Computers, Composition, and the Challenge of "New Music" in Modern India

James Kippen, Bernard Bel

Computers, Composition, and the Challenge of "New Music" in Modern India*

Prof. Jim Kippen
Faculty of Music
University of Toronto — Canada

Bernard Bel
Centre for Human Sciences
2, Aurangzeb road, New Delhi 110 011 — India
Fax (91) 11 301 6441

Introduction

In the early mid-1980s, we developed a computer system called the *Bol Processor* (i.e. "BP") to help us examine improvisatory methods used by North Indian tabla drummers (Kippen & Bel 1992). Designed for the portable Apple™ IIc, the rule-based BP1 was able to analyse musical input (in the form of tabla *bol*s, or onomatopoeic syllables) as well as generate new improvisations that musicians could assess. Details of the philosophy behind BP1 and its *modus operandi* can be found in (Kippen & Bel 1992, Bel & Kippen 1992).

The Bol Processor attracted interest from scholars and musicians alike. It was felt that the formal model embedded in it could be expanded to encompass more general musical structures, and in this form could be of some benefit as a tool for music composition. We therefore decided to implement an alternative version (BP2) of the Bol Processor on the Apple™ Macintosh™.

Initial work on BP2, therefore, was geared towards generalizing concepts inherited from the representational model in BP1. We replaced the concept of *bol*, broadly denoting a "musical gesture", with that of *sound-object*. The idea of sound-object structures lended itself to several developments: polyphony, with *polymetric structures* retaining the notational concepts of Indian rhythm, and an altogether flexible and accurate mapping of *symbolic time* (durations as idealized in musical scores) to physical time.

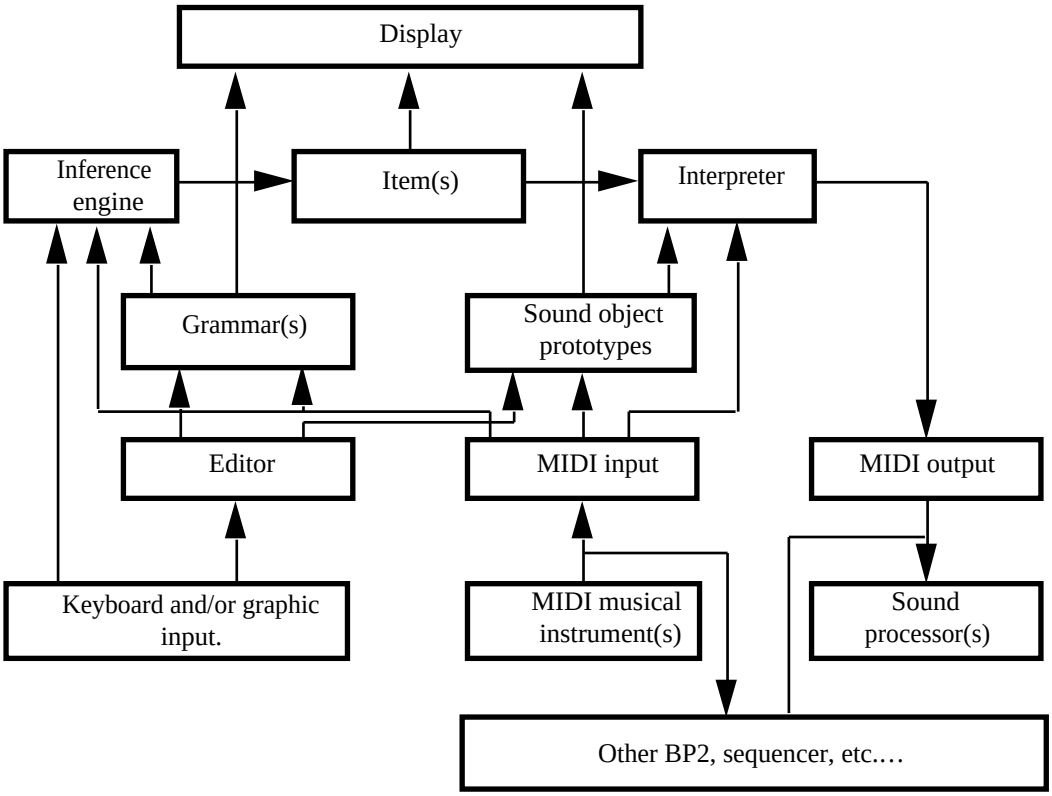
The question of time is crucial in computer-generated music, but its conceptualisation has so far been very poor in most commercial and research music software. In many cases, physical dates and durations are either explicitly entered by musicians or calculated with the aid of rudimentary rules yielding artificial musical output. Nothing is less appealing to ears accustomed to the melodic intricacy of Indian music, than straight-line or exponential *portamenti* generated on simple DSP programs. Similarly, the idea that "humanized" music may be produced with a rigid tempo compensated by randomizing durations and on-setting dates does not hold.

* Published in Leonardo Music Journal, vol. 4. 1994, p. 79-84

BP2 may be seen as an outcome of a “migration of concepts” between Indian and western music / musicological “worlds”, clearly from East to West. The research is now in a new phase because of a starting collaboration with Indian teams.

Some characteristic features of BP2

The following is an outline of the main innovative features of Bol Processor BP2. Developments of the formal language model — procedures for generating strings of symbols mapped to musical events — have been described in (Bel & Kippen 1992) and need not be recalled here. Other features will be summarized on the basis of the following block diagram:



A block diagram of Bol Processor BP2

Three fields are used for storing **grammars**, **items** generated by the **inference engine** and **sound-object prototypes**. These prototypes may be created with the help of a MIDI-interfaced musical instrument (or simply by typing in MIDI codes). Items are represented as structures (strings and sets) of symbols. Each symbol is mapped to a single sound-object prototype. The interpreter generates MIDI codes given the symbolic structure of an item and properties of sound-object prototypes.

Interpretation is in two stages. Musical items (which may be polyphonic) are represented as strings of symbols in a syntactic form that we call a **polymetric expression**. First, a mapping is calculated between the sound-objects contained in a polymetric structure and a set of **symbolic dates**. Symbolic time, here, is an arbitrary ordered set that permits the ordering of sound-objects. This process is called the **interpretation of polymetric expressions**, as missing information regarding the ordering of sound-objects (along

symbolic time) is inferred. Then BP2 proceeds to the **time-setting** of the sound-object structure represented as a complete polymetric expression. Start/clip dates of all sound-objects are calculated, yielding the dates of their MIDI messages. Sound-object properties are taken into account during time-setting only.

The block diagram indicates that an external control can be exerted on the inference engine, grammars and the interpretation module. Specific MIDI messages may be used to change rule weights, the time base and nature of time (*striated/smooth*). These messages may also be used for synchronizing events during their performance and even assigning computation time limits. Such features are currently used in improvisational rule-based composition.

Several BP2's may be linked together and to other software devices such as MIDI sequencers. Messages on the different MIDI channels may be used for communicating between machines or for controlling several sound processors. It must be kept in mind that a "sound-object" is not necessarily a sound-generating process. Depending on the implementation it may contain any kind of control/synchronization messages as well.

Polymetric structures

A major development of BP2's representational model has been *polymetric expressions*, here meaning incomplete string descriptions of concurrent processes (Bel 1992). A polymetric expression can be fully determined (*expanded*) by an algorithm inferring a strict ordering of events (*sound-objects*, see infra) along symbolic time: should a, b, c, d, e be superimposed on another sequence of three sound-objects f, g, h, the algorithm would suggest the following ordering:

```
    {a b c d e, f g h}              a _ _ b _ _ c _ _ d _ _ e _ _
                                    f _ _ _ _ g _ _ _ _ h _ _ _ _

Incomplete representation              Complete representation
 (polymetric expression)               (computed by algorithm)
                                       Symbol '_' prolongates the
                                       preceding sound-object.
```

The complete representation shown above is a *phase diagram*, i.e. a kind of musical score in which simultaneous events are grouped in the same column. The corresponding polymetric expression in a unidimensional string format would be:

{a _ _ b _ _ c _ _ d _ _ e _ _ , f _ _ _ _ g _ _ _ _ h _ _ _ _ }

The same algorithm is able to process multilayered polymetric expressions.

Sound-objects

Informally, a sound-object is a sequence of elementary processes that produce or modify sounds. For instance, in the current version of BP2, a sound-object may represent any sequence of MIDI events. In later versions, other kinds of sound-objects will be defined as segments of digitized sound files, procedures (e.g. granular synthesis) generating MIDI code, or procedures in the Csound format (Vercoe 1994) producing or modifying sound files. To make things clear, a simple note would be a sound-object containing either a NoteOn/NoteOff pair, or the specification of a sampled sound and start/clip dates indicating where the note may be found in that sampled sound.

Sound-objects may be assigned properties used in determining their actual durations and locations on physical time. They may be performed in *striated* time (with regular or irregular beats) or in *smooth* time (with no beat). Durations depend on the "local tempo" and on metrical properties stipulating the acceptable range of object contraction/dilation.

A naive interpretation of sequences of sound-objects would be to arrange their time intervals in a strictly sequential way. Stroppa (1990) suggested a more abstract approach, starting from the assumption that any sound-object may possess one or several time points playing a particular role, e.g. a climax. These points are called *time pivots*. We retained a simplified version of this idea, assigning each object one single pivot.

Let us for instance consider a polymetric structure {S1,S2,S3} derived as

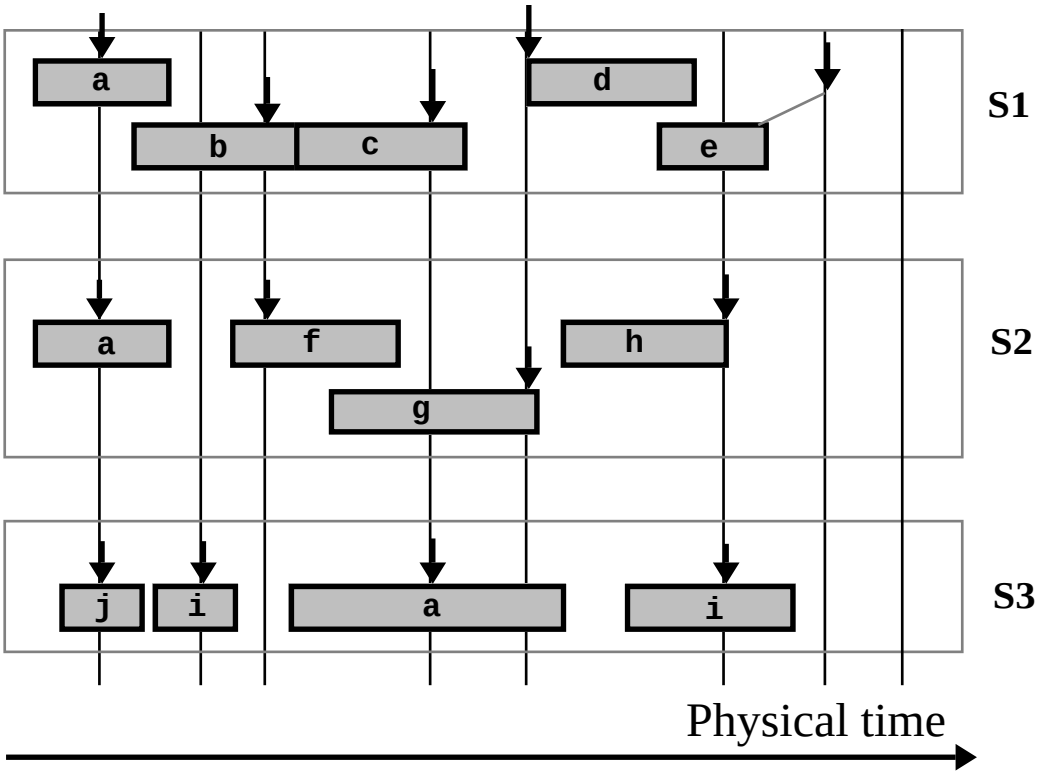
{a _ b c d _ e , a _ f _ g h _ , j i _ a _ i _ }

yielding the phase diagram:

a	_	b	c	d	_	e
a	_	f	_	g	h	_
j	i	_	a	_	i	_

The definition of each object contains the relative location of its pivot and metrical properties allowing the calculation of its time-scale ratio.

The following is a graphic representation of a possible *instance* of this polymetric structure:



A structure of sound-objects

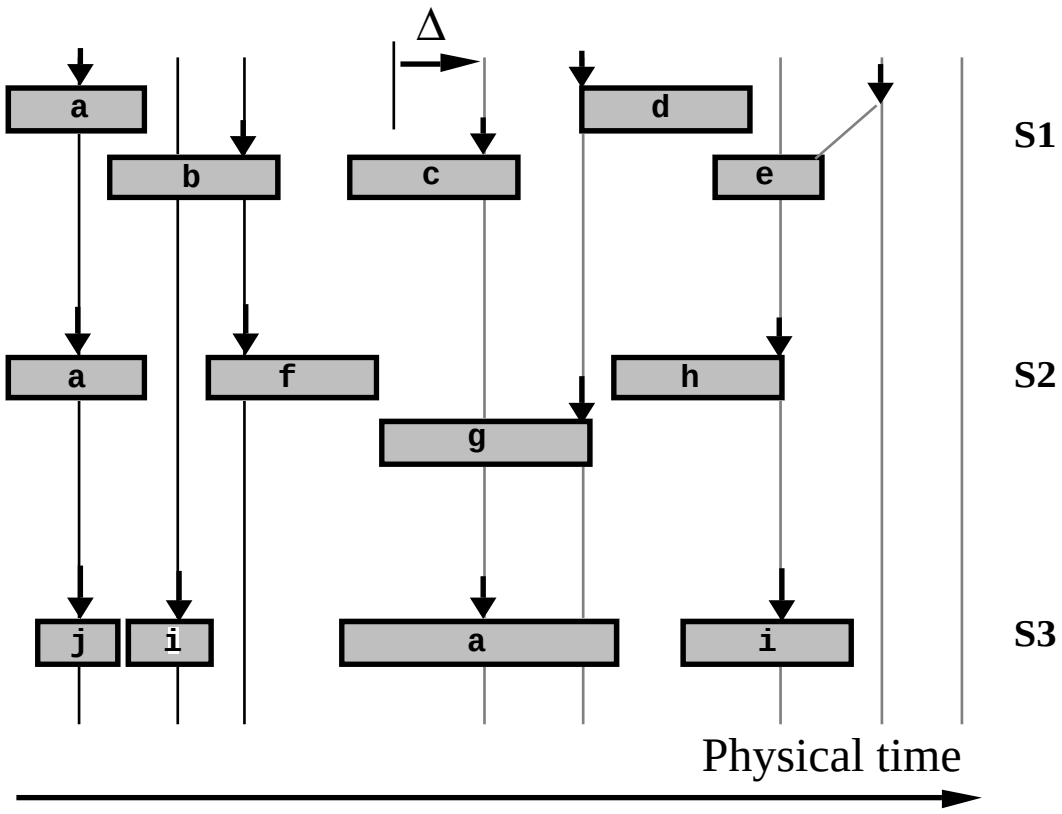
The structure of time is for instance an irregular pulsation represented with vertical lines (time streaks). The time-span interval of each sound-object is shown as a rectangle with arbitrary vertical width and vertical position. These positions have been chosen to separate objects on the graphic: it is clear for example that “c”, “f”, “g” and “a” have overlapping time-span intervals between the third and fourth streaks. Lengths of rectangles represent the physical durations of sound-objects.

Vertical arrows indicate time pivots. As shown with object “e”, the pivot is not necessarily a time point within the time-span interval of the sound-object.

This graphic represents the default positioning of objects with their pivots located exactly on time streaks. Although it is reasonable that instances of “c”, “f” and “a” are overlapping between the third and fourth streaks since they belong to distinct sequences which are performed simultaneously, it may not be acceptable that “f” overlaps “g” in a single sequence S2; the same with “d” and “e” in sequence S1. For similar reasons, it may not be acceptable that the time-span intervals of “j” and “i” are disjoint in sequence S3 while no silence is shown in the symbolic representation.

How could one deal with a constraint such as <<*the end of sound-object “f” may not overlap another sound-object in the same sequence*>> ? If object “g” is relocatable then it may be delayed (shifted to the right) until the constraint is satisfied. We call this a *local drift* of the object. However, the end of “g” will now overlap the beginning of “h”. Assume that this also is not acceptable and “h” is not relocatable. One should therefore look for another solution, for example truncate the beginning of “h”. If this and other solutions are not acceptable then one may try to shift “f” to the left or to truncate its end. In the first case it might be necessary to shift or truncate “a” as well.

So far we mentioned a constraint propagation within one single sequence. In the time-setting algorithm the three sequences are considered in order S1, S2, S3. Suppose that the default positioning of objects in S1 satisfies all constraints and no solution has been found to avoid the overlapping of “f” and “g” in S2. Another option is to envisage a *global drift* to the right of all objects following “f” in S2. The global drift is notated Δ on the following picture. All time streaks following the third one are delayed (see dotted vertical lines).



A structure using global drift

This last solution is similar to the the *organum* in conventional music notation.

The process of locating — i.e. "instantiating" — sound-objects, as briefly illustrated in this example, is the task of an efficient *time-setting algorithm* imbedded in BP2. (Bel 1992).

Time accuracy and flexibility

It is important for a musician working with a computer to be given the ability of manipulating precise durations. In sequencing software this accuracy is generally limited by time quantization and errors cumulated while adding durations. To maintain the highest accuracy (within the one millisecond limit of Macintosh's time manager) throughout a piece of music, the option taken in BP2 has been to represent durations as integer ratios. For instance, we had to create a sequence of intricate polyrhythmic beat patterns matching exactly he opening section of "*La Création du Monde*" by Darius Milhaud (EMI CDC-7 47845 2), on which we wanted to synchronize the recitation of Victor Hugo's poem "*Les Djinns*". The global specification was that the time-base should generate 1000 beats in exactly 1578 seconds. The poem is made of verses with variable meters: 2/2/2/2/3/3/3/3/4/4/4/4 …up to 10/10/10/10 —except 9/9/9/9— and then decreasing. We decided that every verse should keep the same duration within 16 beats. The result was a polyrhythmic piece with one instrument showing beats at a fixed speed and another one playing "ticks" at variable speeds. (Bel 1993a:16)

In the above example it may be argued that absolute accuracy yields extremely complex phase diagrams, due to the fact that the lowest common multiple of 6, 7, 8 and 10 is an extremely large number. However, the dates of events belonging to two consecutive

columns do not differ significantly. Therefore, while keeping a specified accuracy (e.g. 100 milliseconds) it is possible to merge columns and reduce the size of the diagram accordingly. Understandably merging should not occur once the phase diagram has been built, because building it entirely would take a prohibitive amount of memory. We found out that it was possible to build smaller diagrams by simplifying the polymetric expression on the basis of the expected accuracy (quantization) and the current metronomic speed. It should be well understood that this kind of quantization does not decrease the accuracy of long time intervals (e.g. duration of the example piece always remained 1578 seconds) whereas ordinary quantization found in sequencing programs would provoke a cumulated error due to the rounding of basic time units. (Bel 1994)

Flexibility of time intervals is another important feature which, in commercial software, is often limited to the "humanize" option (inserting random "errors" within certain limits) or the "swing" option (entering durations on a MIDI keyboard). BP2 offers the possibility of specifying *time patterns* (ratios applicable to consecutive time intervals) either numerically or from MIDI keyboard input. Each pattern may be assigned specific label to be manipulated *at the symbolic level* by the editor or a grammar. Suppose for instance that the following note sequence is to be played unevenly:

do5 re5 mi5 fa5 - la5 si5 do6_ mi6

(in which "-" is a silence and "_" is the prolongation of do6). We first decide to divide the piece in two sections, the duration of the first part (do5 re5 mi5 fa5 -) being two thirds of the second part (la5 si5 do6_ mi6). The corresponding pattern may be notated:

t1 = 1/1 t2 = 3/2

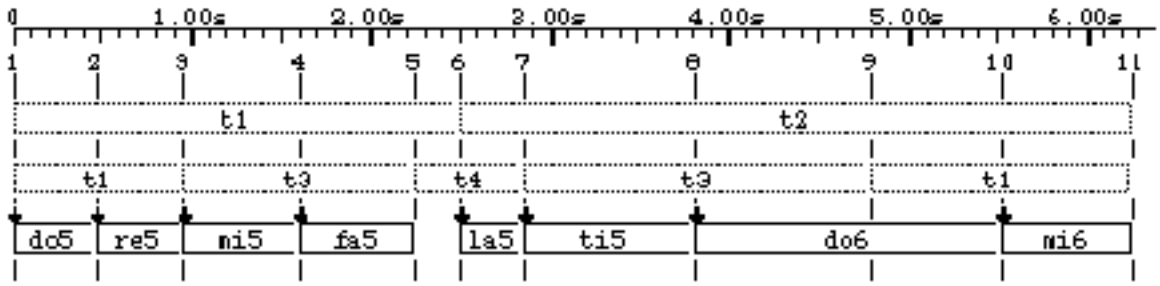
Let the initial part (do5 re5 mi5 fa5) of the first section be further divided into two subsections with pattern

t1 = 1/1 t3 = 4/3

while the following note sequence (- la5 si5 do6_ mi6) is controlled by pattern:

t4 = 1/2 t3 = 4/3 t1 = 1/1

The resulting timing of sound-objects is shown below:



Time pattern with irregular beats

Note that, for the sake of generality, time intervals are not arranged in a tree hierarchy, but as a join-semilattice: t4, for instance, overlaps the two main sections, so that its duration is a combination of the scalings imposed by t1 and t2.

Polymetric representation in BP2 makes it very easy to represent complex time structures. Labels t1, t2, etc., are treated as time-objects (similar to sound-objects except that they do not produce sounds). A polymetric expression representing the sequence of notes with its time patterns is

{10,t1 t2,{t1 t3 t4, do5 re5 mi5 fa5 - la5}{t3 t1, si5 do6 _ mi6}}

in which "10" specifies that the symbolic duration of the entire expression is 10 beats. These beats are forced to become irregular because of time patterns, as indicated by vertical lines numbered 1 to 11 on the above picture.

Present developments

Since February 1994, work on BP2 has been reconsidered in the context of a collaboration between CENTRE FOR HUMAN SCIENCES, the CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING (C-DAC, Pune), and other researchers working at the NATIONAL CENTRE FOR THE PERFORMING ARTS (NCPA, Bombay) and POONA UNIVERSITY.

C-DAC has been engaged on the *Gandharva* project aiming at creating a platform for the composition of melodic music based on Indian *ragas* (Upadhye 1993). The core of the system is a machine-learning device able to construct an automaton from musical input data (note sequences). The infered automaton is able to generate sequences of notes in the same style or raga. An interesting finding is that the predictive power of the learning algorithm could be enhanced by imbedding domain-dependent knowledge based on the principle of consonance (of melodic intervals). Similarly, Prof. H.V. Sahasrabuddhe, in Poona University, devised a learning system using a Markov model. Musical output was improved by "performance rules" able to reconstruct the characteristic note treatments (*alankara*), and the global structure of *bandishes* (compositions) and rhythmic improvisations. These projects are complementary with Bol Processor completed with its learning device QAVAID (Kippen & Bel 1989), and it was decided that Gandharva and Bol Processor would be merged for the sake of producing research, music composition and educational software. At the NCPA, a new version of Bel's *Melodic Movement Analyzer* (MMA) is under development with a particular focus on identifying raga characteristic shapes (Rao 1994), automatic transcription and the resynthesizing of melodic music. This project also relates to the automatic learning of melodic sequences, in that the MMA will produce detailed melodic data (note names and note connections) to be fed into learning devices, while infered automata may later become able to determine the parameters for the shapes of synthesized melodic lines.

While there is no concept of pitch in the present version of BP2, the idea of sound-object will be refined so that it may also represent note *treatment* (melodic connections). This will also make it possible to introduce abstract definitions of scales and arbitrary tunings.

At a later stage, representational models and composition methods will be transferred to a multimedia software in complement to encyclopaedic knowledge about Indian music. In fact, a learning device is very attractive in a multimedia system because it allows users to manipulate and transform the data stored. Thus, while "frozen" musical performances may be stored in the system, it will also be possible to use them as sample sets and let the machine produce music therefrom. Users will also be given the possibility to experiment with mixing data from different sources (performers, ragas, styles) much in the same way strange hybrid pictures are composed by the MORPH program. Experiments with

MORPH indicate that distorting and hybridizing pictures is an excellent way of intuitively grasping their characteristic features.

The cultural and scientific challenge

Indian software engineering is one of the fastest growing industries in the world, with exports reaching $350 million in 1993 —a growth of 55% since 1992— and software teams obtaining the highest ratings on international standards (Yourdon 1994:6). While it is expected that computer generated music will slowly enter in Indian life, there are doubts as to whether technological achievements *per se* are bound to induce similar achievements in modern music. To quote Laske (1993:2),

> If workers in the interpretive study of art do not decide what aspects of their subject matter are fundamental to their efforts, engineers will. This means, one-dimensional, purely explanatory thinking will prevail, to the detriment of understanding aesthetic subject matter.

With the advent of new sound and music processing techniques, the young generation of Indian musicians is expecting to find new environments for composing classical as well as innovative music. However, the software tools that are readily available to professional musicians are based on western musical concepts and extensions thereof.

When faced with these limitations, Indian musicians may either resolve to do plain MIDI sequencing or use digital multitrack editing techniques for vocal and instrumental sounds. In either case the product is not at the level of their own skills as performing musicians: MIDI environment will generally not cope with the melodic and rhythmic intricacies they are familiar with, and studio arrangements do not retain the improvisational and interactive aspects of composition in Indian music. Therefore, there is a gap between the expectations of Indian artists and the response of technology:

> Computer generated music will slowly enter in Indian life as computers entered a couple of years ago. Because if we do not do it, the Americans or the Japanese will do it. In fact they are doing it. The sad part is that the Indian mind which is often impressed by foreign labels is not adaptative to new changes and experiments. I am afraid that tomorrow we will have to depend upon these foreigners to make instruments like Tanpura, Sitar, etc. Are we going to wait for them to certify that Indian music is most easy to generate on computers?
>
> Upadhye, 1993:21,56.

Conclusion

The western approach to modern composition is often described as rule-based and highly conceptual, while the Indian approach is supposed to be model-based and empirical. However, this simplistic dichotomy leads to ethnocentric biases, notably the assumption that creativity is a salient feature of Euro-American culture while non-western cultures are perceived as stagnant and "traditional", thereby meaning unable to innovate. The background prejudice is that western countries should keep a leading role in modern art production while others should content themselves with preserving their cultural heritage. Indeed, it is not easy to create conditions for a new art form to emerge in the context of a traditional society even though there is an increasing social demand for it.

The work of cultural anthropologists during the past decades has shown that compositional knowledge (as defined in western art music) may also be acknowledged among musicians who are not given the status of composers in their own culture (Blacking 1989). (Conversely, it would be important to discover the "ethnicity" of western contemporary art forms.) For instance, the variety and complexity of tabla *qa'ida* improvisation by great masters is an indication that compositional knowledge is

not necessarily a design process involving graphic media nor methods amenable to introspection or verbal description. Therefore, the challenge of contemporary art forms in non-western cultures is to set up interactive environments in which artists may be able to use and expand their creative potential without being confronted with procedures exclusively elaborated in a foreign culture.

In its attempt to link musical practice with theory (the handling of musical *objects* and compositional *processes*), computational musicology is probably the unavoidable "interface" between musicians and software designers. The remaining question is how this interaction may be encouraged in the present Indian social-cultural context.

Bibliography

Bel, Bernard (1992). Symbolic and sonological representations of sound-object structures. In *Understanding Music with AI*, M. Balaban, K. Ebcioglu & O. Laske, Eds., AAAI Press, 1992:64-109.

(1993). *Bol Processor BP2 — QuickStart and Reference Manual.* Available in electronic format along with shareware program from ftp site *ftp.ircam.fr*.

(1994) Rationalizing musical time: syntactic and constraint-satisfaction approaches. *The Ratio Symposium*, Den Haag (The Netherlands). (Forthcoming)

Bel, Bernard, & Jim Kippen (1992). Bol Processor grammars. In *Understanding Music with AI*, M. Balaban, K. Ebcioglu & O. Laske, Eds., AAAI Press, pp.366-401.

Blacking, John (1989). Challenging the myth of 'ethnic' music: first performances of a new song in an African oral tradition, 1961. In *Yearbook for Traditional Music,* 21. New York: International Council for Traditional Music, pp.17-24.

Duthen, Jacques, & Marco Stroppa (1990). Une représentation de structures temporelles par synchronisation de pivots. In *Le fait musical — Sciences, Technologies, Pratiques*, eds. B. Vecchione and B. Bel. Colloque CRSM-MIM "Musique et Assistance Informatique", Marseille, October.

Kippen, Jim, & Bernard Bel (1989). The identification and modelling of a percussion "language", and the emergence of musical concepts in a machine-learning experimental set-up. *Computers & Humanities* 23.3:199-214.

(1992) Modelling music with grammars: formal language representation in the Bol Processor. In *Computer Representations and Models in Music*. A. Marsden and A. Pople., Eds., Academic Press, London.

Laske, Otto (1993). Mindware and Software: Can They Meet?: Observations on AI and the Arts. *IAKTA/LIST International Workshop on Knowledge Technology in the Arts,* September 16, Osaka, Japan.

Rao, Suvarnalata (1994). Musicological perspective on synthesis of North Indian music. *International Conference on Pattern Recognition*, ISI, Calcutta, forthcoming.

Upadhye, Rajeev (1993). Future of Indian music in computer age. *Lensight*, July, pp.18-21 and 56.

Vercoe, Barry (1994). *Csound software and documentation.* Available as freeware from ftp site *cecelia.media.mit.edu*.

Yourdon, Ed (1994). *Guerrilla Programmer*, 1, 2, February.