ETNA Builder - Interactively Building Advanced Graphical Tree Representations of Music

Wolfgang Chico-Töpfer

SAS Institute GmbH
In der Neckarhelle 162
D-69118 Heidelberg

e-mail: woccnews@web.de

Etna Builder - Interactively Building Advanced Graphical Tree Representations of Music

Wolfgang Chico-Töpfer

e-mail: woccnews@web.de

Abstract Etna Builder is a flexible visualization tool to build advanced graphical tree representations that capture the underlying transformational processes in music composition. It can be utilized in computer-assisted music analysis and synthesis but also in musicology as a standardized visualization means for transformational processes. It offers a modern GUI to build graphical representations in a straightforward and interactive way. The representation is saved in a human-readable format thus allowing full interchangeability e.g. with Salieri. Reductions can be concatenated and exported to Midi and Guido. The graphical views can be exported to the Gif format, thus making these visualizations easily embeddable in various documents.

1 What is Etna?

Etna[1] is a representation I originally designed to be able to describe the transformational processes behind musical composition more adequately. I was able to first apply it within a semi-automated Salieri[2]-based composition system called *AVA*[3]. Precisely speaking, I applied it in the human-based analysis phase to obtain a deep structure of the musical themes that served as input material. Then *AVA* took over and transformed those deep structures into foreground elaborations.

Etna has a number of advantages that overcome drawbacks of former representations. It abstracts from specific transformational music theories such as the GTTM [Lerdahl; 1983], while being fixed on transformational aspects and not on the musical ones that may differ from theory to theory. Its definition is formally given [Chico-Töpfer; 2001], thus yielding precision, clarity and unambiguity. That definition also includes a linear notation which is equivalent to the graphical representation. As a consequence, Etna makes algorithmic processing a straightforward task, a feature that is exploited in the Etna Builder. Another advantage is that, in spite of superposing with other representations that use tie notation (as in CMN[4]), representation clarity is still kept.

Figure 1 shows a graphical Etna description of the melody (Fig. 2) of the B part of one of Mozart´s piano variation themes[5]. It uses simplified so-called SALIERI-based *complex segments* [Chico-Töpfer; 1998] since they are straight-forward and clear enough to allow their application without much explanation. For instance, $\{a2/2.\ c\sharp\ e\}$ stands for the A major chord played in the 2nd register with a relative duration of

[1] Elaboration-orientated Tree Notation for *AVA*

[2] an interactive computer music system, see [Hoos et al.; 1998a]

[3] Aleatoric Variation Algorithm

[4] Conventional Music Notation

[5] Mozart´s piano variations actually formed the basis for *AVA*´s first musical data base, see [Chico-Töpfer; 1998] and http://www.informatik.tu-darmstadt.de/afs/salieri/ava/ava.html for the results. Note that the example K'265 has been simplified and is shown without ornamentation.

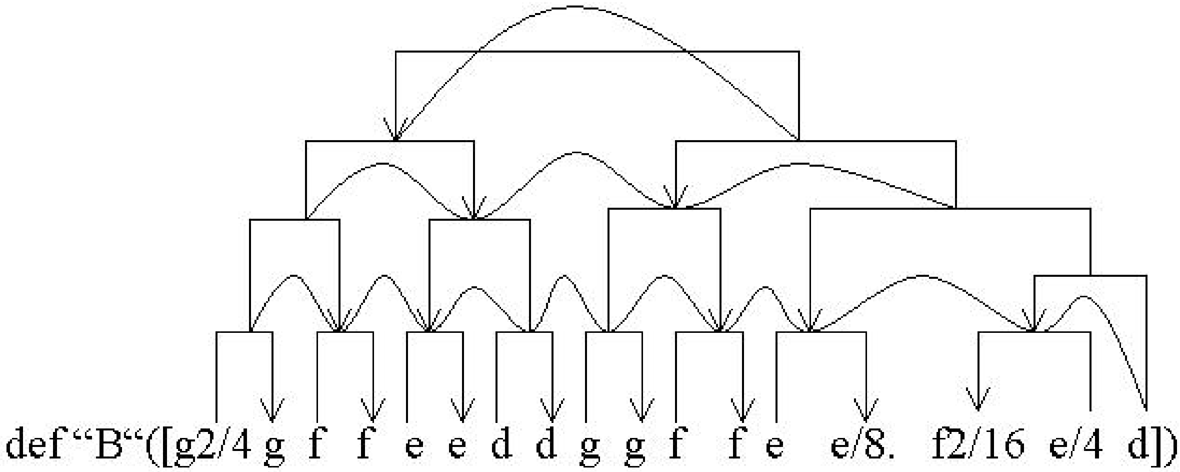

Figure 1: A graphical ETNA description of K.265´s melody (part B) including superposed groundlines

3/4 i.e. 1/2 + 1/4 because of the dot. For simplicity, we leave out indications of intensity and instrumentation. Alternatively, we may as well build the tree upon a CMN representation. Figure 1 also shows so-called superposed groundlines which

Figure 2: K.265´s melody (part B)

is a superposition that displays which reductions may be used to elaborate a new composition e.g. by *AVA*. Each groundline is graphically shown as a chain of ties - or, simply speaking, a wave- that passes along the reductions which it contains. For example, the wave at the top of the tree consists of a ground line $[g2*2 \;\; d*2]$ i.e. a $g2$ along 2 measures followed by a $d2$ of equal duration. In other words, groundlines are basically sequences composed of reductions. The superposition of groundlines shows how visually clear a superposition can be as a consequence of carefully designing a representation to accomodate a superposition.

The corresponding *linear* ETNA representation of this example can be seen in Figure 3. Note that the linear notation is an equivalent representation [Chico-Töpfer; 2001].

$$(((g2/4 \rightarrow g) \rightarrow (f \rightarrow f)) \rightarrow ((e \rightarrow e) \leftarrow (d \rightarrow d))) \leftarrow$$
$$(((g \rightarrow g) \rightarrow (f \rightarrow f)) \leftarrow ((e \rightarrow e/8.) \leftarrow ((f/16 \leftarrow e/4) \leftarrow d)))$$

Figure 3: A linear ETNA representation of K.265´s melody (part B)

2 What is the ETNA Builder for?

The ETNA Builder software provides the ETNA user with a lot of support for drawing and mapping ETNA representations. S/he is assisted with various design styles and a so-called customizer (see Fig. 4) to interactively build an ETNA representation.

Moreover, support is given to export, organize and superpose an ETNA representation.

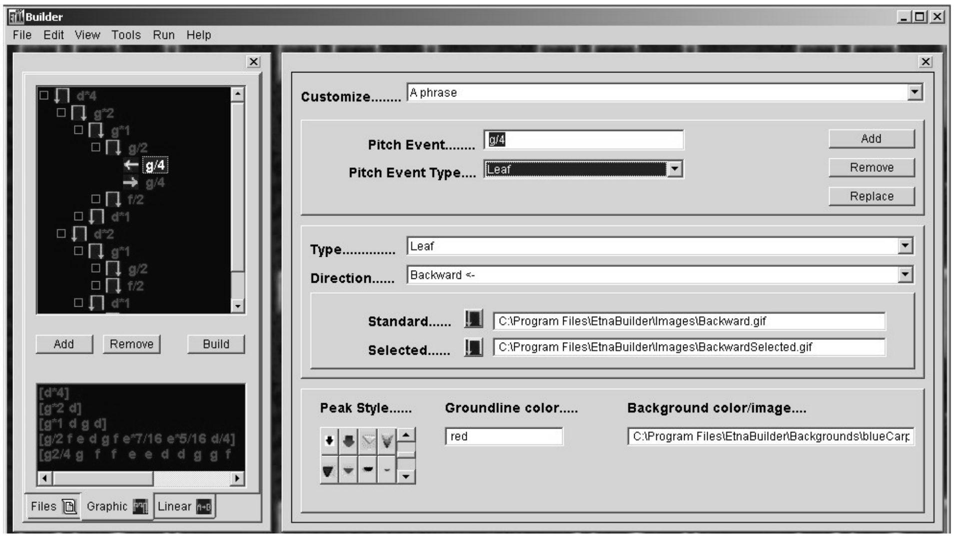

Figure 4: The ETNA Builder's customizer to build and modify an ETNA representation

Figure 5 shows the ETNA Builder while visualizing the graphical representation of the example K'265. Note that saving a graphical ETNA representation as EPS file lets the ETNA Builder default to saving it as a black and white graphic (as applied here), while saving it as GIF yields the original graphic with all the applied colours. *AVA*'s own theme input format is also supported so that we can feed *AVA* with ETNA Builder output. Various views are implemented:

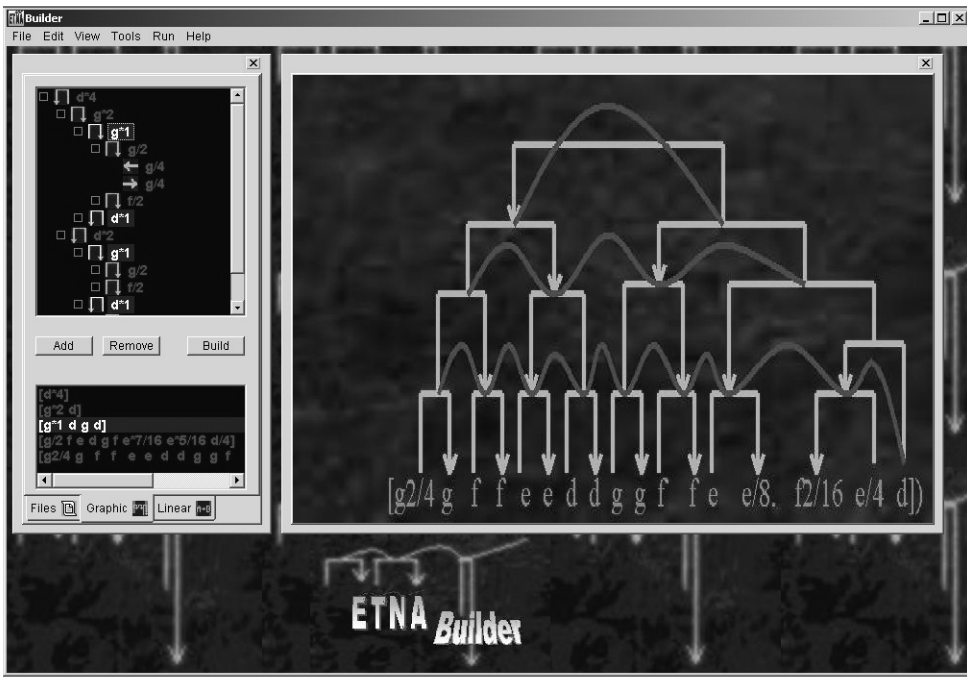

Figure 5: The ETNA Builder while being applied for the theme of K'265

1. **the graphical view:** This is the main display of ETNA representations (Fig. 5). It also allows to build or modify an ETNA representation through its customizer (Fig. 4) which is accessible by simply hitting the *Build* button.

2. **the notational view:** The linear representation is shown according to the selected display format. This can either be the space-saving *compact* format or the *pretty* one. If the latter is selected, the representation is blown up by indentation and line feeds according to its bracket structure. This allows a more human-friendly view of the ASCII representation (Figure 6).The user may build her/his representation here as if working with a text editor.

3. **the file view:** Etna Builder is a project-orientated application and each Etna representation is associated with files of various types e.g. GIFs, GMN[6]s, and MIDs. So one can select a certain view by simply clicking on the rsp. file listed in the explorer.

These views are further refined: For instance, the notational view also supports CMN as can be seen in Figure 7. Note that only those groundlines are visualized that are selected on the explorer pane on the lower left hand side where the groundlines are listed as complex segments (for consistency reasons, the musical surface and the deepest reduction are listed as groundlines by default although they are not displayed as such in the graphical view). Figure 7 shows the case when all groundlines are selected. To hear a groundline, a double click on the rsp. groundline suffices to generate and play a corresponding Midi file. Multiple groundlines can be heard together by double-clicking on the CMN representation (Fig. 7).

The graphical representation is further refined by yet another visualization that consists of an explorer-like tree view that allows to expand and collapse parts of the tree (see Fig.5 on the upper left-hand side). Note that in this view the icons are needed to denote the direction of the transformation and thus keep a time-dependent order. Groundlines can be built by simply picking up the rsp. reductions and clicking on *Add*. A click on a groundline on the left suffices to see which reductions make this particular groundline e.g. see Fig.5 for $[g*1 \quad d \quad g \quad d]$. This explorer-like tree view is also used

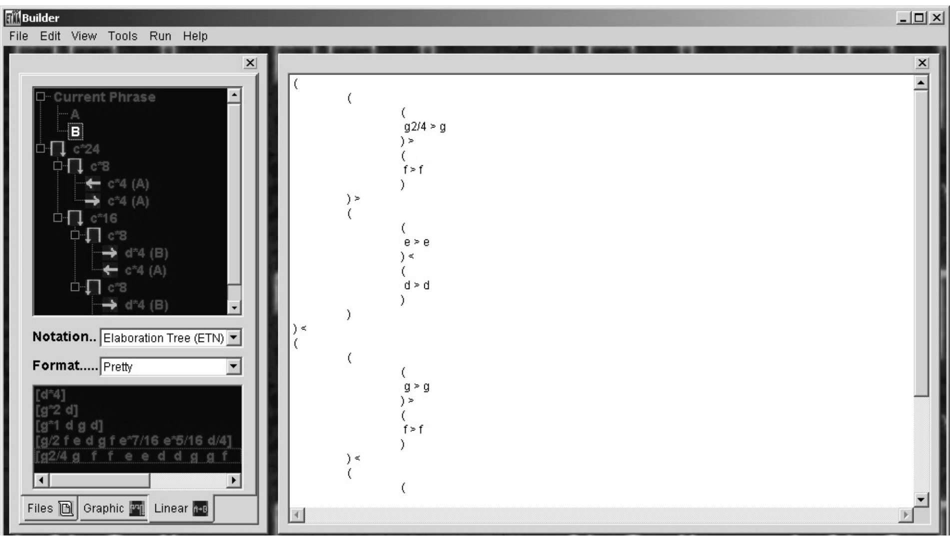

Figure 6: The Etna Builder's notational view displays linear Etna representations

to visualize the other transformational processes on the larger scale of structure (see Fig. 6 and 7 on the upper left-hand side). We see an explorer-like representation

[6]Guido Music Notation, see [Hoos et al.; 1998b]

of the phrase structure that lets one see the overall structure at a glance. This also allows building large ETNA representations comfortably by building them part by part e.g. first the main tree and then the leafs. The latter are sub trees, so-called *Builder phrases* that are themselves graphical ETNA representations. This way one can switch between various trees that make up the rsp. piece. For instance, in K'265 one can switch between A and B (note the selected B on the upper left-hand side of Fig. 6 in *Current Phrase*).

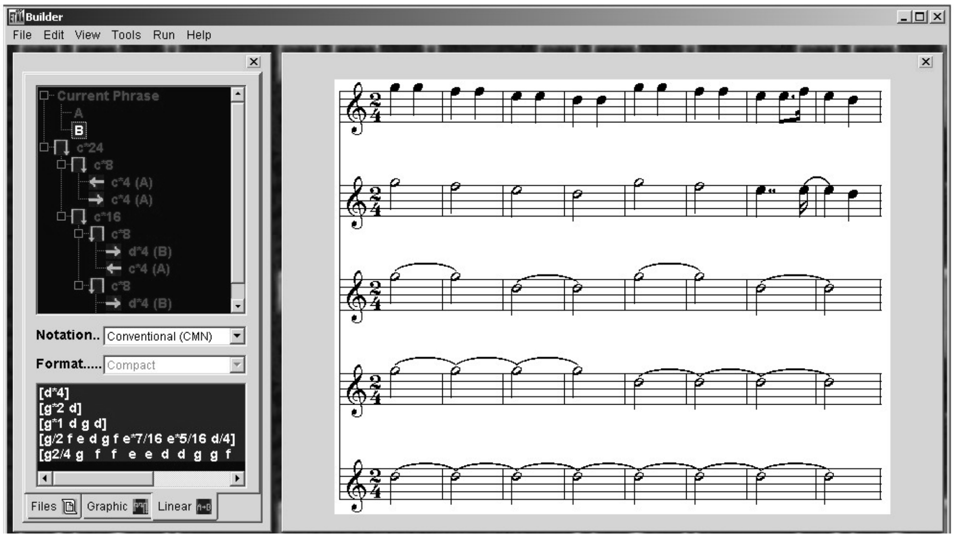

Figure 7: The ETNA Builder's notational view also displays CMN to visualize groundlines

3 Background and Related Work

ETNA has served to define the basis for the semi-automated composition environment *AVA* [Chico-Töpfer; 1998] and has thus already been applied as a musical representation system. In terms of structural generality and expressive completeness as defined in [Wiggins; 1993], ETNA is able to represent anything that can be represented in CMN while being structurally focussed on hierarchical information representation.

It is interesting to see how ETNA relates to other musical representation systems. For instance, the PIMS approach [Assayag; 1993] seems at first sight structurally comparable with ETNA. In fact, it turns out that groundlines can be generally understood as partially instantiated musical structures because they reflect a generic structure made of substitutes for sub structures that are to be elaborated. For instance, in Fig.2 the lowest groundline depicts a partial instantiation as only the last element is fully instantiated whereas the rest is made of substitutes to be further elaborated. Moreover, elaborating a groundline means to automatically follow the relations and constraints defined in the ETNA representation by obeying the definition of an ETNA reduction [Chico-Töpfer; 2001].

Comparing Elody's [Letz; 1998] theoretical foundation with ETNA shows some interesting aspects. Its basis are musical objects such as the constructors *Seq* and *Mix* which consume objects such as notes and rests to generate more complex structures. Typically, ETNA uses such constructors to represent the resulting musical material, in other words the foreground elaboration. It would be interesting to see how much effort is involved in expressing Etna within the confines of Elody. It seems that Etna reductions could be expressed as Elody abstractions whereas elaborations, on the other hand, might be generated through Elody applications (well-known from lambda-calculus) to instantiate musical structures. This process would work by applying several layers such as a rhythmic layer that would account for the actual reductions with regard to their duration.

David Cope's application [Cope; 1991] of Augmented Transitional Networks [Woods; 1970] may provide a basis for further extensions of *AVA* [Chico-Töpfer; 1998]. However, this representation is much more orientated towards a mechanism of music generation than to the representation of musical structure. On the other hand, David Cope has also developed parse diagrams[Cope; 1991,] which along with [Lerdahl; 1983] may count as the closest relatives of Etna. Both, however, have in common certain drawbacks that Etna has overcome by means of precision, unambiguity and formal foundation [Chico-Töpfer; 2001].

Symbolic Composer [Morgan; 1992], OpenMusic [Assayag; 1997] and Common Music [Schottstaedt; 1997] resemble each other in that they all apply an existing universal programming language's (namely LISP) data structures to represent musical structures. Etna may well be represented in one of these environments. The main difference is that Etna has not been devised with a certain environment in mind and is thus not really comparable.

Comparing Etna with less computer-orientated musical representations such as those used in the GTTM [Lerdahl; 1983] makes much more sense. For example, consider the trees used to represent the time-span reductions: By losing concrete information such as duration values one can transform an Etna tree into a time span tree provided that it makes sense with respect to the GTTM's preference rules. Thus, to make sure it makes sense, it is -in GTTM terms- a question of combining Etna with a certain weight matrix as a selector for GTTM preference rules. The other way round i.e. transforming a GTTM tree into an Etna representation poses the question if prolongation trees can be expressed with the current definition of possible reductions, which will be further investigated in forthcoming work. Note, however, that there are a number of possibilities to build GTTM trees that violate properties that are required in the GTTM itself [Lerdahl; 1983, p.114, Fig. 5.7a-c]. As Etna representation, such a tree is by definition impossible to describe.

4 Conclusion

Etna is a powerful tree representation to describe transformational processes in composition. It is proposed for general use within music analysis and synthesis. This is heavily supported by the Etna Builder which enables one to easily draw and manage Etna representations. Etna Builder is planned to be released as free software in the 3rd quarter of 2001.

References

[Assayag; 1993] The Music Representation Project at IRCAM; ICMC 1993 Proceedings, pp.206-209.

[Assayag; 1997] An Object Oriented Visual Environment for Musical Composition; ICMC 1997 Proceedings, pp.364-367.

[Chico-Töpfer; 1998] W. Chico-Töpfer. *AVA*: An Experimental, Grammar/Case-based Composition System to Variate Music Automatically Through the Generation of Scheme Series; ICMC 1998 Proceedings, pp.98-101.

[Chico-Töpfer; 2001] W. Chico-Töpfer. ETNA - a New Advanced Graphical Tree Representation of Music; SBCM 2001 Proceedings. VIII Brazilian Symposium on Computer Music.

[Chomksy; 1957] N.Chomsky. Syntactic Structures. The Hague: Mouton, 1957.

[Chomsky; 1965] N.Chomsky. Aspects of the Theory of Syntax. MIT Press. Cambridge, Massachusetts 1965.

[Cope; 1991] D.Cope. Computers and musical style. The Computer Music and Digital Audio Series, Volume 6.A-R Editions. Madison, Wisconsin 1991.

[Hoos et al.; 1998a] H. H. Hoos et al. SALIERI - a General, Interactive Computer Music System; ICMC 1998 Proceedings, pp.385-392.

[Hoos et al.; 1998b] H. H. Hoos et al. The GUIDO Notation Format - a Novel Approach for Adequately Representing Score-Level Music; ICMC 1998 Proceedings, pp.451-454.

[Lerdahl; 1983] F.Lerdahl, R.Jackendoff. A Generative Theory of Tonal Music. MIT Press Series on Cognitive Theory and Mental Representation. J.Bresnan, L.Gleitman, S.J.Keyser editors. MIT Press, 1983.

[Letz; 1998] S.Letz,D.Fober,Y.Orlarey. The Role of Lambda-Abstraction in Elody; ICMC 1998 Proceedings, pp.377-384.

[Morgan; 1992] N.Morgan. Symbolic Composer - a description and evaluation. IMPAC Consultants. Tonality Systems. Wakefield, 1992.

[Schottstaedt; 1997] B.Schottstaedt; Common Music Notation; http://ccrma-www.stanford.edu/CCRMA/Software/cmn/cmn.html.

[Wiggins; 1993] G.Wiggins, E.Miranda, A.Smaill, M.Harris. Surveying Musical Representation Systems: A Framework for Evaluation; Department of Artificial Intelligence, Department of Music. University of Edinburgh. Edinburgh, Scotland. Published in: Computer Music Journal Vol.17/3. pp. 31-42.

[Woods; 1970] W.A.Woods. Transition Network Grammars for Natural Language Analysis. Communications of the ACM, Vol.13 / Nr.10 / October, 1970 pp.591-606.