from the recording of sound sources or from direct electronic synthesis, and symbolic representations which represent discrete musical events such as notes, rhythm etc. [28]. *The proposed system is a symbolic representation, and is content-aware and can relate musical events to formalized concepts of Carnatic music theory.*

The musical representation systems can also be classified according to the encoding system (format) used for

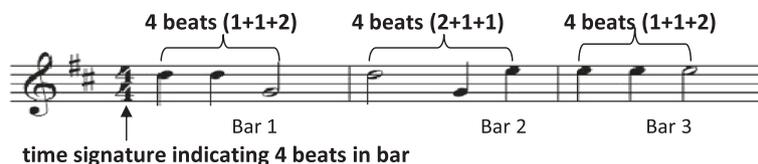


Fig. 4 Example of grouping of notes with time in western music notation

storing the information. Thus, it can be classified into binary, ASCII-based, XML-based, and proprietary formats. The popular binary formats are MP3, WAV etc. The ASCII formats like Cadenza [5], DARMS [29], Guido [4], ENP [30], LilyPond [11], and Humdrum [3] encode musical score information using ASCII-based text. They can be further classified as record-based, command-based, codes, and LISP-based. The XML formats like MIDI XML [31], MusicXML [32], MEI [13] etc. provide hierarchical XML representation of musical information. Also, many popular score-writing programs like Rhapsody and Sibelius use proprietary formats. *We propose to use Unicode standards to encode the Carnatic music notation called “Sargam” (or modern notation), and it can be considered as the first Unicode-based music representation system.*

Even though most of these music representation systems were evolved around western music tradition/notation, there were a few attempts to extend its applicability to other regional music traditions like Korean [24], Greek [33], ***Bhat* [34] etc. *Unlike these extensions of western music, the proposed system is a unique approach to representation of South Indian Carnatic music based on Indian music theory.*

5 The iSargam language

As opposed to many ASCII-based representations like DARMS [15], Guido [4] etc., iSargam is formed as a Unicode-based music notation representation language. That means we use various Unicode symbols to represent musical entities in Carnatic music.

In this section, we describe the iSargam representation system by explaining its approach, encoding logic, and algorithm.

5.1 Terminologies

Before describing the encoding logic, we would like to illustrate a few concepts which we developed as part of the encoding logic, viz, singleton/grouped entity, and music constituent.

5.1.1 Singleton/grouped entity

The musical symbols used in Sargam notation are classified as singleton or grouped entity according to whether they have meaning or sense in single form or they make sense only when they combine with another musical entity. For example, anumandra, the octave specification symbol makes sense only when it is joined with a musical note (Svara). Singleton musical entities are always found independent in the notation and have semantics of their own. An example of singleton entities are tala marking symbols, anga/avarta mark etc. This classification among music symbols is required due to difference in encoding single and group entities, where group entity symbols can be encoded together only and not individually.

5.1.2 Music constituent

Here, we attempt to define the term “music constituent” which is a concept developed as part of the encoding logic. A music constituent is the most basic unit of music. Here, it consists of a pitch symbol (swara), a sthayi (octave), and duration information. It may be noted that a rest note does not have pitch and octave but has duration. Thus, the basic constituent elements are swara and rest. So, the general syntax of a pitched music constituent can be defined as

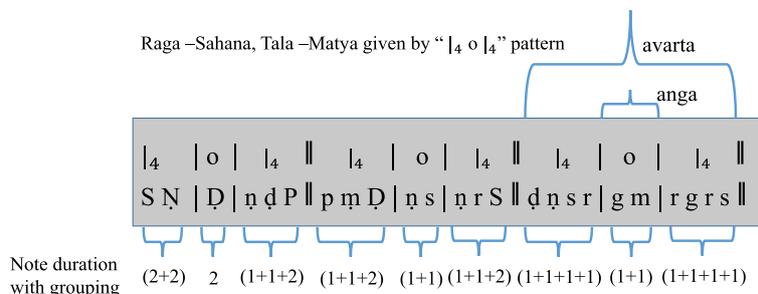


Fig. 5 Arrangement of music notes in Sargam notation

$$\langle \text{swara-syllable} \rangle [\langle \text{octave} \rangle][\langle \text{duration} \rangle] \quad (1)$$

or in the case of an unpitched (rest) note defined or expressed as

$$([\langle \text{duration-symbol-comma} \rangle | \langle \text{duration-symbol-semicolon} \rangle] *) \quad (2)$$

It may be noted that the swara syllable alone is a complete musical constituent since it already contains octave and duration information.

The basic element can be grouped according to some rhythmic pattern or it can be further augmented with additional symbols or other music notes, forming various grouped entities. In our approach, the former is called rhythmic group and the latter is called notational grouping. This latter is again classified into intra notational and inter notational group entities. Inter notational groupings are always associated with music constituents.

Intra notational grouping occurs when the music constituent is further augmented by adding parameters which apply in a single note level. This is denoted by adding extra signs or symbols to the base syllable. The musical entities in this category are stressed note symbol (w), gamaka mark (~), foreign note symbol (*), violin marks, upward stroke of the bow (v), and downward stroke of the bow (^). In case of inter notational grouping, multiple musical notes are grouped together, mostly to give a musical expression such as a musical phrase and ascending or descending glides.

5.2 Encoding logic

Having defined the basic terminologies, now we attempt to present our encoding logic. Initially, our system maps every Sargam notation symbol to a Unicode symbol. The chosen Unicode character resembles the Sargam notation symbol used. Each symbol is also assigned a priority number.

The encoding logic depends on whether the music symbol is a singleton or grouped entity. So, for encoding of a notated composition, we take every symbol and

check if it can be further split into different characters as illustrated in Fig. 6. This is done by checking the baseline and upperline of the character. An atomic symbol is a singleton entity and it is directly mapped to its representation Unicode symbol.

If the character is a grouped entity, then it might be a Swara symbol indicating a pitched note or it is a rest note, where both are music constituents. We process the constituent symbols together with a priority queue [35]. The priority queue orders them according to the pre-assigned symbol priority and thus produces unambiguous encoding. A pseudo code for the proposed encoding scheme is given in Table 3.

The iSargam system chooses the unique numbers carefully so as to make sure that the corresponding Unicode character almost fully resembles the actual music notation in appearance, even in the case of the grouping or joining of music notation symbols. The advantage here is that Unicode symbols appear discrete in encoding, which favors easy identification of music entities for music processing, but in appearance, it appears joined, resembling the original notation. Also, it may be noted that in such a representation, a combined notation can be easily split to its constituent basic music entities.

We use Unicode full width forms for standalone music elements like swara syllable or duration, and Unicode combining diacritical marks for adjunct symbols like octave, stress, foreign note indication, duration symbols which symbolize duration less than one unit, etc. More specifically, all the intra notational symbols and violin marks are represented by combining diacritical marks. Additionally, we use Unicode full width symbols for representing symbols in the rhythmic group like anga, avarta symbols, which are analogous to measure and bar markings for western music, and other rhythm-specific elements like laghu, plutum etc., which mark the number of beats within a measure.

Having said encoding logic, we now attempt to illustrate our encoding approach for each music entity, illustrated with an example, as given in Table 4.

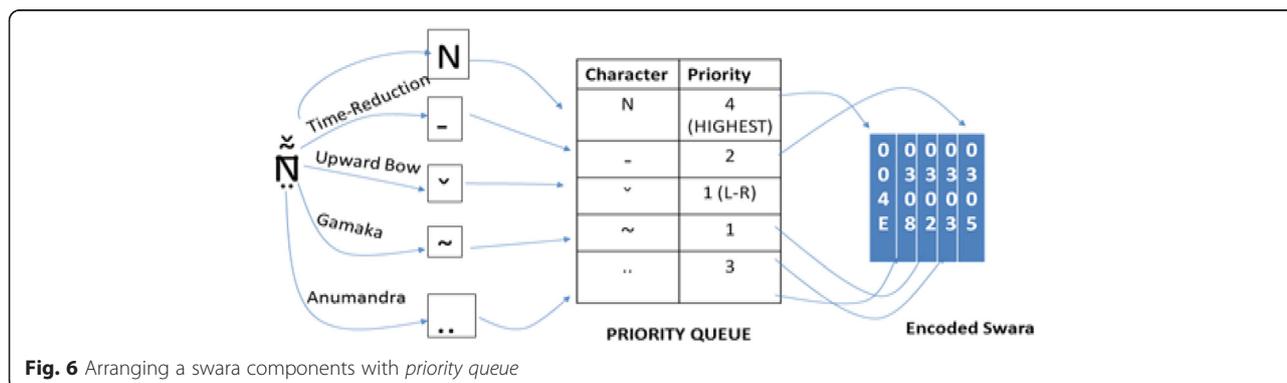


Fig. 6 Arranging a swara components with priority queue

Table 3 Abstract encoding logic

```

Function iSargamEncoding(Document doc)
For i=1 to doc.length
  X=readInputChar(i)
  If stand-alone- item(X)
    write(Unicode(X))
  Else If grouped-item(X)
    If MusicConstituent(X)
      If pitchedSwara(X)
        Components= split(symbol)
        For j=0 to length(Components)
          pushToPriorityQueue(Components[j])
        next j
        For j=0 to length(Components)
          y=popFromPriorityQueue(Components[j])
          write(Unicode(y));
        next j
      else If unpitchedNote(X)
        rest=readFullRestSymbol(i);
        i=i+length(rest);
        For j=0 to length(rest)
          write(Unicode(rest[j]))
        next j
      End If
    Else
      write(Unicode(X));
    End If
  End If
Next i
End Function

```

The encoded file consists of various sections, viz, the header, rhythm markup, and actual composition, explained as follows.

5.2.1 Header section

The header section accompanies every music notation. It mainly consists of two sections, viz, a compulsory part and an optional part. The compulsory part is known as the music description part, and it specifies the most important elements for interpreting the notation. These most important fields are raga name and tala name. The optional part consists of the fields composition title, composer, arohana/avarohana, and mela. The header elements are considered as keywords which are case-insensitive and are separated by a colon character. These values are case-sensitive.

5.2.2 Tala (rhythm) markup section

The tala section marks the rhythm pattern of the composition. Usually, a rhythm pattern is defined as a combination of its basic elements called angas, as described in the previous section. For encoding, each anga symbol is assigned a Unicode-based identifier and is separated from the others by using the vertical line symbol (U+007C). The avarta is marked by a double pipeline

symbol (U+01C1) at the beginning and the end as shown in Table 4.

5.2.3 Music notation section

This section contains the notation of the actual composition. It consists of music constituents along with required signs and symbols with notational and rhythmic grouping. The encoding of the actual music notation is illustrated in the following subsections.

5.2.3.1 Encoding of rhythm grouping The music notes are grouped according to the tala specification, splitting them into many anga and avarta as described in the previous section. The angas are separated by vertical line symbol (U+007C), and avarta is marked by double pipeline symbol (U+01C1) at the beginning and the end.

The general form of avarta can be given by

$$\| \langle \text{music-note} \rangle \dots | \langle \text{music-note} \rangle \dots | \dots (\text{r}) \| \quad (3)$$

Sometimes, a repeat symbol is inserted in front of the avarta end to denote repetition of an avarta. The symbol used is “(r)” and the encoding is encoded by the Unicode symbol parenthesized Latin small letter R (U+24AD).

Table 4 iSargam encoded file

```

<!--Header--!>

Raga:sankarabharana

Tala:Matya

Mela:28

Title:JANYA RAGA LAKSHANA GITA

Composer:PAIDALA GURUMURTI SASTRI

Araohana:s,r,g,m,p,m,d,n,s

Avarohana:s,n,d,p,m,G,m,R,g,r,s

<!--Tala specification--!>

||4|°| 14|

<!--Composition--!>

||S N. |D.|n.d.P.| p.m.D.|n.s |n.rS|

||d.n.sr|gm|rgrs || n.sgr|n.s |D.d.n|

||d.p.m.d.|D.|n.sr||Mmd|D|Nns.||d.n.sr| gm | rgrs||n.sgr| n.s |D.d.n||

||d.p.m.d.|D.|n.sr||Mmd|D|Nns.||
    
```

5.2.3.2 Encoding of music note The encoding strategy followed for a pitched music note and an unpitched note is different. The encoding logic for an unpitched note is straightforward like a singleton entity. But a pitched note is regarded as a grouped entity and we use priority-based encoding for this set as illustrated in Fig. 6.

Unlike the encoding strategy used for singleton entities, encoding for grouped entities is done together and not as individual elements. It can be observed that any pitched note is an extension of a pitched music constituent. In case of a pitched music constituent, the swara symbol is followed by octave and duration symbols, as mentioned in the previous section. Additionally, musical notes can contain additional characters which augment the basic music note like stressed note symbol, foreign note symbol etc. The musical note may then be part of another group in case of occurrence of musical phrase or glide expressions. It might look straightforward to assign symbol priorities in the same order. But this does not work due to a difference in the type of notation symbols used. So, iSargam develops a new encoding logic for notating a music note, which is explained here. In this context, we

would like to redefine the general syntax of music notes given in the previous section to enable easy encoding.

A musical note can be defined as a music constituent optionally followed by other additional qualifiers which may denote some kind of ornamentation.

$$\begin{aligned}
 < \text{music-note} > = < \text{music-constituent} > \\
 & \quad [\text{additional-symbols}][\text{glide} < \text{musical-note} >]
 \end{aligned}
 \tag{4}$$

$$\begin{aligned}
 < \text{music-constituent} > = < \text{swara-syllable} > \\
 & \quad [< \text{octave} >][< \text{duration-1} >]
 \end{aligned}
 \tag{5}$$

The iSargam system modifies this general syntax to construct the syntax for our encoding, to take advantage of certain properties of Unicode combining diacritics, as given below. Also, we assign priorities for these symbols to ensure proper ordering in the encoded script, which avoids ambiguous representations.

$$\langle \text{swara-syllable} \rangle [\langle \text{octave} \rangle][\langle \text{duration-1} \rangle] [\text{additional-symbols}] \tag{6}$$

or

$$\langle \text{swara-syllable} \rangle [\langle \text{octave} \rangle][\text{additional-symbols}] [\langle \text{duration-2} \rangle] \tag{7}$$

The <duration-1> symbol consists of duration symbols which have duration of less than one aksharakala unit, and we use Unicode diacritics symbols to represent them. <additional-symbols> are also represented by Unicode diacritics symbols. The <duration-2> consists of duration which have duration greater than one aksharakala unit, and we use basic Latin Unicode symbols comma and semicolon to represent them.

Now, we introduce symbol priority which defines the encoding order within music note symbols as illustrated in Fig. 6. The swara syllable letters are from the English alphabet and are assigned with the maximum priority. We use Unicode Latin letters to represent them. All the octave symbols are adjoint symbols with the swara letter, and they form the next priority level. They are represented by Unicode diacritic symbols. Next, we deal with the duration elements by splitting it into two groups, viz, duration-1 and duration-2. The first set consists of duration symbols which are adjoint symbols with the swara character. So, their position is placed close to the swara letter and is assigned a priority value of three. The second set of duration symbols consists of standalone symbols which are encoded with Unicode full width forms. The symbols used are comma and semicolon, and they are assigned only the next priority after the additional marks, which is a priority value of five. The term <additional-symbols> represent the set of musical entities which augment the basic music note, which are, a foreign note marked by an asterisk, a stressed note marked by wavy line, a gamaka mark symbolized by a tiled symbol, and violin marks upward and downward circumflex accent. All these additional symbols are to be placed above the swara syllable and so we assign them a priority value of four and we use Unicodes of combining diacritical marks to encode them. The full encoding algorithm is given in Table 5. Here, we use Unicode character “combining x above” for foreign note, “combining inverted double arch” symbol for stressed note, “combining tilde” symbol for indicating gamaka, and combining carons and combining circumflex accent for violin marks.

5.2.3.3 Inter notational grouping The inter notation grouping occurs within an avarta. For providing certain musical effects, more than one music note is grouped

Table 5 iSargam encoding algorithm

| Algorithm | Convert-MUSIC-NOTATION-TO-REPRESENTATION. Given a vector V containing N elements, this algorithm converts music notation to its representation |
|-----------|--|
| 1. | [Initialize variables and vector] X ← "" OUTPUT ← "" |
| 2. | [CHECK FOR START OF NOTATION] If V[i] ≠ " " then OUTPUT ← Unicode(V[i]) i ← i + 1 else report ERROR. End If |
| 3. | Repeat thru Step 4-5 for i=0 to LENGTH(vector) |
| 4. | If isSwarat(V[i]) then i ← i + 1 Q ← SplitToChars(V[i]) for j=0 to LENGTH(Q) do Select case (Q[j]) Case ['S' or 'R' or 'G' or 'P' or 'D' or 'N' or 's' or 'r' or 'g' or 'p' or 'd' or 'n']: PRIORITY_VALUE ← 1 ENQUEUE(PRIORITY_QUEUE,Unicode(Q[j]), PRIORITY_VALUE) Case ['*' or '^' or '~' or 'ˆ' or '˜' or '˘' or '˙']: PRIORITY_VALUE ← 4 ENQUEUE(PRIORITY_QUEUE,Unicode(Q[j]), PRIORITY_VALUE) Case [',' or ';' or 'ˆ' or '˘']: PRIORITY_VALUE ← 2 ENQUEUE(PRIORITY_QUEUE,Unicode(Q[j]), PRIORITY_VALUE) OUTPUT ← Unicode(Q[j]) Case [',', or ';']: PRIORITY ← 5 ENQUEUE(PRIORITY_QUEUE,Unicode(Q[j]), PRIORITY_VALUE) Case [ˆ or ˘ or ˙ or ˚]: PRIORITY ← 3 Return End Select Case OUTPUT=OUTPUT+DEQUEUE_IN_PRIORITY_ORDER(PRIORITY_QUEUE) OUTPUT=OUTPUT+Unicode(V[i]) Goto Step 4. For k=0 to PRIORITY_QUEUE.LENGTH OUTPUT=OUTPUT+DEQUEUE_IN_PRIORITY_ORDER(PRIORITY_QUEUE) End If |
| 5. | Case ':': OUTPUT=OUTPUT+Unicode('ˆ') i ← i + 1 Goto Step 4. |
| 6. | Case 'ˆ': temp ← Unicode(V[i]) i ← i + 1 While V[i] ≠ ' ' do If V[i] = 'ˆ' or V[i] = '˘' or V[i] = '˙' then temp ← temp + Unicode(V[i]) i ← i + 1 End If Next |
| 7. | Case 'ˆ' or '˘': OUTPUT=OUTPUT+Unicode(V[i]) i ← i + 1 If isSwarat(V[i]) then Goto Step 4 Else Error End If |
| 8. | Case '˙': OUTPUT=OUTPUT+Unicode('˙') i ← i + 1 If isSwarat(V[i]) then Goto Step 4 Else Error End If |
| 9. | Case '˚': OUTPUT=OUTPUT+Unicode('˚') Return |

Table 6 Actual display in Unicode-supported text editor

```

<!--Header-->

Raga:sankarabharana

Tala:Matya

Mela:28

Title:JANYA RAGA LAKSHANA GITA

Composer:PAIDALA GURUMURTI SASTRI

Araohana:s,r,g,m,p,m,d,n,s

Avarohana:s,n,d,p,m,G,m,R,g,r,s

<!--Tala specification--!>

||4|°| |4|

<!--Composition--!>

|S N|Ḍ|ñ.ḍP̣|p̣ṛñḌ|ñs |ñrS|

|ḍñsr|gm|rgrs | ñsgr|ñs |Ḍñl|

|ḍp̣ṛṃḍ|Ḍ|ñsrslMmd|Ḍ|Nns|
    
```

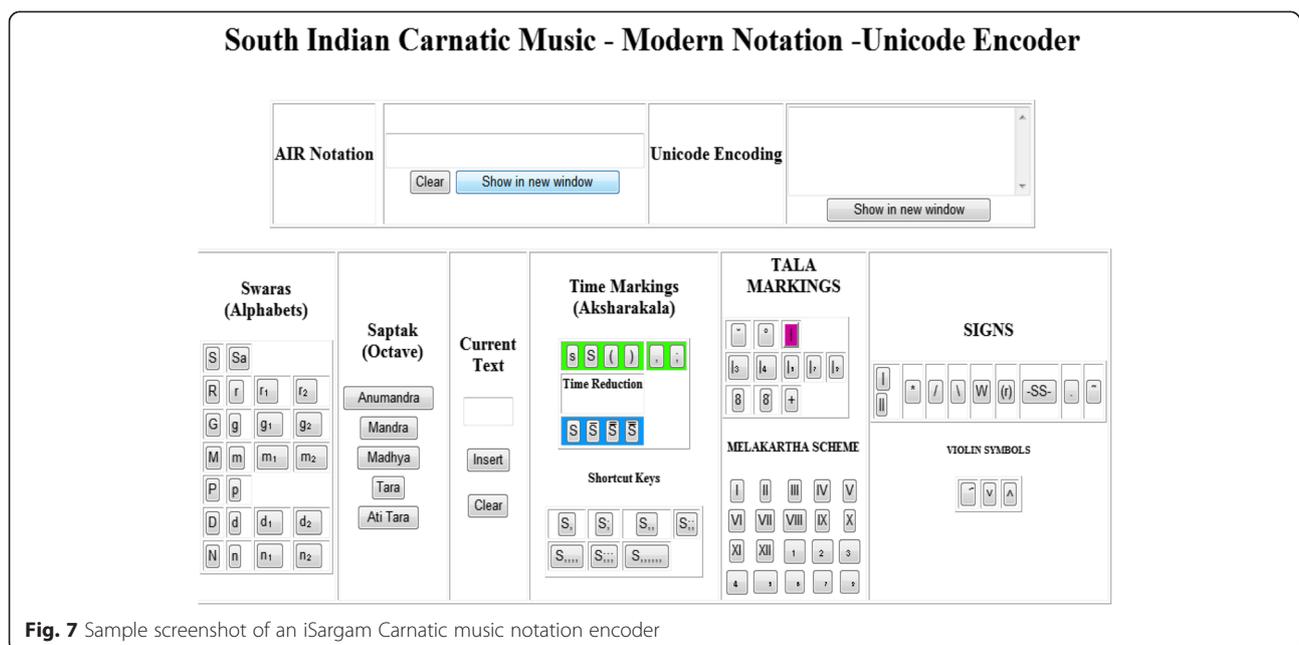


Fig. 7 Sample screenshot of an iSargam Carnatic music notation encoder

together. In the case of Carnatic music, there are two occurrences of such inter notation grouping, viz, the musical phrase and the ascending/descending glides. In Carnatic music, the meaning of music phrase takes on a different meaning than the one used in western music. In Carnatic music, the intended meaning of music phrase is that the given set of notes has to be sung or played within one breath time in the case of vocal or in a single bow in the case of violin. In Carnatic notation scheme, the musical notes belonging to a phrase expression are placed within two hyphens. The general syntax is as given below, and the encoding places Unicode of hyphen-minus (U+002D) symbol before and after the first and last music note belonging to the expression.

$$- < \text{music-note} > < \text{music-note} > [< \text{music-note} > *] - \quad (8)$$

In the case of the glide marks, it can be treated as an operation between two musical notes. The ascending glide places a glyph character between the music notes, while the descending glide places forward slash character between the notes.

$$< \text{music-note} > [/ \text{or}] < \text{music-note} > \quad (9)$$

So, the encoding also follows a simple strategy of placing the Unicode of the backward or forward slash between the two musical note elements. Even though they are grouped symbols, the encoding strategy used encodes them in a straightforward manner by just inserting the Unicode symbol solidus (U+002F) for ascending glide and reverse solidus (U+005C) for descending glide.

5.3 Encoding algorithm

Now, according to our prioritized encoding algorithm, we process note by note, encode them by splitting each musical note into basic music elements, represent them with an assigned unique id taken from the Unicode set, and arrange them in priority order. This procedure is illustrated in Table 5. A sample view of output viewed with a Unicode-based text editor is given in Table 6.

6 Implementation

So far, implementation covers the development of an editor for Sargam notation. Similar to the virtual keypad present in a calculator application program, we have designed an internal keypad which displays a visual keypad on the computer screen with buttons corresponding to the symbols used in the Sargam notation. By pressing the appropriate buttons, a user can enter the desired music notations. On hitting each button, the corresponding entry can be viewed in the editor. The editor uses the iSargam encoding system to store the musical

information entered by the user. A screenshot of the encoder application is given in Fig. 7.

7 Conclusions

The proposed work is an encoding system for Carnatic music notation. The limitation of this work is that it only stores musical information in a retrievable form. A possible extension of this work is to integrate search and retrieval mechanisms upon the encoded form.

Most of the popular music retrieval systems employ search mechanisms with note patterns fed into the system as a set of note or swara symbols using the basic ASCII character set. Similarly, most of the query-by-humming-based music information retrieval systems internally convert the user-hummed query into music notations before the actual comparison process. Since iSargam is a Unicode-based encoding system, an extension of the existing MIR systems to support iSargam files or an application of the MIR approaches to the iSargam musical database will only require an integration of a simple ASCII to Unicode conversion module. The same approach can be used to apply the musical analysis mechanisms to the iSargam files or database.

Another suggested improvement is to integrate the work with a knowledgebase developed for Indian music and to incorporate a playback facility with the editor.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work has been carried out as part of a sponsored project entitled "Representation, Retrieval and Analysis Mechanisms for South Indian Carnatic Music using Computers" with the sponsorship of Ministry of Culture, Government of India, for a period of 2 years. The authors acknowledge the financial support made the agency to carry out the proposed work.

Author details

¹Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India. ²School of Drama and Fine Arts, Dr. John Matthai Centre, University of Calicut, Thrissur, Kerala, India. ³Department of Music, Government Women's college, Trivandrum, Kerala, India.

Received: 15 May 2015 Accepted: 5 February 2016

Published online: 16 February 2016

References

1. M Balaban, *Musical structures: interleaving the temporal and hierarchical aspects to music*. (In Understanding Music with AI: Perspectives in Music Cognition, MIT Press, Cambridge, 1992), pp. 110–138
2. N Han-Wen, N Jan, *LILYPOND, A System for Automated Music Engraving* (Paper presented at the XIV Colloquium on Musical Informatics, Firenze, 2003)
3. D Huron, Music information processing using the Humdrum Toolkit: concepts, examples, and lessons. *Comp. Music J.* MIT Press **26**(2), 11–26 (2002)
4. HH Hoos, KA Hamel, K Renz, J Kilian, *Representing score-level music using the GUIDO music notation format*. (Computing in Musicology, MIT Press, Cambridge, 2001), p. 12
5. HS Field-Richards, Cadenza: a music description language. *Comput. Music. J.* **17**(4), 60–72 (1993)
6. Solesmes, *Antiphonale Monasticum*, vol 1, (Paraclete Press, 2005), pp. 542–543
7. G Bays, *ScoreSVG: a new software framework for capturing the semantic meaning and graphical representation of musical Scores Using Java2D, XML,*

- and SVG (Master's Thesis, College of Arts and Sciences, Georgia State University, 2005)
8. MM Erin, R Jenny, Saffran, music and language: a developmental comparison. *Music. Percept.* **21**(3), 289–311 (2004)
 9. Center for Computer Research in Music and Acoustics, Common Music Notation, (Stanford University, 2010), <https://ccrma.stanford.edu/software/cmn/cmn/cmn.html>. Accessed on 12 April 2015
 10. W Kornfeld, Machine Tongues VII: LISP. *Comput. Music. J.* **4**(22), 6–12 (1980)
 11. M Laurson, *Dissertation, Sibelius Academy*, 1996
 12. M Good, G Actor, *Using MusicXML for file interchange*, In *Proceedings Third International Conference on WEB Delivering of Music, Leeds, UK, September 15–17, 2003* (IEEE Press, Los Alamitos, 2003), p. 153
 13. P Roland, *Design Patterns in XML Music Representation* (Paper presented at the Fourth International Conference on Music Information Retrieval, University of Johns Hopkins, Baltimore, 2003)
 14. R Perry, The music encoding initiative, in *Proceedings First International Conference on Musical Application using XML*, ed. by H Goffredo, L Maurizio, 2002
 15. P Bellini, P Nesi, *WEDELMUSIC Format: An XML Music Notation Format for Emerging Applications*, *First International Conference on WEB Delivering of Music* (IEEE Computer Society, Washington, 2001), pp. 79–87
 16. PP Narayanaswami, VS Jayaraman (eds.), *Sangita Sampradaya Pradarsini English Web Edition*, 2006. <http://ibiblio.org/guruguha/ssp.htm>. Accessed on 15 June 2014
 17. P Sambamurthy, *A Practical Course in Karnatik Music*, 1st edn. (The Indian Music Publishing House, India, 1963), pp. 23–56
 18. L Issac, *Theory of Indian music*. (Shyam Printers, 1967), pp. 18–24
 19. S Bhagyalekshmy, *Ragas in Carnatic Music*, (CBH Publications, 2003)
 20. P Hill, S Holland, R Laney, An introduction to aspect oriented music representation. *Comput. Music. J.* **31**(4), 47–58 (2007)
 21. A Smail, G Wiggins, M Harris, Hierarchical music representation for composition and analysis, computers and the humanities. *Kluwer Acad. Publishers* **27**(1), 7–17 (1993)
 22. X Serra, *The Musical Communication Chain and Its Modeling*, *Mathematics and Music*, 243–255, 2002
 23. M Besson, AD Friederici, Language and music: a comparative view. *Music. Percept.* **16**(1), 1–9 (1998)
 24. JH Lee, JS Downie, A Renear, Representing Korean traditional music notation in XML, in *Proceedings of the Third International Conference on Music Information Retrieval*. (IRCAM Centre Pompidou, Paris, 2002)
 25. AD Patel, Music and the brain: three links to language. *Oxford Handbook of Music Psychology*. (2008). doi:10.1093/oxfordhb/9780199298457.013.0019
 26. F Ramus, M Nesport, J Mehler, Correlates of linguistic rhythm in the speech signal. *Cognition* **73**, 265–292 (1999)
 27. J Kippen, B Bel, Modelling music with grammars: formal language representation in the Bol Processor, *Computer Representations and Models in Music*. (Academic Press Limited, 1992), pp. 207–232
 28. F Lerdahl, R Jackendoff, *A generative theory of tonal music* (MIT Press, Cambridge, 1985), pp. 130–162
 29. R F Erickson, The DARMS project: A status report, *Computing and the Humanities*. Springer **9**(6), 291–298 (1975)
 30. M Kuuskankare, M Laurson, Expressive notation package. *Comput. Music. J.* **30**(4), 67–79 (2006)
 31. MIDI Manufactures Association. Making music with MIDI, <https://www.midi.org/>. Accessed on 25 March 2015
 32. Michael Good, MusicXML. <http://www.musicxml.com/>. Accessed on 1 April 2015
 33. D Politis, D Margounakis, S Lazaropoulos, Leontios, Papaleontiou, George Botsaris, Konstantinos Vandikas, Emulation of ancient Greek music using sound synthesis and historical notation. *Comput. Music. J.* **32**(4), 48–63 (2008)
 34. P Chordia, *A system for the analysis and representation of Bandishes and Gats using Humdrum syntax*, in *Proceedings of the 2007 Frontiers of Research in Speech and Music Conference* (Mysore, India, 2007)
 35. MA Weiss, *Data Structures and Algorithm Analysis in C++*, Pearson Education, 1993
 36. Creative Commons. <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode>. Accessed on 28 Dec 2015

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com