THE NTA PROJECT

Giovanni Grosskopf
Agon Studio, Milano, Italy
Didier Guigue
Departamento de Música, Universidade Federal da Paraíba, João Pessoa, Brazil
ggrossk@tin.it , dguigue@openline.com.br

Summary: The GMT [1] is developing analytic tools for 20th century acoustic music. The theoretical background of these tools is the Object Oriented Analysis (OOA) methodology, which is based on the concept of sonic object. After shortly explaining this concept and methodology, we outline the GMT software project as a whole, then introduce the NTA software, and describe its tools with the help of musical examples.

Keywords: *Computer-assisted statistical analysis; sonic object analysis; 20th century music analysis; Max.*

SONIC OBJECT ANALYSIS

In the domain of acoustic music, a *sonic object* may be defined as the combination and interaction of the musical primary components (pitches) with the so-called secundary or statistical components, such as intensities, densities, and, generally speaking, space (achronic) and time (diachronic) filling up. It is a medium-level structure, between the lower level (pitch classes), and the upper level (macro-structure) [see fig. 1]. The way these objects are linked at this medium level happens to be a very important vector of form in 20th century music, as *timbre* becomes as structurally functional as pitch-classes were in older music or in some serial music.

In the background of the GMT [1] software project is the *Object Oriented Analysis* (OOA) methodology, which is thoroughly described in (Guigue 1996a, 1997a, 1997b). Very shortly and roughly speaking, the method consists in

1) segmenting the whole musical piece in a sequence of sonic objects [2];
2) describing the structure of each object according to a selection of relevant statistical components, and
3) quantifying the gap of sonic continuity between consecutive objects as a whole, or for

[1] GMT (*Grupo de Pesquisas em Música, Musicologia e Tecnologia Aplicada*) is a research Group coordinated by Didier Guigue at the Music Department of Universidade Federal da Paraiba (Brazil), with a grant from CNPQ (Brazilian Council for Research). Its members are researchers, composers, graduate and post-graduate students in music and computers, from various Brazilian universities. Giovanni Grosskopf (Agon Studio, Milano, Italy) is an associated researcher.

[2] The method admits polyphony of objects — i.e. multi-layered sequences.

each component, or for homogeneous groups of components [3]. This quantification configures a crucial aspect of the piece's formal *kynesis* and allows the form to be inferred from the succession of more or less contrasted sonic objects.

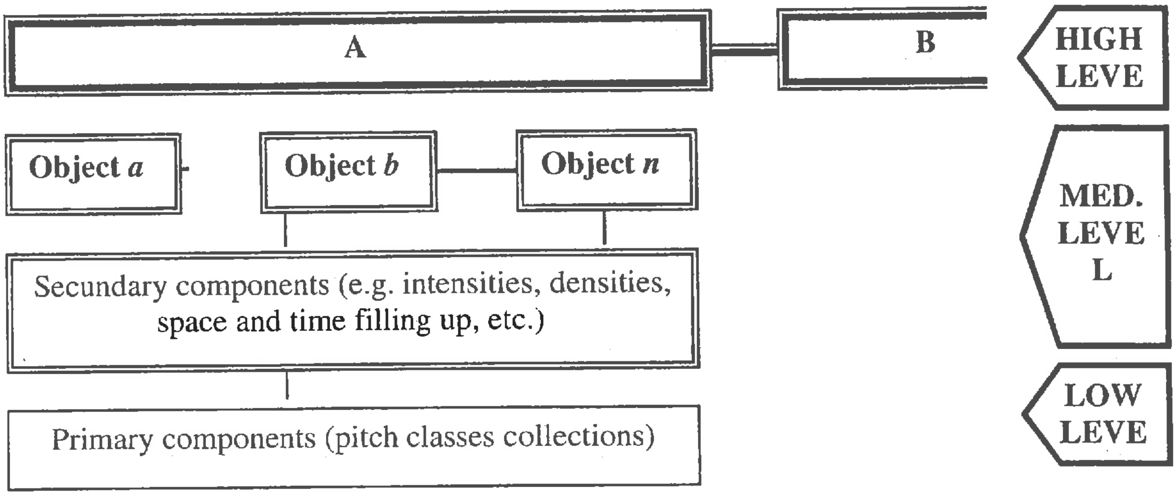


Fig. 1. The three levels of a musical structure, according to OOA model. The sonic object, at the medium level, is a combination and interaction of primary and secundary components. The high level is formed by the sequence of macro-sections A, B, ... N.

The basic rule for evaluating how important is a given component in the design of the sonic structure of a given object, is the rating of its relative *complexity*. The maximum "complexity" corresponds to the configuration that would produce a sonority as "complex" as possible. The meaning of the term "complexity" varies according to the nature of the component. When dealing, for instance, with the number of simultaneous notes (density) in a given object, the maximum value is that corresponding to a fully saturated object (e.g. a *cluster*). On the other side, a hypothetical empty object would receive a density weight of zero. It must also be observed that the concept of saturation (in the sample case of the density) will vary from one object to the other, according to its own boundaries (its lowest and highest pitches). Therefore, as a general rule, *all* quantifications must be relative (e.g. from 0 to 1), for it is the only way to compare and process together heterogeneous components, to compare objects, and, finally, to infer technical, aesthetic or stylistic conclusions from a piece or a set of pieces.

THE GMT SOFTWARE PROJECT

All these step and rules will be implemented in a future stand-alone application we are working on at GMT. The main concern for this software to be fully functional is to implement IA algorithms for segmenting a whole musical piece into a number of units — the sonic objects —, and infer (or suggest) the relevant components to be applied (Trajano *et al.* 2000). By the meantime this segmentation is manually done, applying the

[3] If relevant for the analytic purpose, components can be grouped in a hierarchical structure.

OOA rules to the written score of the music [4].

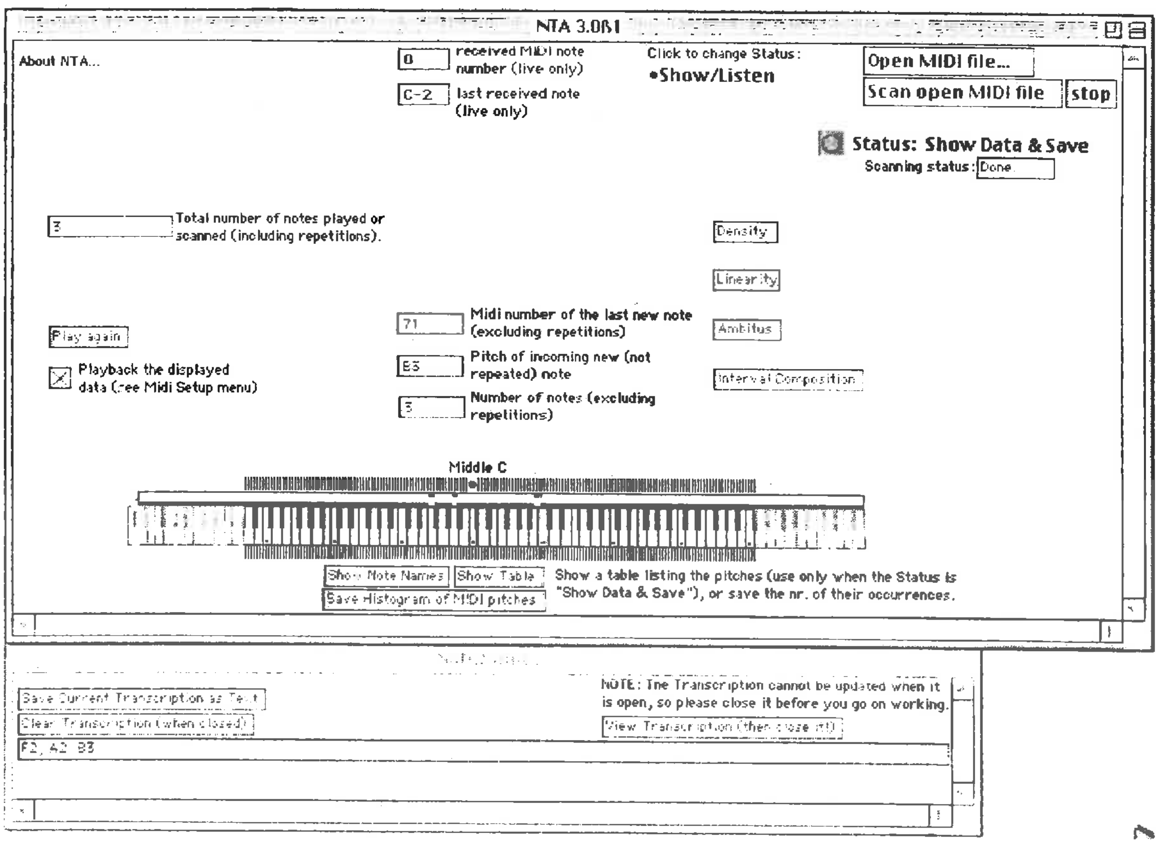
On another hand, we are developing simultaneously two sets of analytic tools:

- the NTA (*NonTonalAnalysis*) software, developed by Didier Guigue and Giovanni Grosskopf on Max; and
- SOAL, a library of tools developed by Didier Guigue and Ernesto Trajano, the first version of which is about to be released for the Patchwork/OpenMusic community [5].

As long as the automated segmentation is not concluded, each segment of the score representing a sonic object must be encoded as a single Midi file, in both software packages.

NONTONALANALYSIS

The present stage (version 3.0) [6] of the *Non-Tonal Analysis* (NTA) software project involves the development on Max (for the Macintosh) of a software capable to integrate as many as possible of the functions of our previous works [Grosskopf 1998, 1999] in a unique environment, providing also a friendly user interface.



---

[4] The focus on the written aspect of a music which is intended to be analyzed in its sonic properties is surely a point to discuss, but this is not the scope of this paper.

[5] A preliminary set of Patchwork abstractions is available at the software section of the GMT site: http://www.liaa.ch.ufpb.br/~gmt . Some of them was described in [Guigue 1996b].

[6] The version 2.0 of NTA is available at http://space.tin.it/musica/ggrossko .

Fig.2. The main window of NTA with the basic elements of its user interface.

NTA acquires the data by opening and scanning a standard midi file (representing the sonic object to be analyzed), or by playing the desired notes on a connected midi instrument. The interface looks as in fig.2.

The algorithms already implemented can perform the following tasks:

- count notes and show which notes are present in a played music object or in a midi file, which have been repeated or not repeated
- show how many occurences of each note are there, in the form of a Histogram
- evaluate the Density of the object
- evaluate the Linearity of the object
- evaluate the Ambitus of the object
- evaluate its Interval composition

Apart from the first task, which has a quite generic use, all the other parameters are analyzed and shown within their own subwindows, which are opened by clicking on the corresponding button on the right part of the main window. The last four parameters deal, exclusively, with the achronic structure of the sonic object. That means NTA considers any input object as a chord [7].

All what follows will be illustrated taking into account that the development of NTA is carried on using the programming environment Max (by Miller Puckette and David Zicarelli) for the Macintosh [Zicarelli *et al.* 1990]. Max, an application originally conceived for real-time midi and sound elaboration during live electronics concerts, can therefore be successfully employed also for data analysis and musicology, owing to its powerful capabilities of interacting with MIDI objects and with the user in a friendly and transparent way, and owing to the many ready-made objects for this purpose which are included in its package (like the *Histo* object, which provides the Histogram).

Density

Density is a measure of the ratio between the number of notes of the object and the maximum number of notes that could be comprised between its extreme pitches in a given musical system (here, the chromatic tempered scale). Thus, an object like a chromatic cluster will have a maximum density value. To implement this algorithm requires first to detect the number of notes of the object, then to detect the extreme (lowest and highest) pitches in the object, in order to complete the calculation, a task to be implemented also in many other algorithms.

Linearity

Linearity (fig.3) measures how far is the object from being composed all of equal intervals. In this case it would be a linear object (like a whole-tones hexatonic scale or an

---

7 Time-related tools are implemented in the SOAP libray.

augmented triad). The possible degrees and variations of linearity depend not only on the extension of the object (the distance between its lowest and its highest note), but also on the number of notes that are contained in its extension. This is a matter of degrees of freedom, and leads to use two different measures of linearity: an absolute one and a relative one.

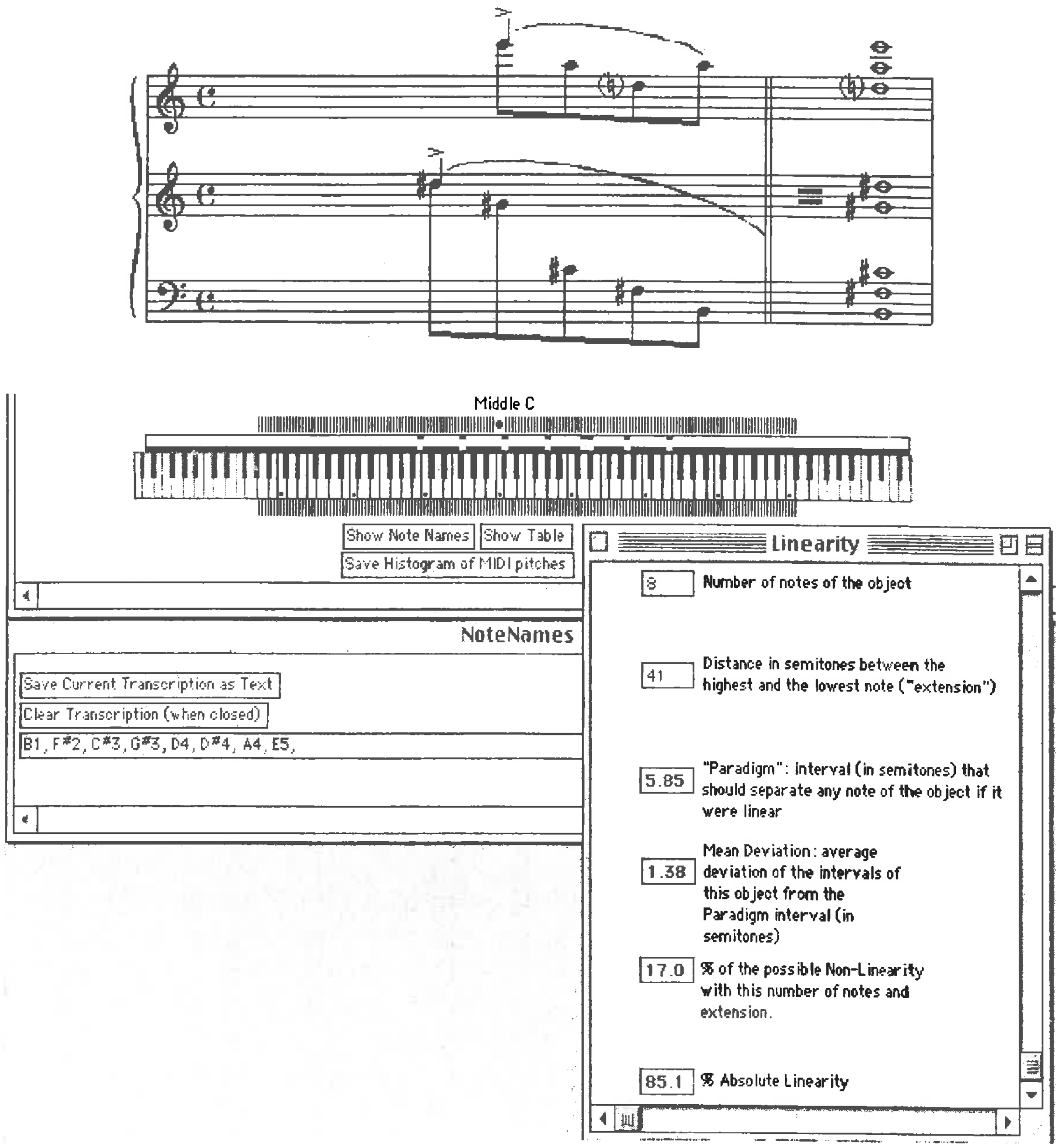


Fig.3. An example of analysis in the Linearity subwindow.
(György Ligeti, Études pour piano, premier livre, Étude 2, bar 5)

To perform Linearity, first NTA calculates the "Paradigm", the ideal interval (in semitones) that should separate any note of the object if it were linear. It is obtained

simply by dividing the extension of the object (the distance between its highest and lowest notes) by the number of consecutive intervals in the object.
The "relative" Linearity measure is then defined as the ratio between the Standard Deviation of the intervals of this object from the Paradigm interval (in semitones), and the Standard Deviation, from the Paradigm, of the intervals of an object with the same number of notes, corresponding to the case of the highest possible non-linearity (all the intervals but one = 1 semitone, and the remaining one = the largest possible interval).
The "absolute" Linearity measure is the same, but takes into account the number of vacant places (the chromatic steps not occupied by notes) actually present in the object, compared to the number of vacant places that would be the maximum possible if this object had only 3-notes [8].

Ambitus

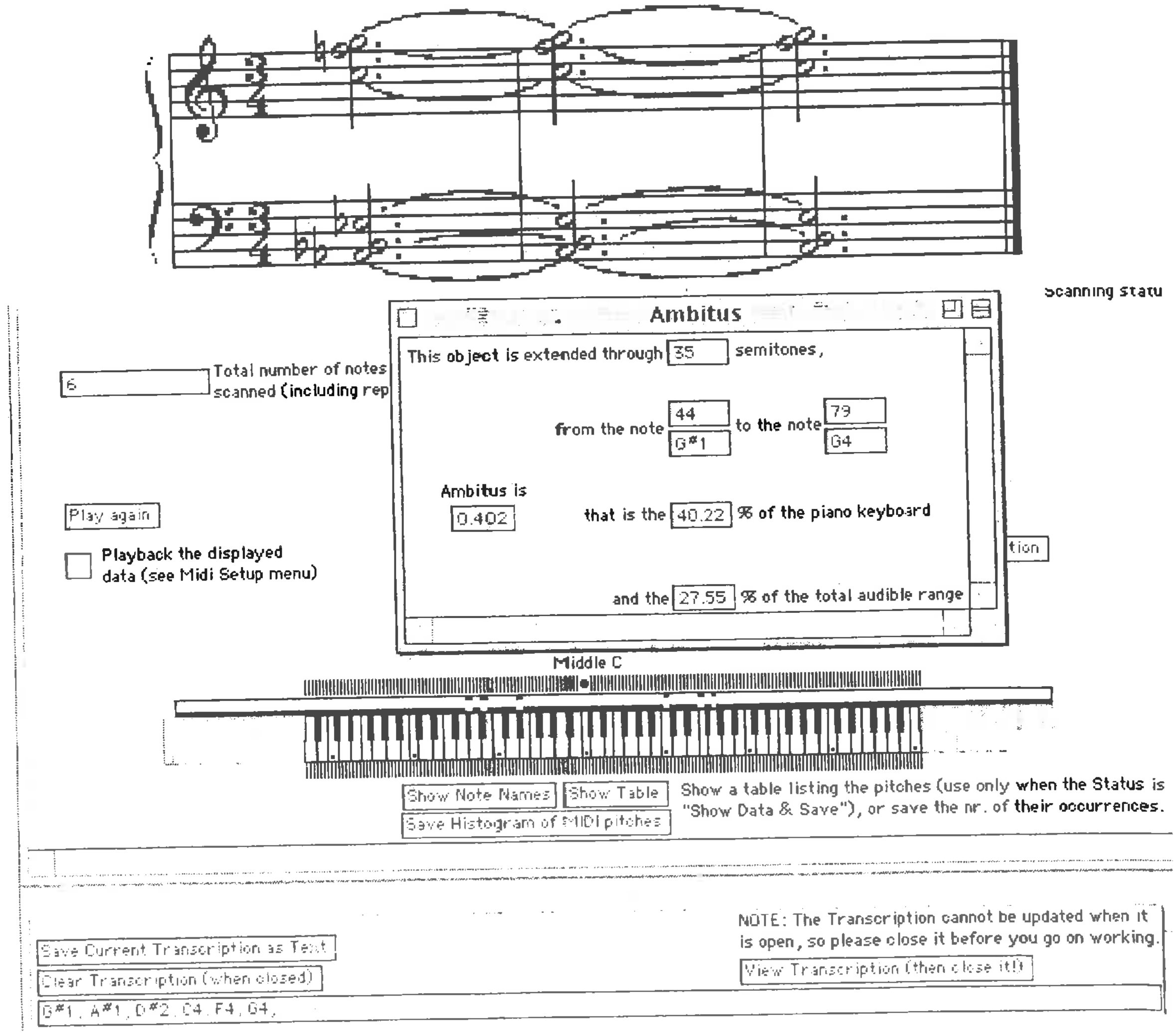


Fig.4 An analysis example in the Ambitus subwindow.

[8] In the case of triads, the two measures, absolute and relative, are one the complement of the other (so that their sum gives 100).

(Niccolò Castiglioni, Come io passo l'estate, piano suite, end of the first movement)

Ambitus (fig.4) evaluates the extension of the object (the distance between its lowest and its highest pitch), and compares the result with some common standards of comparison, like the extension of a piano keyboard (88 notes, that is 87 semitones) or the average audible range (the latter has been approximated to the total range of the MIDI notes, 127 semitones). Other comparison standards could be added.

Interval Composition

The goals of the Interval Composition algorithm (fig.5) are:

- to list all the existing intervals formed by any couple of pitches in the object
- to find how many occurrences of each existing interval are in the object
- to list which pitches form them
- to list their position within the object (e.g. if they are the lowest note, or the second one, or another inner note, or the highest one).

NTA is capable to measure the intervals in three ways:

- Without any octave reduction, having the semitone as the measure unit.
- With a reduction within the compass of one octave, using the usual standard names for the intervals.
- With a reduction within the compass of one major ninth, using the usual standard names for the intervals. This method has been implemented to support the Interval Perception algorithms (see below).

Besides detecting all the intervals, NTA will detect also some combinations of three notes, the sonority of which is not merely the result of the sum of the sonorities of the intervals composing them, but has a recognizable timbre in itself, and can have a large influence on the sonority of the whole object which contains them. To detect these three-notes combinations will be particularly useful for the Interval Perception algorithms (see later). The combinations to be detected are the following:

- major triads, in any position or inversion
- minor triads, in any position or inversion
- augmented triads (like C, E, G sharp), also in the open disposition (E, C, G sharp)
- the so called "conflicts of thirds", that are all the situations occurring when a major third (or minor sixth) and a minor third (or major sixth) are both present and share a common note, but do not form a triad, rather suggesting, from the point of view of traditional tonality, a minor and a major triad with the same root tone. For instance, C, E flat, E natural (that could suggest a conflict between C major and C minor) or C, E flat, C flat (that could suggest a conflict between Ab major and Ab minor: here the root tone itself is not included in the note combination).
- in some cases, the combinations traditionally analyzed as uncomplete chords of dominant seventh (like F, G, B).

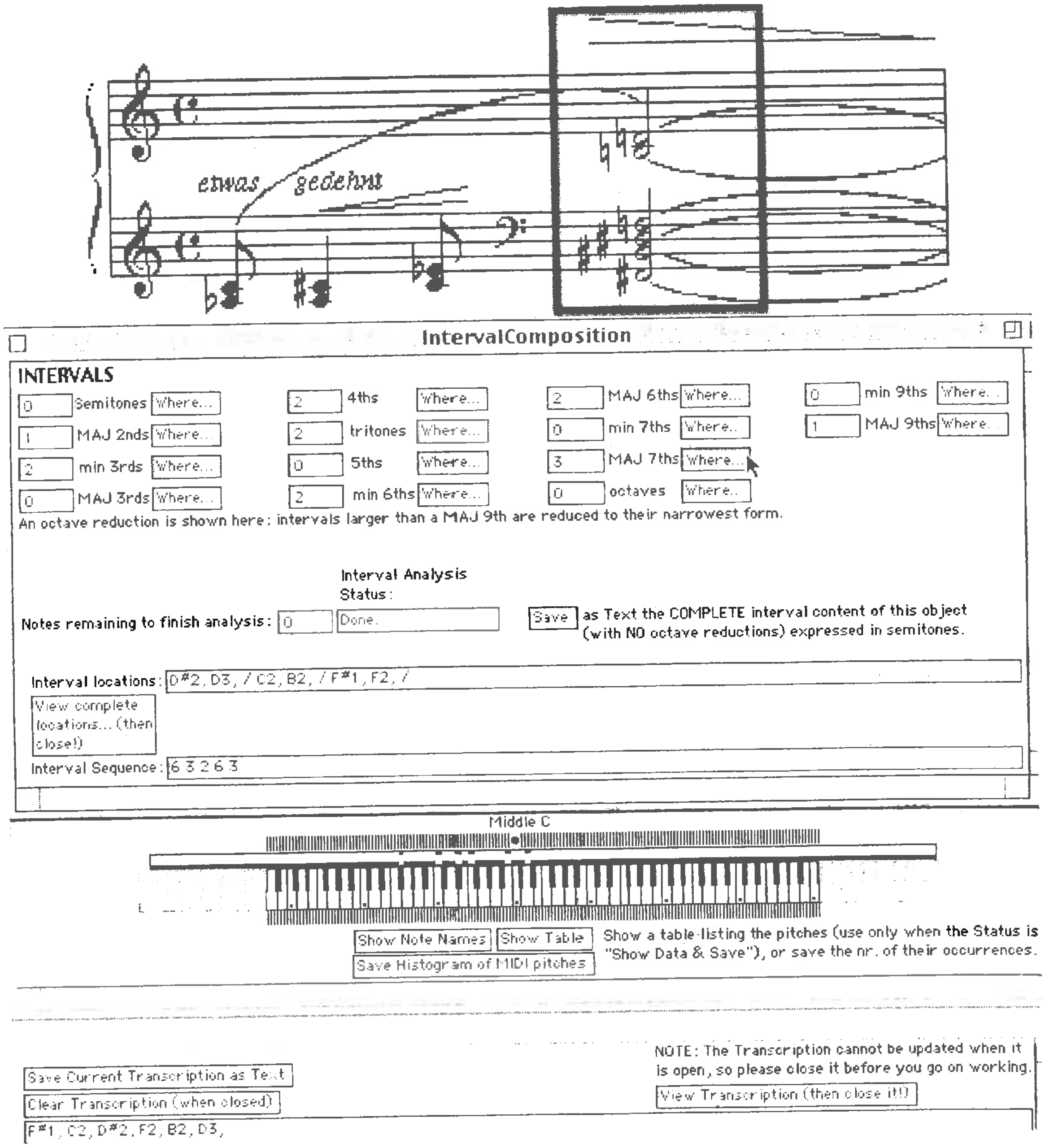


Fig.5. An analysis example in the Interval Composition subwindow.
(Arnold Schoenberg, op.19 n.2, bar 6)

Some future developments

In the near future some Interval Perception algorithms will be added to NTA, which will allow to detect which ones of the intervals composing an object are perceived more clearly and have the greatest influence on its overall sonority, depending on their nature (strong consonances and strong dissonances are perceived better by the human ear than the other intervals) and on their position in the object (for instance, the lower notes of a chord have usually a greater influence on its sonority than the upper ones). This will allow to classify the objects (mainly non-traditional, non-triadic, non-tertian chords), according to their sonority and to the "composition" of their sonority, providing a sort of "interval spectrum" of a harmonic object, which will list all the intervals contained in the object in hierarchical order, according to their importance in contributing to the global sonority of the object itself. This will allow to detect the similarities or dissimilarities in the sonority of different objects, and would be particularly useful in the field of computer-assisted composition.

Another important development will be the implementation of several algorithms (according to different approaches) to measure the dissonance level of an object. We have listed many methods for the measure of dissonance , and it's our goal to implement the largest possible number of them in NTA, in order to provide a comparison among them when they are applied to the analysis of the same object.

A MAX feature requiring attention

A major and not obvious problem with all the Max algorithms is that all the calculation must be triggered by a timed impulse (called the *bang*), that is, the order to perform the calculation must be given at a certain precise moment, and, owing to the particular configuration of Max, this moment must often be controlled by time-related objects (delays, timers, which count milliseconds, and counters, which count the occurrences of an event), and not by data-checking objects (objects which could check if certain data have already been produced or not). However, the exact instant in which Max *outputs* the data from the algorithm which *precedes* the one we are working on is sometimes quite unpredictable (for instance: if we open a very large midifile and transmit its contents from one module of Max to another, sometimes we cannot know exactly how many milliseconds will be necessary for all the data to reach the new module). The problem is therefore that, at the moment in which we would like to start the calculation of the new algorithm, we are not always sure that all the necessary data are available for the calculation to be performed, because they may not have been all produced or detected. This requires a particular care, and the opposite risks are either to slow down an algorithm in order to be sure that all it has all the necessary time to accomplish all its tasks and retrieve all the necessary data, or to have incorrect results with large data structures (like large midifiles), because a calculation starts when not all the necessary data are available yet. Besides, some events produce by themselves a *bang* and start the calculations anyway. The risk of triggering a calculation too early or too late must therefore be considered attentively.

REFERENCES

[Grosskopf 1998]: Grosskopf, Giovanni. "Un approccio diverso all'uso dell'armonia atonale ed all'analisi di accordi non tonali tramite il software "NonTonalAnalysis"", Proceedings of the Italian National Conference of Electronic Music "La Terra Fertile" 1998, Istituto Gramma, L'Aquila, 1998.

[Grosskopf 1999]: Grosskopf, Giovanni. "NonTonalAnalysis: a different approach to the analysis of atonal chords: achieving a clearly perceivable directional logic in atonal harmony.", in Proceedings of the International Conference "The Principles of Musical Compositions" held in 1999, Lithuanian Academy of Music - Lithuanian Composers Union, Vilnius, 2001. http://www.liaa.ch.ufpb.br/~gmt/Grossk/GKpapers/LITPIC.htm

[Guigue 1996a]: Guigue, Didier. "Principes méthodologiques d'une analyse orientée objets de la musique du XXe siècle". Paris, Ircam, Médiathèque Online. http://mediatheque.ircam.fr/articles/textes/Guigue96a/.

[Guigue 1996b]: "Un environnement informatique pour une approche analytique orientée objets de la musique du XXe siècle", in ASSAYAG, M., CHEMILLIER, M., & ELOY, Ch. (eds), 1996 : Troisièmes Journées d'Informatique Musicale JIM 96, Cahiers du GREYC, Vol. 4, pp. 266-272 www.ircam.fr/equipes/repmus/jim96/actes/guigue/guigue.html

[Guigue 1997a]: Guigue, Didier. Une Etude "pour les Sonorités Opposées" - Pour une analyse orientée objets de l'oeuvre pour piano de Debussy et de la musique du XXè siècle. Villeneuve d'Asq, Presses Universitaires du Septentrion, 559 p.

[Guigue 1997b]: Guigue, Didier. "Sonic object : a Model for Twentieth Century Music Analysis". The Journal of New Music Research, Vol. 26 n. 4, pp. 346-375. http://www.swets.nl/jnmr/vol26_4.html.

[Trajano *et al.* 2000]: Trajano, Ernesto. Guigue, Didier. Ferneda, Edilson. ."Segmentação automática de fluxos musicais: uma abordagem via agentes racionais", Revista Eletrônica de Musicologia, Vol. V.2, 12/2000. http://www.cce.ufpr.br/~rem/REMv5.2/vol5.2/guiguetrajano/guiguetrajano.htm.

[Zicarelli *et al.* 1990]: Dobrian, Christopher and Zicarelli, David. "MAX, An Interactive Graphic Programming Language", Menlo Park, California, Opcode Systems – IRCAM, 1990.