The Representation of Musical Information in Machine-Readable Form

The use of the computer for research in musicology is becoming increasingly more common. For many types of applications, the effectiveness of the computer as a research tool depends upon the success with which relevant musical data can be represented in a form which is intelligible to the machine.

This problem of representation, which has challenged musicologists for more than twenty-five years, has no apparent simple or single type of solution. The methods for representing musical information that have been developed over this period of time are many and varied. Several large scale efforts such as DARMS and MUSTRAN have met with reasonably good success. They have received high marks for their utility and completeness, and have gone on to produce important secondary efforts in music printing and analysis. Yet the acceptance and use of such systems is far from universal. In fact, today, more than twenty years after the beginning of the development of DARMS, there are new systems still being devised, and in terms of numbers of projects, the use of custom or special encoding systems far outweighs the use of such major standard systems. In order to gain a better understanding of why this is so, and to find out more about the activities of scholars and system designers in this area, the Center has conducted a survey on methods of music representation, the results of which are reported in this article.

To collect information on music representation, the Center included a section on music encoding in its annual questionnaire on computer applications in musicology. Our objective was to gain some idea about the variety and number of different systems currently in use, how these systems dealt with the problems of organizing and storing information, and finally, what issues needed to be resolved in order for musical information to be transferred from one system to another. Scholars involved in actual projects in musicology (end users) were presented with two simple musical examples and asked to indicate how this music would be encoded in their system. As part of a music printing update, system and software developers were sent four examples of a more complex nature and asked to provide information on musical encoding as well as a description of the internal representation used for storing data and for preparing data for printed output.

The response to this part of the end-user questionnaire was reasonably complete. Of a total of 95 responses describing projects involving the encoding of musical information, 22 indicated the use of well established coding systems such as DARMS, MUSTRAN or Plaine and Easie; 48 indicated the use of more recent encoding systems or of custom designed systems; six indicated the use of commercial software and 19 gave no indication of the kind of system being used. Of the 48 respondents using more recent or custom design systems, 15 sent in descriptions of their systems and another six indicated the use of systems for which the Center has descriptions from other sources.

The response from system and software developers was more variable. While commercial system and software developers were generally eager to demonstrate the music printing capabilities of their systems, they seemed less interested in discussing how they represented their musical data internally. Nevertheless, the Center did receive good responses from some important members of this group. System and software developers working in an academic setting were generally more forthright in describing their input and internal representation systems.

Before proceeding with a description and analysis of the various encoding systems received by the Center, it is worth reviewing some of the factors of system design which influence the form that encoding systems take, their scope and versatility, their speed and ease of use as an input tool, and their suitability for various kinds of research and analysis.

1. **Level of completeness.** Some of the first people to work on the subject of music representation recognized the importance of distinguishing between a *code* and a *language*. In representing musical information there are many conceivable levels of completeness. Scholars working on a selected repertory for the purpose of answering defined questions may only need to consider information about pitch alone, about rhythm alone, or about pitch and rhythm without reference to dynamics or articulation. A simple code can be entirely adequate for a dedicated self-designed system. Encoding languages such as DARMS and MUSTRAN, on the other hand, seek to be able to encode any element of musical information from any repertory that is performed from common musical notation (CMN).

2. **Intended uses of the data.** Encoded musical information can in principle support many applications. It can be used to create sound, it can be used to create a score and/or parts for performance, and it can be used internally as a basis for analysis of perceived sound or of a notated composition. Schemes for representing music which are devised for specified purposes may contain elements of information that are irrelevant to other purposes. Beams offer a good example. A program for printing music must be able to deduce when a beam is necessary and determine from the internal representation the points where that beam starts and ends. Such information would be irrelevant to audio output. The MIDI representation offers another example of information that is adequately precise for one use but is too vague for another. The sounding pitch of the black note between F and G is represented in MIDI by a single integer derived from the chromatic scale. This representation is adequate for the purpose of audio output but does not carry sufficient information for printing or analysis, where discrimination between F# and Gb is a requirement.

3. **Provision for expanded use.** A particular system of encoding may be adequate for a dedicated task, but after a musical repertory is encoded, new uses for it may become apparent. Deciding which data attributes should be included to support unforeseen uses is a perennial responsibility of text-based projects. The incentive to avoid unnecessary encoding is based on considerations of storage space, processing time and ease of representation. The balance between these needs is sometimes difficult to strike in applications involving musical data.

4. **Inclusion of interpretative data.** In theory, scholars would normally prefer data that is "objective" or "unedited," but in practice many scholars enter into the computer data that is descriptive or already partly interpreted. Within the context of specific tasks, this can save encoding time and simplify analysis. Many encoding systems specify global accidentals in terms of a key or a key signature. Some systems also specify a "tonic" reference pitch. This information, which is sometimes difficult to deduce by computer, can be very helpful for certain types of analysis. In other cases, inclusion of such information is not helpful. For modal repertories in which a key signature is not related to a "key" in the

tonal sense and for some twentieth century repertories in which sharps and flats may be found in the same "signature," the simple declarations of "key" or "key signature" cannot be used without obscuring attributes that may be vital to analytical issues.

5. **Provision for shorthand representation.** A good deal of encoding time and storage space can be saved by allowing certain musical attributes to be defined relative to previous information. The degree to which musical attributes can be relatively defined is a property of system design. For example, where pitch notation is represented by the letters A through G and a pitch register, the register needs to be specified only in those places where it actually changes. If a key signature is declared at the start of a piece, pitches may be identified by letter name only, with affected letters assumed to be altered in accordance with the key signature unless accidentals are notated. These examples of shorthand notation can actually improve the readability of the code. Information on note duration can also be condensed by indicating duration only in cases where the rhythmic values change. A succession of eighth notes, for example, may be indicated by a single code element. Such methods of shorthand have many obvious advantages but also need to be employed with caution, since errors in initial declarations can cause havoc in retrieving and interpreting the data, thus limiting the value of the results.

6. **Methods of data organization.** In representing music in the computer, it is generally necessary to deal with musical attributes independently. There is no natural order among these attributes. Some systems give pitch information followed by durational information on a note-for-note basis. Other systems give duration first, followed by pitch. Some systems encode pitch information for an entire sequence of notes, and then encode durational information for the same sequence. This approach is often referred to as a "two-pass" system. When entering information from several musical parts which sound simultaneously, data may be encoded either vertically as a series of simultaneous events, or horizontally as a series of individual parts of specified length. In general, flexibility in the methods of encoding and representing data will contribute to the versatility of a representation system but will also add to its complexity.

Examples of Musical Encoding Systems

The descriptions of the user-designed encoding systems received by the Center are interesting not only for the information they provide on current methods for representing musical data in machine readable form but also for the detail they provide on the kinds of data scholars are finding useful to put into the computer. For this reason, we have chosen to display all of the encodings we received for the two musical examples in the questionnaire.

In taking this approach, we have not attempted to provide explanations on how the various encoding systems function, beyond the comments provided by the respondents themselves. From our own study, it appears that the methods used for each encoding system are self-evident and that part of the value to be gained from studying these systems is acquired in the process of trying to figure them out. In most cases, we have not listed the names of the designers/encoders, since the emphasis here is not on individual systems but on features of commonality and difference. The listing below pairs the various encoding systems with the letters of the alphabet for purposes of identification.

Example 1: 

Example 2: 

A: Example 1 not encoded

Example 2

0 1 2 3 1 2 0 4 3 2 1 1 0

* pitches represented in mode 1

B: **Example 1**

%KS 1- %TS 12:8 *=BAR 1=* =1= (F5 G5 F5) 72 C6 48 (A5 F5 E5 F5) 96 R 48 F5 24 / [other voices in Bar 1 would follow; then --] *=Bar 2=* =1= (G5 F5 F5) 72 C5 48 (G5 A5 G5 F5) 96 - - -

* Example is assumed to be Bar 1, and "voice" number 1 of a multi-part texture. Spaces are not significant.

Example 2

%KS 0 %TS 4:2 C3 192 (D3 E3 F3 D3) 192 / E3 48 C3 48 G3 192 F3 96 / J F3 96 E3 96 D3 144 D3 48 / C3 192

* Example is encoded as an isolated part. Bar numbers and voice numbers are not required.

C: **Example 1**

F 2 1 5 3 1 -2 1 2 1 2 -4 2 3 2 1

Example 2

C 2 3 4 2 3 1 5 4 3 2 1

* System is concerned only with pitch class incipits. Rhythm, duration, and octave are not included.

D: **Example 1**

[1%; 12/8; 3]: 3(f "q"f "), 2A(C"'a"), 3(f "e"f "), ?2A(f ") | 3(q"f "q"), 2A(c"q"), 3(a"g"f ")

Example 2

[; 4/2; 1]: 1(c), 4(defd) | 3B(ecg) - 2(gf) - | - 2(fe), 2A(dd) | 1(c)

E: **Example 1**

1 2 1 5 3 1 D7 U1 1 2 1 2 D5 U2 3 2 1

Example 2

1 2 3 4 2 3 1 5 4 3 2 1

F: **Example 1**

ff8 gg ff ccc4 aa8 ff ee ff r4 ff8 // gg ff gg cc4 gg8 aa gg ff

Example 2

c1 D4 E F D // E C G1 F2+ // F E D2. D4 // C1

* This code was worked out for short tunes of limited range, with ease of input for ordinary music readers a high priority.

G: **Example 1**

1 2 1 5 _ 3 1 - 7 1 0 ~ ~ 2 1 2 - 5 _ 2 3 2 1 ~ ~

* ~ = space. There is a headline for each song where the key-note (f) and the smallest duration of a tone (♪= 08) are coded.

Example 2

1 _ _ 2 3 4 2 ~ ~ 3 1 5 _ _ 4 _ ~ ~ ^ _ 3 _ 2 _ . 2 ~ ~ 1 _ _

* ~ = space. Key-note = c, smallest duration (♩ = 04).

H: **Example 1**

event(F,5,0,1). event(G,5,1,1). event(F,5,2,1). event(C,6,3,2). etc.

* Implied format: event(pitch class, register, attack point, duration). No clefs, key signature, time signature. Coding is very minimal; intended for specific kinds of analysis, not score representation.

Example 2

event(C,3,0,4). event(D,3,4,1). event(E,3,5,1) . . . event(G,3,10,4). event(F,3,14,4). event(E,3,18,2) . . .

I: **Example 1**

1,0,.5, 3,0,.5, 1,0,.5, 2,0,1, 5,0,.5, 1,0,.5, 6,0,.5, 1,0,.5, 0,1, 1,0,.5, 3,0,.5, 1,0,.5, 3,0,.5, 2,0,1, 3,0,.5, 5,0,.5, 3,0,.5, 1,0,.5,

Example 2

2,0,4, 4,0,1, 6,0,1, 1,0,1, 4,0,1, 6,0,1, 2,0,1, 3,0,4, 1,0,2, 1,0,2, 6,0,2, 4,0,3, 4,0,1, 2,0,4,

* where F = 1, C = 2, G = 3, . . . ; 0 indicates the following number is a duration value (measured in quarter notes); and the comma (,) denotes the use of data entry or "continue" key. [spaces added for clarity of presentation]

J: **Example 1**

/1/ =12:8= F48 G4 F4 C54 A48 F4 E4 F4 R4 F48 /2/ G4 F4 G4 C44 G48 A4 G4 F4

Example 2 not encoded

K: **Example 1**

[^ F/G/F/ C^ A/F/E/F/ R F/ | G/F/G/ C G/A/G/F/]

* No key signature or time signature. ^ means up octave. [^ . . .] means all notes up octave. / = half duration. Quarter note is default.

Example 2

[v C00 D E F D | E C G00 (F0 F0) E0 D0. D | C00]

* [v . . .] means all notes down octave. 0 = double duration. (. . .) = tie.

Example 1: 

Example 2: 

L: **Example 1**

key = f1; timesig = 12 8. f g f c> a f ev f _> f |

* assuming automatic beaming

Example 2

--- clef = b a>> "Glo" b "-" c d "-" b |.....

M: **Example 1**

$ G&B 12:4$ 8"FGF 4'"C8"A FEF 4-8F / GFG 4C8G AGF

Example 2

$ F4 2:2$ 1,C (Glo) 4DEFD/FC 1G 2F+/FE 2.D 4D (ri)/ 1C (a)

N: **Example 1**

Key F

P: F4 G F C5 A F E F R (F G)2 C G A G F

R: (8)3 (4 (8)4)3

Example 2

P: C2 D E F D E C G F E (D)2 C

R: 1 (4)6 (1)2 2 2. 4 1

O: **Example 1**

tonal(-1).

temp(12,8).

bar([[1, inst_id], [[f(4,e)], [g(4,e)], [f(4,e)], [c(5,q)], [a(4,e)],
[f(4,e)], [e(4,e)], [f(4,e)], [r(,q)], [f(4,e)]]).

bar([[2], ints_id], [[g(4,e)], [f(4,e)], [g(4,e)], [c(4,q)], [g(4,e)],
[a(4,e)], [g(4,e)], [f(4,e)], ...

* This shows the format in database. Input is done using interactive system.

Example 2

tempo(8,4).

bar([[1], voice], [[c(2,w)], [d(2,q)], [e(2,q)], [f(2,q)], [d(2,q)]]).

bar([[2], voice], [[e(2,q)], [c(2,q)], [g(2,w)], [f(2),tie_start]]).

bar([[3], voice], [[f(2,h),tie_end], [e(2,h)], [d(2,dh)], [d(2,e)]]).

bar([[4], voice], [[c(2,w)], ...

P: **Example 1**

K = b T = 12/8 F+1 QJ G F C+2 CR A+2 QJ F+1 E F R CR

Example 2 not encoded

Q: **Example 1**

23!G !K1- !M12:8 (28ED 30ED 29ED) 33QD 31ED etc.

!G !K- !M12:8 (9 30 9) 33Q 31E (9 8 9) RQ 9 / (30 9 30) 26Q 30E (31 30 9)

!G !K- !M12:8 9E(10 9) 13Q 11E 9(8 9) RQ 9 / 10(9 10) 6Q 10E 11(10 9)

!G !K1- !M12:8 (9 30 9) 33Q 31E (9 8 9) RQ 9E / (30 9 30) 6Q 30E (31 30 9)

* DARMS encoded by Harry Lincoln, Bruce McLean, Stephen Dydo and Stefan Bauer-Mengelberg

Example 2

!F !MC| !&, 4W 5Q 6 7 5 / 6 4 8W 7HJ / 7 6 5. Q / 4W &,@Glo-,~/2WH.,ri-,~Q,a$ &$

!F !MC/ 4W,@Glo-$ 5Q 6 7 5 / 6 4 8W 7HJ / H 6 5. Q,@ri-$ / 4W,@a$

!F !M 4W,@ GLO-$ 5QD 6 7 5D / 6 4 8W 7HJ / 7 6 5.D QD,@RI-$ / 4W,@A$

* DARMS encoded by McLean, Dydo and Bauer-Mengelberg

Letter Q shows four different ways to encode the musical examples using DARMS. The design of DARMS actually permits many types of shorthand representation. The version submitted by Harry Lincoln for example 1 illustrates the unabridged or "canonical" version of the DARMS code. The other versions represent various possible condensations of this code. DARMS also offers considerable flexibility in the order of data entry. For the second musical example, Bruce McLean has chosen to enter the text after encoding the complete musical phrase, whereas Stephen Dydo and Stefan Bauer-Mengelberg have chosen to enter the text simultaneously with the music.

The flexibility of DARMS, as illustrated in these examples, is one of its strong points. However, the variety of representations made possible by this feature has made it difficult to write application programs which accept all versions of DARMS. Bruce McLean has been working on a program which will solve this problem by converting all shorthand and multiple pass representations into a three-dimensional data structure, which can then be collapsed down into the canonical version of the language.

Most of the other systems shown above fall into the category of music *codes* rather than music *languages*. They reflect the individual requirements of their designers and for the most part contain only the information needed for specific projects [see comments under F and H]. In some cases, researchers are interested only in melodic contours and ignore rhythmic attributes such as duration, rests and repeated notes [systems A, C and E]. Some encoding systems deal only with pitch class and ignore register or range [systems C and I]. Some systems are concerned only with absolute duration [H and N], while other systems are interested in the structure of ties [systems B, D, F, G, I, K, M, O, Q]. Some systems show a further interest in note groupings for beaming or other purposes [systems D, M and Q]. System D is particularly noteworthy in this respect, since in the interest of representing a particular rhythmic pattern, it actually inserts a tie (2nd example, 2nd measure) into the representation.

The attributes of pitch and duration are present in most music representation systems. There is, however, a wide variation in the way systems represent these basic attributes. This fact is well illustrated in our set of examples. Here we see two broad categories of pitch representation; representation by the musical letters, A -- G, and representation by numbers. Most of the systems that use letters [systems B, D, F, H, J, K, M, N and O] also indicate a register. This may be done by indications of register change [systems K and P], by repetition of characters [systems D, F and M], or by numbers [systems B, H, J, N, O]. The most common method of representing pitch by number is in relation to a "tonic" pitch [systems A, C, E and G]. The DARMS system represents pitch as a number relative to the staff lines.

For duration, there are at least six methods of representation; (a) letters indicating note types [systems O, P and Q], (b) numbers indicating note types, e.g. 0 = breve, 1 = whole note, etc. [no examples here], (c) numbers indicating inversions of the division of the whole note [systems F, J, M and N], (d) duration as a multiple of a fixed unit [systems B, I and K], (e) duration as a multiple of a variable unit [systems D, G and L], and (f) duration measured in absolute time values. For types d and e, the multiple may be indicated by a number or by other combinations of signs.

Most of the systems above employ some sort of shorthand for pitch and/or duration. Only two of the systems [L and Q] accommodate the presence of text. This is a reflection of the practical nature of these encoding systems and their focus on particular applications.

Internal Representations

The Center received several responses to its request for descriptions of internal representations of musical data. In most cases, respondents chose their own examples to illustrate their systems, rather than using the music printing examples sent out by the Center. This proved to be an advantage, since the examples sent in actually covered a greater range of music representation problems than the examples chosen by the Center. To illustrate some of the methods and techniques of representation currently in use, seven examples with musical excerpts and corresponding internal representations are presented. In some cases, an encoded input string or a second representation is also given. All illustrations of music and encoding are located at the end of the article.

It is important to make a distinction between musical encoding schemes and the input process itself. A stream of input code, once it has been entered into the computer, is actually a form of internal representation.

Internal representations can be divided into four broad classifications. For reference purposes, these may called music/ASCII-code, music/logical, music/parametric, and music/graphic. A possible fifth category, music/acoustic, is outside the scope of the current discussion. The distinction between these classifications is not totally precise, since the representation of musical notation in the computer is itself an abstract concept. The above classification is really an attempt to provide a framework for thinking about the various stages of musical data representation as the data moves from the input process to the various output processes of display, printing, analysis and sound.

The first classification, *music/ASCII-code*, covers musical encoding systems of the type discussed earlier in this article. What makes encoding schemes different from other kinds of internal representation is that most encoding schemes take the form of unstructured character strings. The *music/logical* form of representation is best described as an encoding scheme that has been organized into logical records. In this form, it closely represents the logical meaning of the musical score itself. The next stage of representation, *music/parametric*, includes all the information of the music/logical representation but presents this data as a list of objects (notes, rests, beams, etc.) whose attributes are described in terms of specified parameters. Data in this form is well suited for editing, since the parameters are easy to get at and change. Most music printing programs process their data in this form. The final stage (when moving toward music printing) is *music/graphic*. This form of representation is closely related to the actual printing process. Examples might include font-lists with X-Y coordinates or strings of redefined ASCII characters which translate directly into music graphics.

The first musical example, submitted by Stephen Dydo, is a two-measure excerpt of keyboard music with some unusual rhythmic properties. The first measure sets eighth-note triplets against regular eighth notes. The second measure has a ratio of five written eights in the time of three metric eights. The excerpt is represented as a DARMS string and also in music/parametric form. The parameters are closely related to the DARMS code. The first parameter signals what kind of musical object is being described. E = eighth note; Q = quarter note, H = half note, / = bar line, & = marker and R = ratio designator. The parameters for notes include the horizontal coordinate, the vertical position on the staff, accidental signs (if any) and their horizontal placement, stem direction and beaming information.

The second example, representing the SCORE system of Leland Smith, shows the first two measures of J. S. Bach's sonata for solo violin in A minor, BWV 1003. Special problems of interest in this example include the shifting of various characters to aviod overstriking, the complexity of the beam structures, the placement of five thirty-second notes in the time of one eighth (measure 1) and the shape and placement of the slurs. The input code is given for the top line only and consists of three passes: for pitch, for duration, and for beams and ties. Like the previous example, the internal representation is of the music/parametric type. The first parameter indicates the nature of the musical object being described: 1 = note, 5 = slur or tie, and 6 = beam. In the case of notes, the remaining parameters provide information about staff number, the horizontal displacement, the vertical position, the stem direction and accidental, the type of note (white or black), the note's rhythmic value (1.00 = quarter note), the extension of stem length, and other types of descriptive information. Parameters for beams indicate start and stop points as well as the number and placement of secondary beams.

The third example, submitted by the Ohteru group at Waseda University in Japan, includes such special features as fingerings for the notes and indications of dynamics and articulation. The representation belongs to the music/logical type. It consists of a list of data records organized according to the logical structure of the music. In comparison to the previous two examples, the representation provides less descriptive information about the placement of musical notes, beams and slurs. The computer would be expected to calculate this extra information as part of the process of printing the musical example. In so doing, a parametric list similar to those shown above would be created.

The fourth example is one of the musical excerpts sent out for music printing by the Center. The internal representation, as well as the printed result, were offered by Armando Dal Molin. According to Dal Molin's description, the input process is based on a protocol called PCS (for Pitch, Character, Space), where the Y-coordinate of a notational object is provided by the "pitch attribute," and information on the object itself is provided by the character. The X-coordinate is advanced to the right by successive taps of the space bar; one tap for the smallest metric unit of the line, two taps for the next higher value and so on for each increment of meter. The internal representation achieved by this method is extremely compact. It is also difficult to decipher, if one is unfamiliar with the system. The representation shown in the example is a "hex dump," with each pair of characters representing one byte of information.

The fifth example, provided by Kurt Hebel at the University of Illinois, shows a representation in the OPAL music language, which was developed at Illinois as part of the Interactive Music System. This language, which is used to create, edit and prepare music for printing or for playback, can also function as a representation system on its own. The representation is a good example of the music/logical type. It is versatile and compact, easy to read and understand. The program that prints music from the OPAL representation makes several passes through the data, gathering information and storing it in temporary nodes linked to the main data structure. This process is equivalent to building a parametric data list for the various musical events to be represented.

The sixth example shows two of the internal representations in use at CCARH. The representations are for pitch and duration only. The first representation, a music/logical type, shows the first seven measures of Bach's fugue in E-flat, BWV 852, from the first book of the *Well-Tempered Clavier.* The three voices of the fugue are stored in separate files. Each file has its own "header" containing information about the piece and defining the internal parameters of the file. As a data structure, the design is quite crude, but the information is clearly represented and easy to read and edit, if necessary. The second representation, which is of the music/parametric type and is for the top voice only, was generated directly from the data on the preceding page as a first step to printing out the top voice as shown.

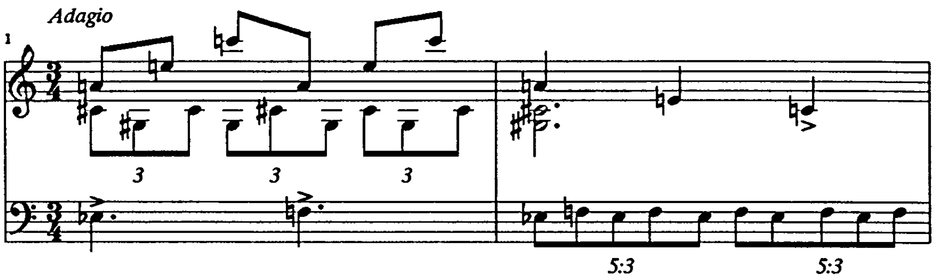
The seventh example, provided by Lelio Camilleri, shows the TELETAU representation for the Bach fugue used in example 6. Like the previous system, the representation covers pitch and duration only. While the musical data represented in the two systems are identical, the TELETAU representation does not organize the data into logical records based on the structure of the music itself. It is basically a series of ASCII strings, broken up into fixed lengths and terminated by semicolons. It is a good example of the music/ASCII-code type of representation.

The representation systems described above are but a small sample of the work that has been done in this area. The MIPS committee [see p. 77], which has been convened to examine various aspects of the problem of the communication of musical information in machine-readable form, has compiled an impressive library of publications and other documents on the subject. The problem of communication does not seem as severe for music scholars as it does for publishers and commercial system and software developers. This is because scholars are more willing to share information about how their systems work. The amount of information on representation collected by the Center and reported in this article is a testimony to scholars' willingness to explain their procedures. The principles of music representation for most applications of interest to scholars are actually quite simple and straightforward, as the examples of encoding systems show. For more advanced applications such as music printing, such open systems as DARMS and MUSTRAN provide excellent vehicles for the input and communication of music. Only a small percentage of respondents to the Center's questionnaire indicated the use of commercial software for their research. While the work of the MIPS committee is extremely important and the benefits to be gained from a standard language for music representation are tremendous, scholars fortunately do not need to wait for this work to be completed before attempting to exchange musical data. Current methods of representation, when adequately explained, will serve the purpose admirably.

Illustration 1

Music Representation -- 1
DARMS

Musical example:



Input code:

```
!I1 !G,!F !M3:4,18@\40\Adagio$ !& !U 4*E( 8*) 13*( 4) 8( 13) / 4*Q 1* -1*> /
& !D !R3 -1#E( -4#,-14@\66\3$ -1) -4( -1#,-14@3$ -4) -1( -4,-14@3$
-1) $R / -1#H.,-4#. / $& $D !-50
!M3:4 6-Q.> 7*Q.> / !R5 6-E( 7* 6,-4@5:3$ 7 6) 7( 6 7,-4@5:3$ 6 7) /
```

Principles of encoding:

1. Pitch and register: in DARMS pitch is encoded purely by identifying (a) clef and (b) staff position [bottom line = 21, top line = 29]. The tens digit is commonly suppressed.
2. Duration: represented by capital letters [W = whole, H = half, R = rest].

Information and example provided by Stephen Dydo

Internal representation:

```
        SAM3.SPA
  1:   999   |     257|   0    0|   1    0|   G    3|     1  0|   1440
  2:   M     |      68|   0    0|   0    0|   3    0|   4    0|  0    0
  3:         a|     88|  18    0|   A    d|   a    g|   i    o|
  4:   &     |       0|   0    0|        0|   0    0|   1    0|     0
  5:   E    N|     136|   4    0|   *   20|  85    0|   1    0|       0
  6:   E    N|     246|   8    0|   *   20|  85    0|  -1    0|       0
  7:   E    N|     348|  13    0|   *   20|  85    0|   1    0|       0
  8:   E    N|     458|   4    0|        0|  85    0|  -1    0|       0
  9:   E    N|     560|   8    0|        0|  85    0|   1    0|       0
 10:   E    N|     662|  13    0|        0|  85    0|  -1    0|       0
 11:   /     |     764| 999    0|   0    0| -50    0|        2|  0    0
 12:   Q    N|     824|   4    0|   *   20|  85    0|   0    0|       0
 13:   Q    N|    1033|   1    0|   *   20|  85    0|   0    0|       0
 14:   Q    N|    1241|  -1    0|   *   20|  85    0|   0    0|       0
 15:   O     |    1241| 999    0|   >    |         |         |
 16:   /     |    1440| 999    0|   0    0| -50    0|        3|  0    0
 17:   &     |       0|   0    0|        0|   0    0|   1    0|     0
 18:   R     |       2|   3    0|        0|   0    0|   0    0|       0
 19:   E    N|     136|  -1    0|   #   20|  68    0|   1    0|       0
 20:   E    N|     203|  -4    0|   #   20|  68    0|   0    0|       0
 21:        a|     203| -14    0|   3    |         |         |
 22:   E    N|     289|  -1    0|        0|  68    0|  -1    0|       0
 23:   E    N|     348|  -4    0|        0|  68    0|   1    0|       0
 24:   E    N|     415|  -1    0|   #   20|  68    0|   0    0|       0
 25:        a|     415| -14    0|   3    |         |         |
 26:   E    N|     501|  -4    0|        0|  68    0|  -1    0|       0
 27:   E    N|     560|  -1    0|        0|  68    0|   1    0|       0
 28:   E    N|     619|  -4    0|        0|  68    0|   0    0|       0
 29:        a|     619| -14    0|   3    |         |         |
 30:   E    N|     705|  -1    0|        0|  68    0|  -1    0|       0
 31:  'R     |       1|   1    0|        0|   0    0|   0    0|       0
 32:   /     |     764| 999    0|   0    0| -50    0|        2|  0    0
 33:   H    N|     824|  -4    0|   #   36|  68    0|   0    0|       1
 34:  -H    N|     824|  -1    0|   #   20|  68    3|   0    0|  0
 35:   /     |    1440| 999    0|   0    0| -50    0|        3|  0    0
 36:   999   |     513|   0    0|   1    0|   F    7|     1  0|   1440
 37:   M     |      68|   0    0|   0    0|   3    0|   4    0|  0    0
 38:   Q    N|     136|   6    0|   -   20|   0    0|   0    0|       1
 39:   O     |     136| 999    0|   >    |         |         |
 40:   Q    N|     458|   7    0|   *   20|   0    0|   0    0|       1
 41:   O     |     458| 999    0|   >    |         |         |
 42:   /     |     764| 999    0|  50    0|   0    0|        2|  0    0
 43:   R     |       3|   5    0|        0|   0    0|   0    0|       0
 44:   E    N|     824|   6    0|   -   20|  68    0|   1    0|       0
 45:   E    N|     889|   7    0|   *   20|  68    0|   0    0|       0
 46:   E    N|     946|   6    0|        0|  68    0|   0    0|       0
 47:        a|     946|  -4    0|   5   :|   3     |         |
 48:   E    N|    1003|   7    0|        0|  68    0|   0    0|       0
 49:   E    N|    1080|   6    0|        0|  68    0|  -1    0|       0
 50:   E    N|    1137|   7    0|        0|  68    0|   1    0|       0
 51:   E    N|    1194|   6    0|        0|  68    0|   0    0|       0
 52:   E    N|    1271|   7    0|        0|  68    0|   0    0|       0
 53:        a|    1271|  -4    0|   5   :|   3     |         |
 54:   E    N|    1328|   6    0|        0|  68    0|   0    0|       0
 55:   E    N|    1384|   7    0|        0|  68    0|  -1    0|       0
 56:   /     |    1440| 999    0|  50    0|   0    0|        3|  0    0
```

Illustration 2

Music Representation -- 2
SCORE

Musical example:

J. S. Bach

Grave

Input code:

```
IN 1 0 0 .8
2 200 1
su/TR/T99 1/PA3:A4:C:E/GS3/F5/E/D/C/B/A//GS/A//B/C/D/E//M/
F/E/D/C/B//A/GS/A/GS/FS/E/DN/FN/E/D/C/D/B/M;
Q/32///64//S////E/S/40X5/S/S./32/S/32//E/
32//S/E/S//S./32X4/64/;
TR 7 25 28;
2B;
3 8/11 +12/13 18/22 23/24 +25/29 30/32 37;
```

Principles of encoding:

1. Pitch and register: pitch and register are encoded using letter name and octave number respectively.
2. Duration: represented in a separate pass (here, lines 5 and 6) using both letters and numbers (Q = quarter, 32 = 32nd note), with a '/' for repetition.

Information and example provided by Leland Smith

Internal representation:

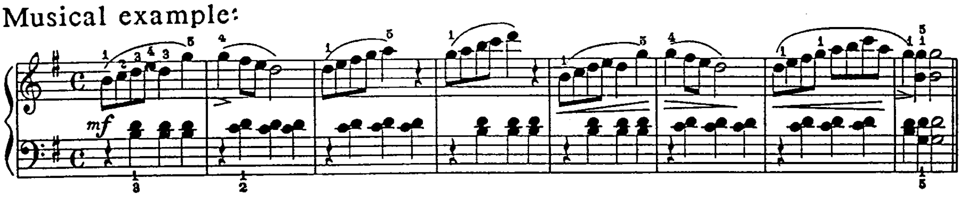
```
8. 3.  .000   7.00   .60
3. 3.  1.376
18. 3.  8.523   .00   99.00   1.00
1. 3.  13.450   -1.00   10.00   .00   1.00   .00   .00   1.30
1. 3.  13.450   6.00   10.00   .00   .00   .00   .00   .90
1. 3.  13.450   8.00   10.00   .00   .00   .00   .00   .40
1. 3.  13.450   10.00   10.00
1. 3.  24.243   -2.00   12.00   .00   .13   15.20   .00
6. 3.  24.243   13.20   13.20   46.80   12.00   .00   .00   21.00   24.24   38.40   31.00   35.70   38.40
1. 3.  28.061   11.00   10.00   .00   .13   2.20   .00
5. 3.  28.061   8.00   4.00   46.80   -3.10   -1.00
1. 3.  31.880   10.00   10.00   .00   .13   3.20   .00
1. 3.  35.699   9.00   10.00   .00   .06   4.20   .00
1. 3.  38.399   8.00   10.00   .00   .06   5.20   .00
1. 3.  41.099   7.00   10.00   .00   .25   6.20   .00   .00   20.00
1. 3.  46.796   6.00   10.00   .00   .25   7.20   .00
1. 3.  52.492   6.00   10.00   .00   .25   .00   .00
6. 3.  52.492   6.00   6.00   63.92   11.00   .00   .00   11.00   52.49   58.22
1. 3.  52.492   4.00   20.00   .00   .00   -.50   1.00
1. 3.  58.224   5.00   12.00   .00   .25   1.00   .00
1. 3.  63.920   6.00   10.00   .00   .50   .00   .00
5. 3.  63.920   6.60   6.60   72.43   1.19   -2.00
2. 3.  63.920   -6.00   1.00
1. 3.  72.434   6.00   10.00   .00   .25   4.60   .00
6. 3.  72.434   10.60   10.60   94.93   12.00   .00   .00   21.00   78.13   91.57
1. 3.  72.434   3.00   20.00   .00   .00   -.50   1.00
1. 3.  78.131   6.00   10.00   .00   .10   4.60   .00
5. 3.  78.131   4.00   8.00   94.93   -1.75   -1.00
1. 3.  81.490   7.00   10.00   .00   .10   3.60   .00
1. 3.  84.849   8.00   10.00   .00   .10   2.60   .00
1. 3.  88.208   9.00   10.00   .00   .10   1.60   .00
1. 3.  91.567   10.00   10.00   .00   .10   .60   .00
1. 3.  94.926   10.00   10.00   .00   .25   .60   .00
14. 3. 100.644   1.00
1. 3. 104.514   11.00   10.00   .00   .38   1.60   10.13
6. 3. 104.514   12.60   8.60   125.10   12.00   .00   .00   21.00   .00   111.76   21.00   121.28   125.10
1. 3. 104.514   2.00   20.00   .00   .00   -.50   1.00
1. 3. 104.514   6.00   10.00   .00   .00   -.50   1.00   .80
1. 3. 111.764   10.00   10.00   .00   .13   1.19   .00
1. 3. 115.583   9.00   10.00   .00   .25   1.45   .00
2. 3. 115.583   -6.00   1.00
1. 3. 121.280   8.00   10.00   .00   .13   1.34   .00
5. 3. 121.280   5.60   4.30   125.10   -1.21   -1.00
1. 3. 125.099   7.00   10.00   .00   .13   1.60   .00
5. 3. 128.610   8.50   7.80   138.71   1.13   -2.00
1. 3. 128.917   7.00   10.00   .00   .50   1.60   .00
6. 3. 128.917   8.60   7.60   145.45   11.00   .00   .00   11.00   137.43   145.45   21.00   137.43   141.63
1. 3. 128.917   2.00   22.00   .00   .00   -.50   1.00
1. 3. 137.431   6.00   10.00   .00   .13   2.09   .00   .00   20.00
2. 3. 137.431   -6.00   1.00
1. 3. 141.631   5.00   12.00   .00   .13   2.83   .00
1. 3. 145.450   6.00   10.00   .00   .25   1.60   .00
1. 3. 151.181   5.00   12.00   .00   .50   .00   .00   .00   20.00
6. 3. 151.181   5.00   3.00   165.43   11.00   .00   .00   11.00   159.73   165.43
1. 3. 159.730   4.00   12.00   .00   .25   -.20   .00
5. 3. 159.730   2.00   1.00   165.43   -1.23   -1.00
1. 3. 165.427   3.00   10.00   .00   .25   .00   .00
1. 3. 171.158   2.00   13.00   .00   .38   4.20   10.00
```

Illustration 3

Music Representation -- 3
Waseda University:
Automatic Translation of Printed Music into Braille

Waseda University has two representation systems. The one shown here from the Braille music system, serves the functions of performance, score display, and music editing.

Musical example:



Internal representation:

```
/* Note: Words between '/*' and '*/' are remarks. */
/* Characters 'p' and 'm' are indicators for 'parallel' and 'measure' */
/* 00 = (........), ff = (11111111) */

        /* Every parallel consist of 1 bar, 2 line */
p m 00 00                  /* 1st bar */
        a0 01 01 00 00              /* G clef */
        a0 02 01 00 00              /* Key signature */
        a0 03 00 01 00              /* time signature */
        c0 11 00 00 00              /* mezzo forte */
        b0 11 01 00 00              /* beginning of slur */
        47 11 00 00 (..1.....)      /* 4th octave, B: 8th note: finger 1 */
        51 11 00 00 (.1......)      /* 5th octave, C: 8th note: finger 2 */
        52 11 00 00 (.11.....)      /* 5th octave, D: 8th note: finger 3 */
        53 11 00 00 (1.......)      /* 5th octave, E: 8th note: finger 4 */
        52 12 00 00 (.11.....)      /* 5th octave, D: 4th note: finger 3 */
        b0 11 80 00 00              /* end of slur */
        55 12 00 00 (1.1.....)      /* 5th octave, G: 4th note: finger 5 */
  m 00 01
        a0 01 03 00 00
        a0 02 01 00 00
        a0 03 00 01 00
        00 12 00 00 00              /* quarter rest */
        37 12 80 00 (.11.....)      /* 3rd octave, B: quarter chord: finger 3 */
        42 12 00 00 (..1.....)      /* 4th octave, D: quarter note: finger 1 */
        37 12 80 00 (........)      /* 3rd octave, B: quarter chord */
        42 12 00 00 (........)      /* 4th octave, D: quarter note */
        37 12 80 00 (........)      /* 3rd octave, B: quarter chord */
        42 12 00 00 (........)      /* 4th octave, D: quarter note */
p m 00 00                  /* 2nd bar */
        a0 01 01 00 00
        a0 02 01 00 00
        b0 11 01 00 00
        55 12 00 (..1.....) (1.......)   /* note with accent */
        54 11 00 00 (........)
        53 11 00 00 (........)
        b0 11 80 00 00
        52 14 00 00 (........)
  m 00 01
        a0 01 03 00 00
        a0 02 01 00 00
        00 12 00 00 00
        41 12 80 00 (.1......)
        42 12 00 00 (..1.....)
        41 12 80 00 (........)
        42 12 00 00 (........)
        41 12 80 00 (........)
        42 12 00 00 (........)
```


Example and information supplied by the Ohteru group (Waseda University)

Illustration 4

Music Representation -- 4
Musicomp Terminal Entry System

Musical example:

Internal representation:

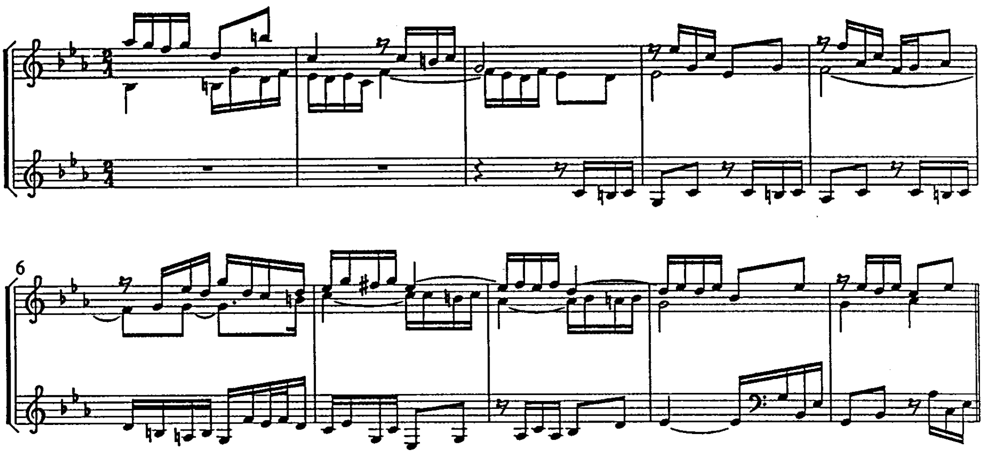
```
1B2F:8000  08 21 99 F0 95 68 81 20-98 68 81 20 99 5E 99 F0
1B2F:8010  95 68 81 20 98 68 81 20-99 5E 99 F2 94 68 81 20
1B2F:8020  97 68 81 20 99 5E 99 F1-93 68 81 20 96 68 81 20
1B2F:8030  99 5E 96 1E 99 24 99 24-81 20 A3 7F A3 1B 51 55
1B2F:8040  41 52 54 45 54 1B 81 20-94 7F 94 1B 56 69 6F 6C
1B2F:8050  69 6E 20 49 1B 81 20 98-2D 9B 6D 82 20 98 6D 82
1B2F:8060  20 9D 6D 82 20 98 6D 82-20 9F 6D 82 20 98 6D 98
1B2F:8070  2D 82 20 99 7C 99 70 82-20 98 2D 9D 6D 82 20 98
1B2F:8080  6D 82 20 9B 6D 82 20 98-6D 82 20 98 6D 82 20 98
1B2F:8090  6D 98 2D 82 20 99 7C 99-70 82 20 98 2D 9B 6D 82
1B2F:80A0  20 98 6D 82 20 9D 6D 82-20 98 6D 82 20 9F 6D 82
1B2F:80B0  20 98 6D 98 2D 82 20 99-7C 99 70 82 20 98 2D 9D
1B2F:80C0  6D 82 20 98 6D 82 20 9B-6D 82 20 98 6D 82 20 98
1B2F:80D0  6D 82 20 98 6D 98 2D 82-20 99 7C 99 70 82 20 98
1B2F:80E0  2D 9B 6D 82 20 98 6D 82-20 9D 6D 82 20 98 6D 82
1B2F:80F0  20 9F 6D 82 20 98 6D 98-2D 82 20 98 7C 99 70 99
1B2F:8100  0D 9A 24 81 20 9A 7F 94-1B 56 69 6F 6C 69 6E 20
1B2F:8110  49 49 1B 81 20 81 20 92-2C 82 20 82 20 96 72 82
1B2F:8120  20 92 3D 92 2C 82 20 92-2C 82 20 92 2C 92 3D 82
1B2F:8130  20 99 7C 9A 70 82 20 98-3D 94 2C 82 20 92 2C 82
1B2F:8140  20 96 2C 82 20 92 2C 82-20 98 2C 82 20 92 2C 92
1B2F:8150  3D 82 20 99 7C 9A 70 82-20 96 3D 96 2C 82 20 92
1B2F:8160  2C 82 20 94 2C 82 20 92-2C 82 20 92 2C 82 20 92
1B2F:8170  2C 92 3D 82 20 92 7C 9C-70 82 20 98 3D 94 2C 82

1B2F:8180  20 92 2C 82 20 96 2C 82-20 92 2C 82 20 98 2C 82
1B2F:8190  20 92 2C 92 3D 82 20 92-7C 9A 70 82 20 96 3D 96
1B2F:81A0  2C 82 20 92 2C 82 20 94-2C 82 20 92 2C 82 20 92
1B2F:81B0  2C 82 20 92 2C 92 3D 82-20 92 7C 9A 70 99 0D 9A
1B2F:81C0  24 81 20 9A 7F 94 1B 20-56 69 6F 6C 61 1B 81 20
1B2F:81D0  81 20 95 6D 82 20 82 20-95 79 82 20 82 20 95 79
1B2F:81E0  82 20 82 20 95 7C 9A 70-82 20 91 2C 82 20 82 20
1B2F:81F0  93 2C 82 20 82 20 95 6D-82 20 82 20 95 7C 9A 70
1B2F:8200  82 20 93 2C 82 20 82 20-91 2C 82 20 82 20 95 79
```

Example and hexadecimal representation supplied by Armando Dal Molin

Illustration 5

Music Representation -- 5
Interactive Music System

Musical example:



Principles of representation:
The music is entered from an electronic keyboard with a metronome. The resulting file contains data that has been automatically interpreted. Each "event" has three components: octave specification (numerical), pitch name (lower case letter), and duration (numerical). Numbers are not reiterated when there is no change from the preceding event.

Information and example provided by Kurt Hebel

OPAL source code:

```
staff    s
staff    a
staff    b
s:       treble key c minor
a:       treble key c minor
b:       treble key c minor
time     2,4
tempo    72,4
*
*
measure 1
format   systems 1,6
print    (s,a),b
s:       5a16 g f g d8 bn
a:       3b4 bn16 4g d f
b:       r2
measure 2
s:       5c4 r16 c 4bn 5c
a:       4e16 d e c f4t
b:       r2
measure 3
s:       4g2
a:       4f16 e d f e8 d
b:       r4 r16 4c 3bn 4c
measure 4
s:       r16 5e 4g 5c 4e8 g
a:       4e2
b:       3g8 4c r16 c 3bn 4c
measure 5
s:       r16 5f 4a 5c 4f g a8
a:       4f2t
b:       3a8 4c r16 c 3bn 4c
measure 6
s:       r16 4g 5e d g d c d
a:       4f8 gt g8d bn16
b:       4d16 3bn an b g 4f e32 f d16
measure 7
s:       5e16 g f+ g e4t
a:       5c4t c16 c 4bn 5c
b:       4c16 e 3g 4c 3e8 g
measure 8
s:       5e16 f e f d4t
a:       4a4t a16 b an b
b:       r16 3a 4c 3a b8 4d
measure 9
s:       5d16 e d e 4b8 5e
a:       4g2
b:       4e4t e16 bass 3g 2b 3e
measure 10
s:       r16 5e d e c8 e
a:       4g4 a
b:       2g8 b r16 3a c e
```

Illustration 6

Music Representation -- 6
CCARH

Musical example:

Logical representation of keyboard input:

```
852    2
Kalmus/Bisch
off 1883

3 1
37 -3 16 4
4 4 0 1    2
measure 1
Bf4    1
G4     1
F4     1
G4     1
Ef4    1
Af4    1
G4     1
Af4    1
C5     2
Bf4    2
rest   2
A4     1
F4     1
measure 2
Ef5    2
D5     2
C5     4
Bf4    1
F5     1
D5     1
Bf4    1
Af4    1
F5     1
D5     1
Af4    1
measure 3
G4     2
Af5    2
G5     2
F5     2
Ef5    1
C5     1
D5     1
Ef5    1
F5     4-
measure 4
F5     1
```

```
Ef5    1
F5     1
G5     1
Af5    1
F5     1
Bf4    1
Af5    1
G5     2
Ef5    1
G5     1
C6     4-
measure 5
C6     2
D5     1
F5     1
Bf5    6
C5     1
Ef5    1
Af5    2
G5     2
measure 6
F5     4
G5     2
D5     2
Ef5    6
F5     2
measure 7
G5     6
A5     2
Bf5    8-
 .
 .
 .
```

```
852    2
Kalmus/Bisch
off 1883

3 2
37 -3 16 4
4 4 0 1    2
measure 1
rest 16
measure 2
rest 16
measure 3
Ef4    1
D4     1
C4     1
D4     1
Bf3    1
Ef4    1
D4     1
Ef4    1
G4     2
F4     2
rest   2
D4     1
Bf3    1
measure 4
Af4    2
G4     2
F4     4
Ef4    1
Bf4    1
G4     1
F4     1
Ef4    1
C5     1
A4     1
F4     1
measure 5
D4     1
A4     1
F4     1
Ef4    1
Df4    1
Bf4    1
G4     1
```

```
Ef4    1
C4     1
G4     1
Ef4    1
C4     1
D4     1
Bf4    1
Ef4    1
C5     1
measure 6
D5     2
Df5    2
C5     2
Bf4    2
Af4    1
G4     1
Af4    1
Bf4    1
C5     4-
measure 7
C5     1
Bf4    1
C5     1
D5     1
Ef5    1
C5     1
G4     1
Ef5    1
D5     2
F5     2
D5     2
Bf4    2
 .
 .
 .
```

```
852    2
Kalmus/Bisch
off 1883

3 3
37 -3 16 4
4 4 1 1    2
measure 1
rest 16
measure 2
rest 16
measure 3
rest 16
measure 4
rest 16
measure 5
rest 16
measure 6
Bf3    1
G3     1
F3     1
G3     1
Ef3    1
Af3    1
G3     1
Af3    1
C4     2
Bf3    2
rest   2
A3     1
F3     1
measure 7
Ef4    2
D4     2
C4     4
Bf3    1
F4     1
D4     1
Bf3    1
Af3    1
F4     1
D4     1
Af3    1
 .
 .
```

Principles of logical representation:

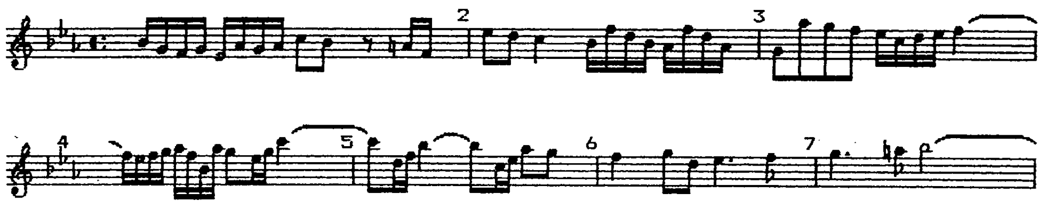
1. Pitch is represented by letter, register by the number immediately following.

2. Representation of duration depends on the meter; here 4 = quarter, 2 = eighth, 1 = sixteenth, etc.

Musical example:

J. S. BACH MUSICAL DATABASE

BWV. #852, Movement 2
Track 1 of 4



Parametric representation of top voice only:

Internal Represention of Measures 1 -- 7

measure 1									
28	0	1	3	0	0	0	2	25	1
26	0	1	3	0	0	0	3	23	1
25	0	1	3	0	0	0	3	22	1
26	0	1	3	0	0	0	1	23	1
24	0	1	3	0	0	0	2	21	1
27	0	1	3	0	0	0	3	24	1
26	0	1	3	0	0	0	3	23	1
27	0	1	3	0	0	0	1	24	1
29	0	2	4	0	1	0	2	26	2
28	0	2	4	0	1	0	1	25	2
100	0	2	4	0	1	0	0	0	0
27	1	1	3	0	0	0	2	24	1
25	0	1	3	0	0	0	1	22	1
measure 2									
31	0	2	4	0	1	0	2	28	2
30	0	2	4	0	1	0	1	27	2
29	0	4	5	0	2	0	0	26	2
28	0	1	3	0	0	0	2	25	2
32	0	1	3	0	0	0	3	29	2
30	0	1	3	0	0	0	3	27	2
28	0	1	3	0	0	0	1	25	2
27	0	1	3	0	0	0	2	24	2
32	0	1	3	0	0	0	3	29	2
30	0	1	3	0	0	0	3	27	2
27	0	1	3	0	0	0	1	24	2
measure 3									
26	0	2	4	0	1	0	2	23	2
34	0	2	4	0	1	0	3	31	2
33	0	2	4	0	1	0	3	30	2
32	0	2	4	0	1	0	1	29	2
31	0	1	3	0	0	0	2	28	2
29	0	1	3	0	0	0	3	26	2
30	0	1	3	0	0	0	3	27	2
31	0	1	3	0	0	0	1	28	2
32	0	4	5	0	2	1	0	29	2
measure 4									
32	0	1	3	0	0	0	2	29	2
31	0	1	3	0	0	0	3	28	2
32	0	1	3	0	0	0	3	29	2
33	0	1	3	0	0	0	1	30	2
34	0	1	3	0	0	0	2	31	2
32	0	1	3	0	0	0	3	29	2
28	0	1	3	0	0	0	3	25	2
34	0	1	3	0	0	0	1	31	2
33	0	2	4	0	1	0	2	30	2
31	0	1	3	0	0	0	3	28	2
33	0	1	3	0	0	0	1	30	2
36	0	4	5	0	2	1	0	33	2
measure 5									
36	0	2	4	0	1	0	2	33	2
30	0	1	3	0	0	0	3	27	2
32	0	1	3	0	0	0	1	29	2
35	0	4	5	0	2	1	0	32	2
35	0	2	4	0	1	0	2	32	2
29	0	1	3	0	0	0	3	26	2
31	0	1	3	0	0	0	1	28	2
34	0	2	4	0	1	0	2	31	2
33	0	2	4	0	1	0	1	30	2
measure 6									
32	0	4	5	0	2	0	0	29	2
33	0	2	4	0	1	0	2	30	2
30	0	2	4	0	1	0	1	27	2
31	0	6	5	1	2	0	0	28	2
32	0	2	4	0	1	0	0	29	2
measure 7									
33	0	6	5	1	2	0	0	30	2
34	1	2	4	0	1	0	0	31	2
35	0	8	6	0	4	1	0	32	2

Principles of parametric representation:

P1 = pitch (diatonic), P2 = accidental, P3 = duration, P4 = note type
P5 = dot, P6 = space code, P7 = tie, P8 = beam code, P9 = position on
staff, P10 = stem direction

Illustration 7

Music Representation -- 7
TELETAU

Musical example:

Input code:

```
DIVISIONE MUSICOLOGICA DEL CNUCE/C.N.R.
CONSERVATORIO DI MUSICA "L. CHERUBINI" FIRENZE
J.S.BACH  - THE WELL TEMPERED CLAVIER - FUGUE NO. 7 VOL. I -
;
;
!1!  T52  V100  -3 4B.24  G F G E A G A 5C.48 4B P  AN.24  F
5E.48 D C.96 4B.24 5F D 4B A 5F D 4A G.48 5A.48 G F E.24 C D
E F1.20 E.24 F G A F 4B 5A G.48 E.24 G 6C1.44 5D.24 F  B1.44
C.24  E A.48 G F.96 G.48 D E1.44 F.48 G1.44 AN.48 B2.16 B.24
G E DF B G DF C.48 E A1.20
;
!2! T52 V100 (P3.84)2 4E.24 D C D 3B 4E D E G.48 F P D.24 3B
4A.48 G F.96 E.24 B G F E 5C 4AN F D AN F E DF B G E C G E C
D B E 5C D.48 DF C 4B A.24 G A B 5C1.20 4B.24 5C D E C 4F 5E
D.48 F D 4B 5E P1.92 E.48 C 4A
;
!3!  T52 V100 (P1.92)10 3B.24 G F G E A G A 4C.48 3B P AN.24
F 4E.48 D C.96  3B.24 4F D 3B A 4F D 3A G.48 4E.48 P 3E A.24
4E C 3A G 4E C 3G
```

Principles of encoding:

1. Pitch by letter; register by number preceding pitch but suppressed when redundant.

2. Duration by number (96 = quarter, 48 = eighth, 24 = sixteenth, etc.) preceded by period; suppressed when redundant.

Information provided by Lelio Camilleri.