Center for Computer Research in Music and Acoustics January 1984

Department of Music
Report No. STAN-M-15

INTELLIGENT SYSTEMS FOR THE ANALYSIS OF DIGITIZED ACOUSTIC SIGNALS

Final Report

by

John Chowning, Loren Rush, Bernard Mont-Reynaud, Chris Chafe, W. Andrew Schloss, Julius Smith

Research sponsored by

National Science Foundation
and
System Development Foundation

CCRMA
DEPARTMENT OF MUSIC
Stanford University
Stanford, California 94305

Center for Computer Research in Music and Acoustics January 1984

Department of Music
Report No. STAN-M-15

INTELLIGENT SYSTEMS FOR THE ANALYSIS OF DIGITIZED ACOUSTIC SIGNALS

Final Report

by

John Chowning, Loren Rush, Bernard Mont-Reynaud, Chris Chafe,
W. Andrew Schloss, Julius Smith

Our research has centered on recognizing and abstracting features of digitized acoustic signals by combining advanced techniques in signal analysis with knowledge-based programming methods. The combination of low level feature detection (signal processing) with high level constraints and heuristic programming techniques has been quite successful, yielding results far superior to what had been achieved with previous approaches.

The application domain we chose for this work is the analysis of performed music, particularly of expressively played 18th century western classical pieces. This domain was deemed especially appropriate for exploring the integration of signal processing and high level constraints, due to the large amount of structure inherent in the signals.

Expressive musical performance typically involves major departures from the metric relationships implied in a score. A central task for our system has been to recover the intended metric relationships from the digitized sound of the performance. This turn allows an analysis of the ways in which the performance differs from the notation.

We have applied our research system to a graded series of musical examples which were digitized from live performance. Methods have been developed in the following areas: low-level feature extraction (notably, signal processing techniques for segmentation and pitch extraction); high-level feature extraction (such as the recognition of rhythmic and melodic patterns); and control and resource allocation in systems with multiple levels of description (top-down and bottom-up context building, multi-criteria decision making, strategies for propagating constraints and/or hypotheses across levels).

The level of competence currently achieved by our system allows single voice examples to be transformed into an internal representation, from which common musical notation and other descriptions can be easily derived. Melody lines from works by Bach, Beethoven, Chopin and Mozart have been performed, recorded and digitized, and then successfully analyzed. The system has also been applied in the same way to other musical styles, notably music by Joplin and Afro-Cuban percussion. This has been possible because the methods used are general in their attempt to capture musical structure, rather than tightly bound to a very specific style of music. In fact, many of the ideas developed in the context of our research are applicable to other types of signals also rich in context, such as speech signals.

The results achieved on this project provide an important basis for the expansion of this type of research. The understanding of the system architecture problems we have derived from our research lead to strong hypotheses about the kind of control structure needed to handle more complex signals. The analysis of polyphonic sound is the next major challenge. It requires a solution to the source segregation problem, which can only be based on the use of the great deal of context and feedback between levels. The techniques we have established so far provide building blocks for the exploration of this next level of difficulty. Successful approaches developed in this context will be of primary interest to researchers working in signal analysis in related domains.

This research was supported by the National Science Foundation under Contract NSF MCS 80-8012476 and System Development Foundation under Grant SDF #345. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Stanford University, any agency of the U. S. Government, or of sponsoring foundations.

1. Overall Project Description

This research combines modern techniques in signal analysis with knowledge-ba programming techniques toward the goal of automatic transcription of musical sou Questions of system architecture, signal processing methods, high level pattern recognit and context driving have been investigated by applying the research system to a gra series of musical examples that were digitized from live performance. This has resulted the development of new techniques for signal segmentation, for multiple criteria decis making, and for the automatic recognition of musical features, both local (such as rhyth and melodic patterns) or global (such as meter and key).

In the current implementation, the analysis system is divided into an acoustic analy subsystem and a musical analysis subsystem, which are largely independent (See Figure 1)

The acoustic analysis subsystem is a front-end primarily concerned with note segmentat and pitch detection. It constructs the first concise description of the digitized sound, as a of discrete events. Signal processing routines provide descriptors for each event, includ the time at which the event occurs, its estimated frequency and amplitude, and in some ca the identity of the source (instrument and/or gesture or stroke causing the sound).

The musical analysis subsystem takes as input the event list produced by the acou analysis front-end. The goal of this subsystem is to infer the musical structure of performed music, including key signature, meter, and individual note values. It relies musical knowledge, in the form of hypothesis and pattern generators, constraints, evaluat criteria, context gathering mechanisms and control strategies, represented by rules or procedures. This subsystem builds a detailed internal map, from which utility progr print musical notation, tempo charts and other data about the performance.

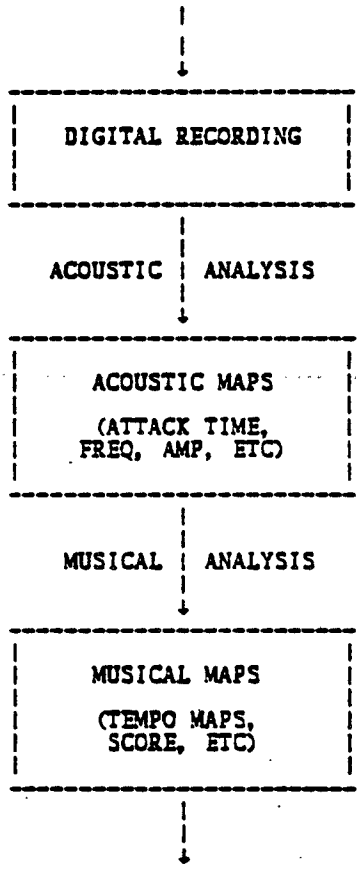

Figure 1: System Overview

The rest of this report is organized as follows. Subsections 1.1 and 1.2 are devo respectively to the acoustic analysis and the musical analysis subsystems. Subsection discusses the system from the point of view of control strategies.

Section 2 gives a much more detailed description of the musical analysis subsystem. particular, an example of analysis is followed step by step through the various stages processing.

Section 3 draws some conclusions and discusses future research. Further information can found in the publications included as appendices.

1.1 Acoustic Analysis: Signal Processing

The lowest level of processing in the system occurs in feature extraction. This is done signal processing algorithms, primarily used for note segmentation (determining onset tir and pitch extraction (estimating frequencies). The control structure driving t algorithms (acoustic analysis subsystem) constructs a note list, which will be the inpu the higher level processing steps.

A variety of signal processing techniques are used. While some of them are standard (suc FFT-based spectrum analysis), others were developed specifically for this effort. The behind the development of signal processing algorithms has been to provide reli detection of the features necessary for each type of music or instrument. The us particular algorithms is left to the control strategy of the higher level processing.

Signal processing methods were tested with a variety of musical instruments, inclu sound that was synthesized from the same musical scores as those used by the musicians played for the recordings. Certain problems of live performance presented difficulties in initial evaluation of new techniques. For example, substantial variations in pitch, timbr amplitude often occur from note to note, or within a note. Also, note onsets can be obsc by the ringing of the previous note and/or by room reverberation. The use of synthes musical data allowed us to provide better controlled inputs for initial testing, before applied the methods to live performance data.

In the current system we have had considerable success using variations of modern spect analysis algorithms. One example is the pitch-based segmentation procedure describe *Toward an Intelligent Editor of Digital Audio: Signal Processing Methods* (Foster et 1982). (See Appendix 2). In this algorithm the sound data is processed in reversed-time. processing the sound in reverse, a note can be identified from a cluster of partials by loc onto it somewhere in the sustained segment, where the note is strong and stable. The at that follows (in reverse) is usually easy to characterize, particularly as to the exact mon that the cluster of partials disintegrates into incoherence. This technique provided accuracy in the note segmentation of single-voiced musical examples. The success obtained were .98 for the piano, .86 for the flute, and .93 for the violin pizzicato.

Unfortunately, this algorithm was found inadequate for polyphonic inputs (multiple so sources). Attempts to generalize the method by tracking several clusters of par simultaneously have been rather unsuccessful so far. This is largely due to the tendenc multiple voices to sound in consonant harmonic relationships, which means that the par overlap to a large extent. The solution of this problem will depend on a stronger integra of the signal processing techniques with artificial intelligence techniques. This was possible in the current hardware configuration. In the single voice case, at least, a succes approach was developed.

Another useful algorithm for note segmentation was implemented using techni originally developed for linear predictive coding of speech signals. This algorithm is b on an autoregressive (AR) model fit of the signal, rather than a Fourier decomposition as used in the above method. Essentially, the algorithm detects events by observing the behavior of an AR model that is recursively fit to the data. By detecting when the mod no longer a good fit to the data (this typically happens at note boundaries), segmentation be performed. This algorithm yields meaningful results even in fairly complex polyph examples.

The algorithms described are but a few of those that have been implemented for derivation of the note list. The collection of algorithms provide a *tool box* for the hi levels of analysis to use in each particular situation. They were developed as needed for instrument studied and each processing problem that was identified in the higher leve analysis. Together they provide a sufficient detail of analysis of the acoustic signals to a processing by the higher levels as described in the following sections.

1.2 Musical Analysis: Construct Recognition

The input to this subsystem is normally the event list produced by the acoustic front- (Data gathered directly from a keyboard can also be used as input). In either case, primary goal is to turn the acoustic description of the performed music into mu notation.

One of the first problems which arise is to convert frequency estimates to pitches musical scale. With the exception of percussive music, for which we simply omit pitch analysis, our examples use the standard twelve semitone scale. Currently, the conversio pitches to scale degrees is simply done by rounding the appropriate function of frequenc the nearest semitone on the scale. While this approach lacks generality, and will fa complex situations, it has behaved adequately with the various examples we have consid so far. More sophisticated schemes have not been implemented for this reason.

The greater part of the effort in the musical analysis subsystem has been aimed at temp analysis. The problem is to turn numbers representing performed durations into me values. This is far from obvious, except in the case of *mechanical* performances of sim music. Previous attempts at automatic transcription have either been limited to those sim situations, or have relied on the user to provide additional input (tempo, beats or b and/or meter).

By contrast, our research is aimed at realistically complex examples. We have alre mentioned that the current system does not handle polyphony. However, within horizon of monophonic signals, our goal has been to handle a great variety of situations. particular, we have found it important to allow much natural expressiveness in performance of the pieces. This implies that simple approximation schemes cannot be tru to give good answers.

Since patterns and context come to play a major role in the search for plausible me notation, we are led to the use of AI techniques similar to those found in other percept systems, such as speech understanding systems. We use musical knowledge for recognition of musical structures (such as rhythmic and melodic patterns) in the sequenc performed pitches and durations. Pattern recognition, hypothesis generation, and cont building heuristics such as *island driving* are all important aspects of the approach.

In addition to the use of knowledge-based programming techniques, a key idea in temporal analysis is to separate global tempo fluctuation from local fluctuation in the tim of individual notes. The problem of assigning metric proportions throughout the piec first addressed at the global level (*structural anchors*) before the local level (individ notes). The concept of *tempo line* was introduced in order to capture the global te fluctuation. This was already a key concept in the CMJ paper (*Toward an Intelligent Edi*

of Digital Audio: Musical Construct Recognition [Chafe et al., 1982], see Appendix 1). Sir then, we have extensively augmented and revised the methods of hypothesis generation a evaluation, and the strategies used for arbitration and context driving. Yet, the tempo li concept has retained its central role. This is quite apparent in Figure 2, which summari the flow of information in the current system. Also note the use of multiple methods generating temporal clues. All this will be discussed in detail in Section 2.

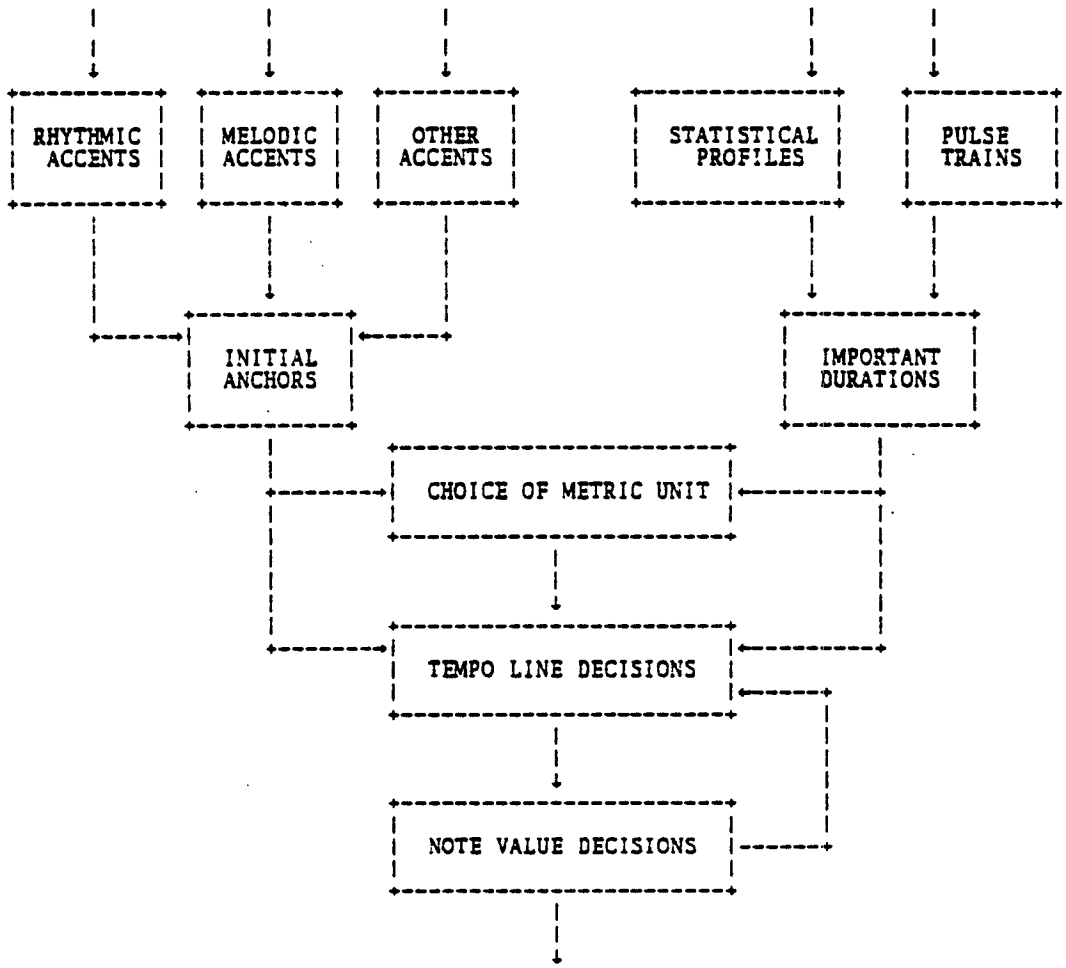

Figure 2: Simplified Data Flow of Musical Analysis

The improvements made since the CMJ paper have allowed the system to deal successfu with highly syncopated music, such as ragtimes or African percussion, while continuing perform well on the earlier examples (18th Century music) whose analysis relies mostly very different structural clues. In the case of the Afro-Cuban rhythms, the system provided standard notation (that experts determined to be correct) for music that is usua not transcribed by anyone, and that Western musicians typically find difficult to transcri

partly because the sounds and the musical patterns are less familiar to them, and p
because the event density is quite high.

The success of the method ultimately relies on the ability of the program to take adva
of complementary approaches which capture alternative clues to the musical structure
us illustrate this point at two successive levels of the analysis of durations. At the lo
level, *features* are extracted either from contrasts in duration, or from regularity
absence of contrast). The only situation in which the program fails to pick up any clu
this level is when successive durations stay in the *gray zone* between *almost equa*
clearly different. Fortunately, in most music, rhythmic interest cannot be maintained
long in that gray zone, so the music is not performed that way. The program is
guaranteed to almost always find significant clues at the lowest level of duration proces

The next level of analysis begins with a search for simple patterns in the melody, i
succession of structural accents, and in the consistency of note durations. All these si
patterns provide hints to the temporal structure, and the program uses them sing
combined in order to form tempo line hypotheses. Music whose accents do not follow si
timing patterns, such as syncopated rhythms, will normally have comparatively little t
fluctuation, and vice-versa. This may be traced back to the fact that most music is wr
and played for listeners who rely on the presence of patterns of one kind or another in
to perceive meaning in the music. This principle seems to apply to music of many cul
and periods. A general music recognition program should be able to identify the
important kinds of patterns, and to use them as structural clues. This is what our sy
does, in a limited but already quite powerful way, in its analysis of temporal structure.

The current version of the system has been quite successful with the transcription
presented. It has proved capable of dealing with syncopation, tempo fluctuation and nu
of phrasing without losing track of tempo. Performances of pieces composed by
Beethoven, Chopin and Mozart have been analyzed with excellent results. Other s
(ragtimes, Afro-Cuban rhythms, ...) have been handled with similar accuracy. In secti
we will examine in detail the performance of the system on an 18th Century piece tha
been played quite expressively.

1.3 Control Strategies

The level of performance achieved by the system shows that important steps have
taken in the direction of robustness and generality. A closer look reveals that the sy
derives its strength not so much from the individual analysis techniques, all of whic
useful but limited, as from the strategies which order the use of the various techniques.

The problem-solving environment for this research shares many aspects with environ
used in speech recognition research. Multiple layers exist within the system, with fea
or hypotheses corresponding to varying levels of abstraction from the original si
Constraints may be active within a single layer, indicating the degree of compatibili
hypotheses at a given level of description. They may also act across layers, in which
they may be used to propagate hypothesis in bottom-up order (i.e., data-driven) or
down order (i.e., goal-driven) or both. Besides, constraints may express soft prefe
criteria or hard consistency rules.

We will now discuss hypothesis evaluation and pruning, and introduce the use of domina relations. First, it is clear that excessive pruning during the early stages of analysis o results in missing desired solutions. Too little pruning, on the other hand, not only ca inefficiency, but also hides patterns that can could otherwise be used for context driv This will be further discussed below. The system should therefore have ways to mainta balanced degree of indecision.

Pruning methods rely on hypothesis evaluation criteria. In our system, multiple evalua criteria are usually obtained from a variety of considerations. In such a multiple crit situation, a popular approach is to use a weighting scheme to produce a single numeric ra from all the available criteria. However, we consider this to be a method of last res because all too often it leads to erroneous decisions. Trying to improve the approach making the weighting scheme dependent on context leaves the problem unchanged for in decisions (which must be made before sufficient context is established) and forces on develop ways of changing the weights as a function of context, a rather hazard enterprise.

We have chosen instead to retain the multiplicity of criteria as long as possible, and particular during pruning stages. The initial idea behind the scheme we use is quite sim if hypothesis A is no worse than hypothesis B in any of the criteria, and better in at least criterion, then (and only then) B should be pruned. This corresponds to the standard pa ordering of the N-dimensional space of criteria. We have generalized this idea to arbitr partial orderings, which we call *dominance relations*, defined in the space of criteria. say that an hypothesis is *dominated* if it is less than some other hypothesis, with respec the chosen partial ordering. *Undominated* hypotheses are those that correspond to mini elements in the space of criteria. Whenever arbitration is needed between compe hypotheses, the various criteria are formed, and we prune all dominated hypotheses.

Since the retained hypotheses are minimal in the partial ordering, they trade one crite for another: one hypothesis might be closer to the observed data while the other is sim but farther from the data, and a third is intermediate in both respects. When using algorithm in practice, it is often the case that a single (minimum) hypothesis remains, for greatest benefit of subsequent processing. When there are several minimal hypotheses, t are all passed along to higher levels of processing. Usually, we also order them according weighting scheme, but the weights do not affect pruning. The use of dominance relat yields an elegant solution to the pruning problem, since the pruning criterion always t the same form, yet leaves much flexibility in the choice of an appropriate partial order In our application, it is so effective that it is hard to imagine using any other technique.

Another technique we have found useful is the optional use of context in the generation evaluation of hypotheses. Optional use means that generators and evaluators can wor the absence of context parameters (this is typically needed during an initial bottom-up p but will take advantage of contextual information when and if it has been collected. form taken by contextual information varies. It is is often produced via statistics, or o ways of characterizing *group behavior* in a set of previously generated hypoth Contextual information is usually translated to soft constraints rather than hard ones. example, a change in some of the evaluation criteria may bias the system towards hypoth that are *similar* to other ones, but does not remove any hypothesis from consideration. allows local deviations from the observed group behavior. This use of context prov feedback from the global levels to local levels. It does so by influencing an essenti bottom-up scheme via top-down constraints or by *peer pressure*.

Specific examples of the application of these ideas will be discussed in the next section, af the necessary objects and properties are defined. For the purposes of the current discussi consider instead a hypothetical situation, where we assume that the program has someh produced a 1.0 seconds estimate for the duration of the bar. In such a situation, adding contextual assumption that the piece is in 3/4 meter should make it easier to interpre duration of 0.4 sec as 1/3 of a bar (a quarter-note). On the other hand, if we assumed 4 meter, the same duration of 0.4 sec would more likely correspond either to 1/2 of a bar half-note) or to 3/8 of a bar (a dotted quarter). In a top-down scheme, such context assumptions might be tried in succesion.

In fact, our system uses a different approach. It gathers statistical information which highly correlated with the choice of meter, and uses this *metric context* to influe hypothesis evaluation, before any choice is made regarding meter. This happens via *pe pressure*: for example, if hypothesized metric values of 1/3 are obtained with a high rat much more frequently than are 3/8 or 1/2, this should make it more likely to interpret as 1/3. Peer pressure can only be applied to push the interpretation of rather ambiguo situations in the direction already taken for less ambiguous ones. It cannot be used uni there is a fair number of somewhat obvious situations in the first place. Experience w the system has shown this type of reasoning to be quite helpful.

A further aspect of control strategies that we have not yet sufficiently mastered concrete implementation is optimizing the trade-off between the expected cost and expec returns of alternative methods of gathering information. Decisions regarding such tra offs are currently built into the program. Yet, we have taken some steps in the direction dynamic planning, by exploring this idea at a more formal level. The discussion of *Information-Theoretic Approach to Search Strategies in Expert Systems* (Rockmore a Green, 1982) is included in Appendix 3.

To summarize, the art of control strategies is expressed throughout the system in a varie of ways. No single uniform strategy has been found to solve all problems, but we ha found several useful techniques (or design principles) of broad applicability. The three m important ones, which permeate the design of our research system, are (a) approachi multiple criteria decisions in a uniform manner, based on *dominance relations* (this critical in maintaining a balanced level of hypothesis pruning); (b) applying *peer pressure* form of context driving in which one summarizes the group behavior of hypothe previously selected at some level, in order to influence subsequent hypothesis generati and/or evaluation at the same level; and (c) providing alternative approaches to hypothe generation. The combined use of these techniques increases performance (notab robustness) while keeping computational demands low, which is largely the aim of cont strategies.

2. Musical Analysis: Detailed Description

In this section, we follow in some detail the operation of the musical analysis subsystem. T order of presentation follows the analysis of a single example. Successive steps illustrated with intermediate output from the program. This mode of presentation occasionally interrupted to allow for a more abstract presentation of some methods or issu

The example used for this description is an Etude by Chopin, Opus 28, No. 15 from which initial section was selected (see Figure 3a). Naturally, since the system does not yet d with polyphonic inputs, we have recorded and analyzed a single voice piano rendition of t section of the piece (see Figure 3b).

This example was chosen because it is relatively short, allowing a more thorough accoun the program than would otherwise be possible. The CMJ paper focused on a larger exam (Mozart Sonata) but provided less detail in its description of the system.

2.1 Input and Preprocessing

The melody line of the Chopin Etude was played on the piano, recorded, and digitized. It v then handed to the acoustic analysis subsystem, which determined note attacks and provi amplitude and frequency estimates. Figure 4a shows the input file used by the musi analysis. It consists of the output of the acoustic analysis, together with some informat that was added by hand, for debugging convenience. In the figure, the line beginning w PARS (for parameters) indicates the nature and the order of the input fields in the note that follows.

```
PARS BEG DUR FRQ AMP UVAL;
```

The field BEG is the estimated time (in seconds) at which the note occurs, and DUR is the ti elapsed until the beginning of the next note. Observe that DUR is not necessarily the durat of the note, since the acoustic analysis does not detect a trailing rest, only the beginning the next note. (The last note is treated specially).

Since the fields BEG and DUR are related, it is sufficient to provide one of them. All other fie are optional. In order to run, the analysis only needs either BEG or DUR data. Of course, program makes use of additional data (such as frequency, amplitude, ...) when given.

The field FRQ is the estimated frequency in Hertz. (In the analysis of percussion examp this field is omitted, but a stroke description field is normally provided). The field AMP gi an average energy estimate for the event.

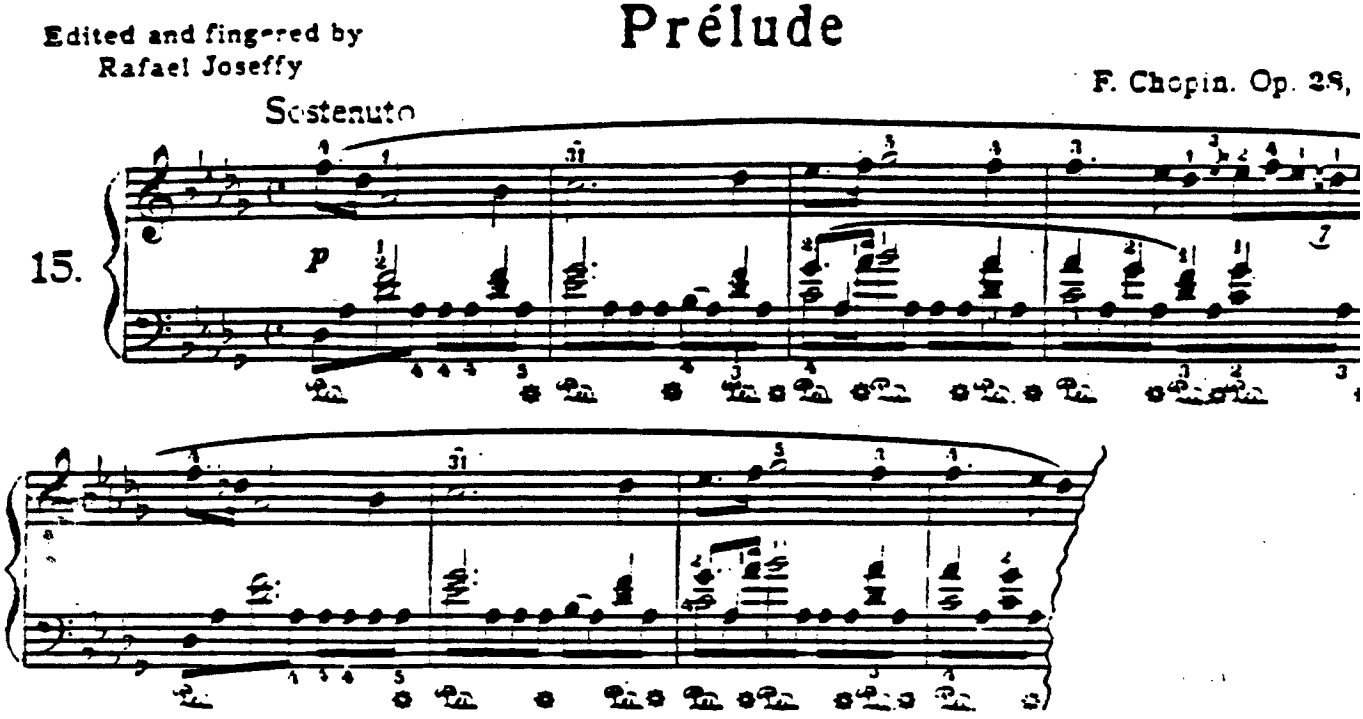

Figure 3a: Excerpt from the Prelude as it is Printed

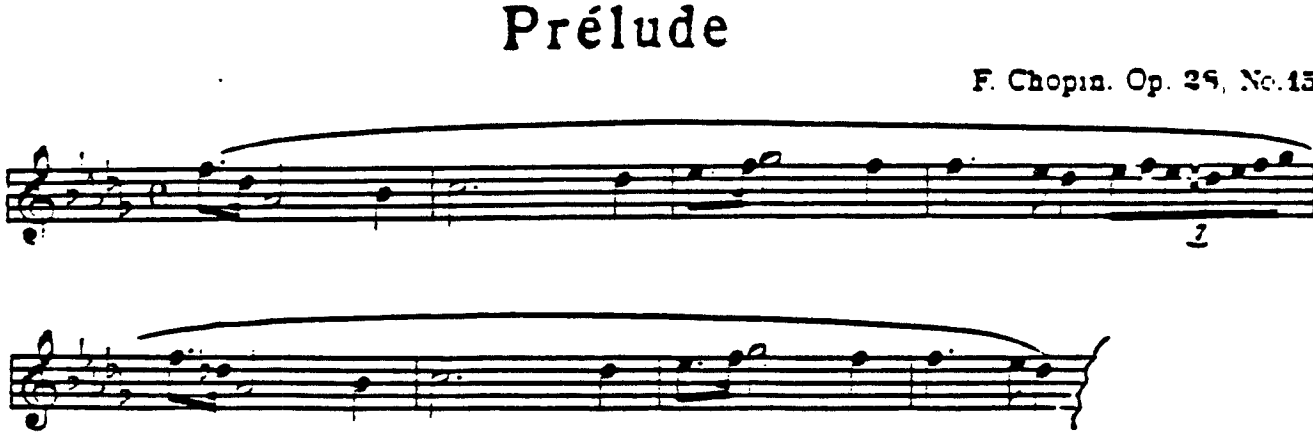

Figure 3b: The Portion that was Played and Analysed

```
TITLE   Chopin Prelude (Opus 28, No15) ;

BARS    4/1  0/1 ;

PARS BEG     DUR   FRQ    AMP   UVAL;

    .000    .800  701.  .740  3/4 ;
    .800    .256  553.  .534  1/4 ;
   1.056   1.824  416.  .596  2/1 ;
   2.880    .960  467.  .769  1/1 ;

   3.840   2.752  522.  .383  3/1 ;
   6.592    .992  553.  .753  1/1 ;

   7.584    .768  624.  .773  3/4 ;
   8.352    .256  701.  .807  1/4 ;
   8.608   1.920  740.  .557  2/1 ;
  10.528    .992  697.  .506  1/1 ;

  11.520   1.408  701.  .573  3/2 ;
  12.928    .576  621.  .451  1/2 ;
  13.504   1.088  553.  .498  1/1 ;
  14.592    .224  620.  .447  1/7 ;
  14.816    .192  702.  .795  1/7 ;
  15.008    .160  624.  .644  1/7 ;
  15.168    .160  585.  .546  1/7 ;
  15.328    .192  621.  .729  1/7 ;
  15.520    .160  702.  .999  1/7 ;
  15.680    .256  740.  .608  1/7 ;

  15.936    .864  701.  .913  3/4 ;
  16.800    .320  553.  .349  1/4 ;
  17.120   1.824  416.  .389  2/1 ;
  18.944   1.024  467.  .350  1/1 ;

  19.968   2.624  522.  .505  3/1 ;
  22.592   1.088  554.  .590  1/1 ;

  23.680    .800  624.  .809  3/4 ;
  24.480    .288  702.  .907  1/4 ;
  24.768   1.888  740.  .567  2/1 ;
  26.656   1.024  701.  .643  1/1 ;

  27.680   1.536  698.  .528  3/2 ;
  29.216    .608  619.  .180  1/2 ;
  29.824   2.160  554.  .170  2/1 ;
```

Figure 4a: Input to Musical Analysis

The fields described so far are outputs of the the acoustic analysis. There are other optio fields, which may be added by the user to help check the results of the musical analysis. T field UVAL (User's preferred value) represents the rhythmic value that the user believes to correct for this note. It is expressed as a fraction of the reference unit, chosen arbitrarily the user. The choice used here (quarter-note) is typical.

When UVAL is defined, the system prints out the differences between the preferred values a the values actually produced by the analysis, which facilitates debugging. In addition, BARS input line now takes effect. It gives the number of units per bar, and the met position of the first bar. In the current example, BARS 4/1 0/1 indicates that there are fo units per bar, and that the first bar occurs at the first note. This hint concerning placement is again used by the program for formatting the output, and/or counti deviations between desired and actual behavior, but not to influence program decisions any way. This terminates the discussion of the input format.

While the input file is scanned, some pre-processing steps are performed, including varic error checks.

First, since the current version of the program is monophonic, and the signal process routines do not handle rests, the duration of an event is taken to be the difference betwe the attack time of the event and the attack time of the next event. Thus BEG and DUR redundant. If both are given as input, the invariant that relates them is checked withi numerical tolerance. If instead only one of BEG and DUR is given, the other is automatica derived using the invariant.

Amplitude estimates are normalized to the maximum amplitude in the piece.

Frequency estimates (in Hertz) are translated to pitches on a 12 semitone scale. There currently no analysis of the tuning system implied by the data. The program assumes the of a standard scale and standard tuning. It would be desirable to have at least a rob retuning method, which could be achieved in a number of ways, but since it has not b necessary so far, we have not implemented it. In Figure 4b, the *CTS* column gives deviati in cents from the standard tuning. For example, the first note is 7 cents above the stand frequency of the F in the 5th octave, and the second note is 5 cents below the standard B in the same octave. Since there are 100 cents in a semitone, the maximum possible round error is 50 cents. Note that in the example, deviations do not exceed 10 cents. If they w larger, an investigation of tuning would be in order.

In any case, after input frequencies have been used to determine positions on the semit scale, subsequent processing uses only these positions. Tuning and conversion to semito should probably be handled by the acoustic analysis back-end, rather than the musi analysis front-end.

```
TITLE   Chopin Prelude (Opus 28, No15) ;

BARS    4/1     0/1 ;


KEY  C  major ;
SIGNATURE  C :  ;

PARS
    BEG    DUR    AMP   FREQ CTS   PIT UVAL   UMPOS ;


@ BAR 1;
   .000   .800   .740   701.   7    F5  3/4     0/1 ;
   .800   .256   .534   553.  -4   Df5  1/4     3/4 ;
  1.056  1.824   .596   416.   4   Af4  2/1     1/1 ;
  2.880   .960   .769   467.   4   Bf4  1/1     3/1 ;
@ BAR 2;
  3.840  2.752   .383   522.  -4    C5  3/1     4/1 ;
  6.592   .992   .753   553.  -4   Df5  1/1     7/1 ;
@ BAR 3;
  7.584   .768   .773   624.   6   Ef5  3/4     8/1 ;
  8.352   .256   .807   701.   7    F5  1/4    35/4 ;
  8.608  1.920   .557   740.   1   Fs5  2/1     9/1 ;
 10.528   .992   .506   697.  -4    F5  1/1    11/1 ;
@ BAR 4;
 11.520  1.408   .573   701.   7    F5  3/2    12/1 ;
 12.928   .576   .451   621.  -4   Ef5  1/2    27/2 ;
 13.504  1.088   .498   553.  -4   Df5  1/1    14/1 ;
 14.592   .224   .447   620.  -6   Ef5  1/7    15/1 ;
 14.816   .192   .795   702.  10    F5  1/7   106/7 ;
 15.008   .160   .644   624.   6   Ef5  1/7   107/7 ;
 15.168   .160   .546   585.  -7    D5  1/7   108/7 ;
 15.328   .192   .729   621.  -4   Ef5  1/7   109/7 ;
 15.520   .160   .999   702.  10    F5  1/7   110/7 ;
 15.680   .256   .608   740.   1   Fs5  1/7   111/7 ;
@ BAR 5;
 15.936   .864   .913   701.   7    F5  3/4    16/1 ;
 16.800   .320   .349   553.  -4   Df5  1/4    67/4 ;
 17.120  1.824   .389   416.   4   Af4  2/1    17/1 ;
 18.944  1.024   .350   467.   4   Bf4  1/1    19/1 ;
@ BAR 6;
 19.968  2.624   .505   522.  -4    C5  3/1    20/1 ;
 22.592  1.088   .590   554.  -1   Df5  1/1    23/1 ;
@ BAR 7;
 23.680   .800   .809   624.   6   Ef5  3/4    24/1 ;
 24.480   .288   .907   702.  10    F5  1/4    99/4 ;
 24.768  1.888   .567   740.   1   Fs5  2/1    25/1 ;
 26.656  1.024   .643   701.   7    F5  1/1    27/1 ;
@ BAR 8;
 27.680  1.536   .528   698.  -1    F5  3/2    28/1 ;
 29.216   .608   .180   619.  -9   Ef5  1/2    59/2 ;
 29.824  2.160   .170   554.  -1   Df5  2/1    30/1 ;

@ PARS
    BEG    DUR    AMP   FREQ CTS   PIT UVAL   UMPOS ;
```

Figure 4b: Preprocessed Input

In Figure 4b also, the UMPOS column (User-provided metric position) results from summatic of the UVAL input, beginning at 0/1. This column keeps track of musical time *according to th user*, as opposed to musical time as determined by the program, for the purpose of checkin the latter. In particular, whenever the metric position (minus the metric offset al provided by the user -- 0 in this example) becomes a multiple of the bar length, a bar mark printed.

Some easily identifiable signal processing errors can be corrected at this early stage. Not which have either an extremely low amplitude, or both a low amplitude and a low duratio are suspected of being false attacks, due to errors in the acoustic analysis. They are compare to surrounding notes, in terms of both duration and amplitude. If they turn out to l significantly low in the context of their neighbors, they are removed from the note lis Simple rules (based on pitch comparisons) determine whether the spurious note is merg with the previous one or the next. Adjacent attack times and durations are adjusted. In th current example, no signal processing artifacts are present. (In the analysis of the Moza Sonata, five false attacks are corrected, and this greatly facilitates the subsequent analysis Ideally, once an event is suspect and an appropriate context has been established, it would l best to feed this information back to the acoustic analysis for further investigatio Feedback links from the musical analysis to the acoustic analysis are not possible in th current hardware configuration, but such links will be added as soon as possible.

After these pre-processing steps, real work may begin.

2.2 Important Events and Accents

The current system has two initial sets of handles on temporal structure: *important even* and *important durations*. The heuristic methods that detect important events a independent from those which detect important durations, and focus on different aspects the data. This is a source of strength for the program, since either type of handle may l useful at any point in a piece. Important durations are discussed in Section 2.3.

The importance of structural accents was discussed at some length in the CMJ paper. The are potential *anchor points* for the temporal structure of a piece. The program must be ab to identify structural accents quite early in the analysis, at a time when little if any *understood* about the piece. (Thus, the notion of accent used here is not necessarily th which a music analyst would use). There are currently two kinds of accents, rhythm accents and melodic accents. Dynamic accents have not been implemented, awaiting a bett model of their perception.

Rhythmic accents essentially correspond to long notes (*agogic* accents). To be precise, a no gets an accent if it is *clearly longer* than the previous note, and the following note does n deserve an accent by the same rule. The training method used to define the meaning *clearly longer*, and the related psycho-acoustic research, were fully discussed in the CM paper, so we will omit this discussion here.

Melodic accents are related to particular patterns of pitch and duration. A search for specif pitch patterns is made within sequences of notes of roughly equal duration. The patter currently include stepwise ascending and descending sequences, and trills. Tunab thresholds control the minimum length necessary for the pattern to fire. Sequences

roughly equal duration may be extended to include a longer note at either end, and t changes the length thresholds for pattern detection.

```
TITLE   Chopin Prelude (Opus 28, No15) ;

BARS    4/1     0/1 ;

KEY  Df major ;
SIGNATURE  Df: Bf Ef Af Df Gf  ;

PARS
    BEG   DUR D  PIT pr TRS;

∂ BAR 1;
   .000  .800 >   F5 T-  ~ ;
   .800  .256 <   D5 J-  ~ ;
  1.056 1.824 >   A4 S+  R ;
  2.880  .960 <   B4 S+  ~ ;
∂ BAR 2;
  3.840 2.752 >   C5 S+  R ;
  6.592  .992 >   D5 S+  ~ ;
∂ BAR 3;
  7.584  .768 >   E5 S+  ~ ;
  8.352  .256 <   F5 S+  ~ ;
  8.608 1.920 >   G5 S-  R ;
 10.528  .992 <   F5 ..  ~ ;
∂ BAR 4;
 11.520 1.408 >   F5 S-  R ;
 12.928  .576 <   E5 S-  ~ ;
 13.504 1.088 >   D5 S+  R ;
 14.592  .224 =   E5 S+  ~ ;
 14.816  .192 =   F5 S-  ~ ;
 15.008  .160 =   E5 S-  ~ ;
 15.168  .160 = Eff5 S+  ~ ;
 15.328  .192 =   E5 S+  ~ ;
 15.520  .160 <   F5 S+  ~ ;
 15.680  .256 <   G5 S-  ~ ;
∂ BAR 5;
 15.936  .864 >   F5 T-  R ;
 16.800  .320 <   D5 J-  ~ ;
 17.120 1.824 >   A4 S+  R ;
 18.944 1.024 <   B4 S+  ~ ;
∂ BAR 6;
 19.968 2.624 >   C5 S+  R ;
 22.592 1.088 >   D5 S+  ~ ;
∂ BAR 7;
 23.680  .800 >   E5 S+  ~ ;
 24.480  .288 <   F5 S+  ~ ;
 24.768 1.888 >   G5 S-  R ;
 26.656 1.024 <   F5 ..  ~ ;
∂ BAR 8;
 27.680 1.536 >   F5 S-  R ;
 29.216  .608 <   E5 S-  ~ ;
 29.824 2.160 >   D5 ..  R ;

∂ PARS
    BEG   DUR D  PIT pr TRS;
```

Figure 5: Accents, Pitch, Key

Figure 5 shows the accents that were obtained in the current example. The column label shows duration relationships (clearly longer, about equal, clearly shorter) printed as >, =, < respectively. The resulting rhythmic accents show with an R under the TRS heading.

Also under the TRS heading, an S is used to indicate a sequence accent, while s marks the ot notes in the sequence. Similarly, T marks the beginning of a trill, and t marks the other n in the trill. It is possible for a note to be both a rhythmic and a melodic accent (sequenc trill). However, in this example, the simple melodic patterns recognized here do not occu

The accents just determined provide a list of *important events*, which is useful in sev ways. The first use is to guide the determination of the key in which the piece is writ The current method assumes that the entire piece is in a single key. It is described in the paper (see Appendix 1). In this case, the program has determined that the key is Df ma and announces the five flats in the key signature (Bf Ef Af Df Gf). The use of this signature is implied in all subsequent printing of the pitch names. The interested reader compare the names of the pitches in Figure 4b, before the key is known (C major is use default), and Figure 5.

The second use of the accents is to provide anchors for the temporal structure. T important events, together with the first and the last event, constitute the initial struct anchors, used in the determination of the tempo line, which is our next topic. The *in bridges* are the time spans from accent to accent (see Figure 6).

```
INITIAL BRIDGES

bridge from   .000 to  1.056, dur   1.056
bridge from  1.056 to  3.840, dur   2.784
bridge from  3.840 to  8.608, dur   4.768
bridge from  8.608 to 11.520, dur   2.912
bridge from 11.520 to 13.504, dur   1.984
bridge from 13.504 to 15.936, dur   2.432
bridge from 15.936 to 17.120, dur   1.184
bridge from 17.120 to 19.968, dur   2.848
bridge from 19.968 to 24.768, dur   4.800
bridge from 24.768 to 27.680, dur   2.912
bridge from 27.680 to 29.824, dur   2.144
bridge from 29.824 to 31.984, dur   2.160
```

Figure 6: Accent to Accent Intervals

2.3 Important Durations and Pulses

The CMJ paper (Appendix 1) went on from here to determine the tempo line, but the me does not extend to syncopated rhythms, and we need an independent source of clues t tempo variation. These will be provided by what we call *important durations*.

There are two ways that a duration can become important. The first is to be the av duratio:: in a *pulse train*, which is a group of successive repeated durations. This inv

thresholds for the maximum acceptable variation within the group, and for the mini group length. The notes included in a given pulse train will later be treated as if they w the same duration.

The other way for a duration to be important is to occur frequently. This statist approach is used in intervals (time ranges) not covered by pulse trains. If a time rang very small, statistics would be worthless, and the important duration is propagated from previous pulse train. If instead the time range is quite large, so that it could encompass a amount of global tempo fluctuation, it gets divided recursively until the time range is sn enough.

The important duration for a given time range corresponds to the highest peak in a speci constructed histogram of the durations in the range. Figure 7 shows the last histog obtained for the performance of the Chopin piece (the time range is from 24.5 to 32 second

A first peculiarity of this histogram is that we use a logarithmic scale for durations. Thi because the ratios of durations are more important to the analysis than their differences. discrete scale uses an integer number of divisions for every factor of 2 in duration, so t ratios equal to a power of 2 correspond to a shift in index. In addition, since we use divisions per factor of 2, ratios of 6; 3, 3/2, 3/4, ..., and their inverses also correspond integer shifts (with a negligible error). This is because the ratio of 3/2 corresponds alm exactly to 7 divisions on the scale, a fact that will not surprise anyone familiar with construction of the equal tempered scale, in which the interval of a fifth (3/2) is v closely approximated by 7 semitones.

```
SMOOTHED HISTOGRAM OF LOG(DUR), time range [ 24.480 -- 31.984]

DURATION | DENSITY  (pulse is 1.498)
    .088 |
    .094 |
    .099 |
    .105 |
    .111 |
    .118 |
    .125 |
    .132 |
    .140 |
    .149 |
    .157 |
    .167 |
    .177 |
    .187 |
    .198 |
    .210 |
    .223 |-
    .236 |---
    .250 |--------
    .265 |--------------
    .281 |--------------------
    .297 |--------------------
    .315 |-------------
    .334 |--------
    .354 |---.
    .375 |-
    .397 |
    .420 |
    .445 |
    .472 |-
    .500 |-----
    .530 |------------
    .561 |--------------------
    .595 |----------------------------
    .630 |--------------------------
    .667 |------------------
    .707 |----------
    .749 |----
    .794 |--
    .841 |------
    .891 |--------------
    .944 |------------------------
   1.000 |----------------------------------
   1.059 |--------------------------------
   1.122 |----------------------
   1.189 |--------------
   1.260 |-----------
   1.335 |-----------------
   1.414 |---------------------------
   1.498 |****************************************
   1.587 |--------------------------------------
```

Figure 7: Pulse Detection, Sample Histogram

A second peculiarity of the histograms is that a weighting scheme is used to emphasize lon notes. Without the weights, short notes would seem comparatively too important, si they occur much more frequently. Making the weights proportional to the duratic themselves would instead favor the longer notes. The compromise used is to give duratio the weight w(D) = D**P, where the power P is chosen between 0 and 1 (the value p=0.4 been chosen empirically).

Thirdly, histograms are smoothed. Each duration contributes to a number of neighbor ce in the histogram. This amounts to a convolution of the unsmoothed histogram with a b shaped curve of appropriate bandwith. The effect is a partial correction of quantizati noise, and an averaging effect of moderately distant durations.

The important duration selected for a given time range is the highest peak in the smooth histogram. Secondary peaks are significant as well, but are currently not used.

2.4 Pulse Line and Reference Unit

The *important durations* found in this analysis, from either of the available techniques, summarized in the *pulse line* (see Figure 8). This table gives for each important durati called a *pulse* for short, the pulse value, the time range over which it is active, the meth used to compute it (pulse train or statistical profile) and the number of events upon wh the pulse is based. In addition, successive pulses are compared, and guesses are m concerning their ratio. The next section explains how this is done. For example, successive pulses are 0.37 and 0.71, one would interpret this as a change in metric unit b factor of 2/1, combined with tempo adjustment by the factor 0.71/(2*0.37) = 0.71/0.7 0.96.

PULSE LINE

Time range	Value count	Pulse	=	Units	*	Norm	(method used)
.000 -- 8.352	5	.944	=	1/1	*	.944	(profile)
8.352 -- 14.816	6	1.000	=	1/1	*	1.000	(profile)
14.816 -- 15.680	5	.173	=	1/6	*	1.037	(pulse train)
15.680 -- 24.480	6	1.000	=	1/1	*	1.000	(profile)
24.480 -- 31.984	4	1.498	=	3/2	*	.999	(profile)

Figure 8: Summary of Important Durations

Another decision is made from the combined set of pulses: the choice of a *reference unit* metric values. This unit is needed to express metric value hypotheses in terms o consistent unit. All metric values from then on, beginning with those in the *pulse line*, expressed in terms of the reference unit.

The reference unit is typically a half-note, a quarter-note or an eighth-note. Which one i does not matter at this point. In the current example, the reference unit turns out to be beat (quarter-note).

The decision is made by choosing, among the pulses, one that has maximally sim relationships with all other pulses. The simplest relationship is the 1:1 relationship. N come 2:1 and 1:2. Relationships such as 3:4, 4:3, 1:6 or 6:1 more complex, but not as comp as, say, 6:7 or 27:32. The topic of rational approximations and their complexity is discus in the next section. The current scheme for choosing the reference unit, even thoug handles many cases quite well, can be fooled by a rapid accelerando, sustained over extended period of time. A better scheme, whose implementation is planned, will integ the rules used to make pulse line decisions with similar rules concerning the tempo line. latter are discussed in section 2.6.

2.5 Rational Approximation Generation

An important aspect of the operation of a transcription program is that it approxim relationships between performed durations by simple metric proportions. Since the meth used to generate rational approximations are critical to the success of the program, discuss this topic in some detail.

To get started, suppose that we have a local estimate of the metric unit, say 1.5 seconds, that we want to assign a metric value to a performed duration of 0.6 seconds. The r 0.6/1.5 = 0.4 is the target to be approximated by a *reasonable* fraction of the metric u Hypotheses such as 1/2, 1/3 and 3/8 should definitely be considered. On the other hand, 3/4 and 1/1 should probably be rejected for not being close enough to the target. We wo also reject 2/5, even though it is remarkably close to 0.4, if a stylistic constraint exclu denominators with factors other than 2 and 3; in a different situation within the progr or under different stylistic constraints, the ratio 2/5 might be perfectly acceptable, but necessarily the best hypothesis.

The choice of rational approximations must take into account at least two criteria: closer to the data (or *fit*) and simplicity of the fraction. The latter must be understood in term how simple or natural the resulting musical notation would be. It is also clear that context of a note, both locally (e.g., the durations of nearby notes) and globally (e.g., m hypotheses) should have a part in the final choice. Since most contextual constraints can be expressed until metric hypotheses are formulated over an entire region, we must rely rational approximation generator to supply a number of hypotheses, and postpone the ch between the generated hypotheses until sufficient context has been acquired. A g rational approximation generator is one that produces all *reasonable* hypotheses, and v few unreasonable ones, while being given little contextual information.

The following control parameters are supplied to the generator: a set of acceptable frac denominators, given explicitly or via generation rules; a set of constraints on numera (fractions are always expressed in lowest terms); a way to compare the simplicity of fractions; and a measure of the *fit* between a given real number and a fractio approximation. The basic idea of the algorithm is to consider (at least implicitly) all poss fractions N/D, to rate them by the two criteria of simplicity and fit, and to retain only fractions which compare reasonably well with others in terms of *both* criteria. comparison uses a suitable partial ordering of the two-dimensional space of criteria, base the given orderings of individual criteria. Only the hypotheses that are minimal in partial ordering are retained.

For the partial ordering in two dimensions, we currently use the standard cross-product the (partial or total) orderings of individual criteria. We say that (x1,y1) dominates (x2,y if if x1≤x2 and y1≤y2, and (x1, y1) is not equal to (x2, y2). The selection of minir hypotheses amounts to the elimination of dominated hypotheses.

When both criteria are totally ordered, the selection rule can be implemented ve efficiently, provided one can generate hypotheses in increasing order of one of the crite The program currently uses a scheme which allows more freedom in the choice of crite and ordering relations, but still leads to a reasonable implementation. One maintains at times a set of undominated hypotheses. A newly generated hypothesis is discarded if it found to be dominated. Otherwise, it is added to the current set, which may cause otl hypotheses to be discarded, if they become dominated.

Returning to the example, we want to find approximations for 0.4 (obtained as the ratio c duration of 0.6 sec to a hypothesized local metric unit of 1.5 sec). Let us suppose that acceptable denominators include 1, 2, 4, 8, 16, as well as 3, 6, 12, and that numerators unconstrained. For the measure of fit, let us use absolute difference, with the usual tc order, and a worst fit threshold of 0.3. For intrinsic complexity, assume for now that complexity of a reduced fraction N1/D1 is less or equal that of N2/D2 iff N1 divides N2 a D1 divides D2. According to this partial ordering, which is a rather weak one, 1/2 is simp than 3/2 or 1/4 or 1/6, but none of the last three fractions is simpler than another. Un these conditions, the algorithm yields five approximations of 0.4. Ordered by increasing (given in parentheses), they are: 5/12 (.0166), 3/8 (.025), 7/16 (.0375), 1/3 (.066), and (.1). Note that 1/1 (.6) has not been considered because .6 exceeds the fit threshold of .3, a that 1/4 (.15) has been discarded because it is dominated by 1/2 (.1).

Suppose that we used a stronger complexity ordering, according to which the complexity N1/D1 is less or equal that of N2/D2 if N1≤N2 and D1≤D2. The effect would be to disc 7/16 (.0375), which is now dominated by 3/8 (.025), leaving us with the four hypothe 5/12, 3/8, 1/3, and 1/2. If we also had a constraint on numerators which excludes 5, would be left with three hypotheses, 3/8, 1/3 and 1/2.

The earlier result and its variations illustrate the range of behaviors of the system's ratio approximation generator. The main point is that the generator operates on local informati and leaves a small number of reasonable hypotheses for consideration in larger contexts. concept of *dominance* plays a major role in the construction of a good generator.

The program relies on the same generator for a variety of different tasks, by vary parameters such as the acceptable numerators and denominators, the measure of complex and the error thresholds. The most obvious use is in the generation of hypotheses individual note values. We have discussed another use in the preceding section, i.e., construction of the pulse line. Further uses are discussed below and in the next section.

The actual program differs from the description above in minor ways. First, the genera refrains from forming linear combinations of criteria in its selection rules, but it d embody an overall rating, called the *cost* of the hypothesis, computed as a linear combinat of closeness of fit and simplicity. The fractions returned are ordered by increasing cost help calling routines when they need to make an easy decision among the retur hypotheses.

Second, the generator is not given as input the ratio dur1/dur2 between the duration du to be approximated, and the reference duration dur2. Instead, it is given both durations du and dur2. This allows the program to deal with numbers expressed a known physical sc (e.g. milliseconds) so that the uncertainty in the given durations can be taken into accou The effect of throwing in small uncertainty margins is negligible when the durations a large, but in the case of small durations this tends to orient approximations towards simp ones, which is desirable. For example, one may want to consider fewer distinct hypothe for .08/1.0 than for .40/5.0, because small differences in the first case are perceptua indistinguishable.

Another property of the implementation is that the complexity measure for metric fracti changes over time. The a priori measure (used before any context is available) is la replaced by an a posteriori measure which takes context into account. For exam fractions such as 1/3 and 2/3 may become *simpler* than 1/4 or 1/2 after sufficient statisti evidence of ternary meter has been gathered. The implementation of this idea will discussed in the sequel.

2.6 Tempo Line Decisions

After this long digression, which allowed us to discuss the generation of ratio approximations, we return to the step by step description of the program. So far, we ha chosen a set of accented notes, soon to be used as structural anchors, and we have used independent source of cues to determine important durations (called *pulses*) in success time ranges. We have also established simple rational relationships between success pulses, and chosen a reference pulse, which gives the unit in which all metric values w now be expressed.

The next step is to construct a line-segment approximation to the correspondence betw physical time (the actual performance timings) and musical time (the notated time). T will be the initial tempo line. For the endpoints of the line segments, which we structural anchors, we use the attack times of accented notes (as well as the first and l attack). The problem at hand is thus to assign a metric value to each of the interv between structural anchors, called *bridges*.

The method we use represents a compromise between two types of clues. The first type clue is that successive bridge lengths should be in simple rational relationships with e other. The older version of the program, described in the CMJ paper, relied exclusively this type of clue, which seemed sufficient for the 18th Century pieces considered. The n for a second type of clue became obvious when we analyzed ragtimes and percussive mu which have more syncopation. This is what led us to use pulses, and eventually to build pulse line. According to this second clue to tempo, bridge lengths should be in sim rational relationships to the local pulse values.

Since the rational approximation generator essentially trades closeness of fit for simplicity ratios, the simpler the ratios are, the more timing fluctuation is tolerated. This work principle of the rational approximation generator seems to reflect a similar principle ab how people perceive music. While the older program essentially relied on fixed threshold tempo fluctuation from bridge to bridge, the current version is much more adaptive.

In order to keep things reasonably simple, the decisions are made from left to right, tha the number of units assigned to various bridges is determined in temporal order. In sc cases, it might seem better to use a different order (safest first) but this would complic the control structure a great deal, and make the program less stream-oriented.

The program combines the different available clues in the following way. Suppose we w to assign to the current bridge, whose duration is D seconds, a rational number of refere units. It would seem that this is done by first applying the rational approximation genera to D and U, where U is the local estimate of the reference unit, in seconds, and then someh choosing one of the produced approximations. However, we do not have a single value c to consider. The pulse line gives a local estimate PU of the duration on the reference u Another estimate, BU, of the same quantity is based on past bridges: we divide the combi duration of the last N bridges by their combined numbers of units, which have been assig by recent decisions. A suitable value of N is chosen, typically N=2.

When using BU (resp., PU) for an estimate of the local reference unit, one deals with bridge bridge (resp., bridge to pulse) tempo fluctuation. In this way, we gain access to information on which the two types of clues are based. The rational approximat generator is applied to obtain metric hypotheses for the bridge duration D, separately both estimates BU and PU. The program then uses a number of rules to make its decisio based on the hypothesis sets BH and PH returned by the generator. The rules form a sm production system, which is shown in Figure 9. The first rule which fires makes decision. No further rules are evaluated.

```
In evaluating the rules, we use the following symbols:
    - D is the duration of the bridge;
    - PU is the local estimate of the reference unit, from the pulse line;
    - BU is the local estimate of the reference unit, from previous bridges;
    - approximate(D, U) returns a set of rational approximations of D/U.
The rules are defined below:


    Let PH = approximate(D, PU);
    Let BH = approximate(D, BU);

STRONG AGREEMENT RULE:
        IF (unique(BH) or unique(PH))
        AND best(BH) = best(PH)
        THEN choose best(BH)

WEAK AGREEMENT RULE:
        IF stands_out(BH)
        AND stands_out(PH)
        AND best(BH) = best(PH)
        THEN choose best(BH)


    Let CU = (BU + PU)/2;
    Let CH = approximate(D, CU);

UNIQUE COMPROMISE RULE:
        IF unique(CH)
        THEN choose best(CH)

SUBSET RULE:
        IF unique(BH)
        THEN choose best(BH)
        ELSE IF unique(PH)
        THEN choose best(PH)

INITIAL FALL-BACK RULE:
        IF there are no previous bridges
        THEN choose best(CH)

COMBINED COST RULE:
        choose_least_combined_cost_hyp_in(CH)
```

Figure 9: Rule System for Tempo Line Decisions

In this figure, the predicate `unique(H)` is true if the set H of hypotheses contains a sing member. The function `best(H)` returns the hypothesis with minimum cost in H. Cost mea the linear combination of complexity and fit computed by the approximation numb generator. Figure 10a shows the production rules in action on the Chopin example.

```
AT   .000, bridge of length  1.056
  PU is  .944,  guesses for 1.056 are  1/1  ( .08792)
  BU is  .944,  guesses for 1.056 are  1/1  ( .08792)
  STRONG AGREEMENT RULE  -->  treat  1.056  as 1/1 units

AT  1.056, bridge of length  2.784
  PU is  .944,  guesses for 2.784 are  3/1  ( .02767)
  BU is 1.056,  guesses for 2.784 are  5/2  ( .11737)
  CU is 1.000,  guesses for 2.784 are  3/1  ( .08625)
  UNIQUE COMPROMISE RULE  -->  treat  2.784  as 3/1 units

AT  3.840, bridge of length  4.768
  PU is  .944,  guesses for 4.768 are  5/1  ( .06474)
  BU is  .960,  guesses for 4.768 are  5/1  ( .06090)
  STRONG AGREEMENT RULE  -->  treat  4.768  as 5/1 units

AT  8.608, bridge of length  2.912
  PU is 1.000,  guesses for 2.912 are  3/1  ( .04151)
  BU is  .944,  guesses for 2.912 are  3/1  ( .03896)
  STRONG AGREEMENT RULE  -->  treat  2.912  as 3/1 units

AT 11.520, bridge of length  1.984
  PU is 1.000,  guesses for 1.984 are  2/1  ( .00773)
  BU is  .960,  guesses for 1.984 are  2/1  ( .03101)
  STRONG AGREEMENT RULE  -->  treat  1.984  as 2/1 units

AT 13.504, bridge of length  2.432
  PU is 1.037,  guesses for 2.432 are  5/2  ( .12495),  2/1  ( .16368)
  BU is  .979,  guesses for 2.432 are  5/2  ( .06725)
  STRONG AGREEMENT RULE  -->  treat  2.432  as 5/2 units

AT 15.936, bridge of length  1.184
  PU is 1.000,  guesses for 1.184 are  1/1  ( .13862)
  BU is  .981,  guesses for 1.184 are  1/1  ( .15268),  3/2  ( .19175)
  STRONG AGREEMENT RULE  -->  treat  1.184  as 1/1 units

AT 17.120, bridge of length  2.848
  PU is 1.000,  guesses for 2.848 are  3/1  ( .06391)
  BU is 1.033,  guesses for 2.848 are  3/1  ( .09791)
  STRONG AGREEMENT RULE  -->  treat  2.848  as 3/1 units

AT 19.968, bridge of length  4.800
  PU is 1.000,  guesses for 4.800 are  5/1  ( .09890)
  BU is 1.008,  guesses for 4.800 are  5/1  ( .10786)
  STRONG AGREEMENT RULE  -->  treat  4.800  as 5/1 units

AT 24.768, bridge of length  2.912
  PU is  .999,  guesses for 2.912 are  3/1  ( .04034)
  BU is  .956,  guesses for 2.912 are  3/1  ( .02625)
  STRONG AGREEMENT RULE  -->  treat  2.912  as 3/1 units

AT 27.680, bridge of length  2.144
  PU is  .999,  guesses for 2.144 are  2/1  ( .06922)
  BU is  .964,  guesses for 2.144 are  2/1  ( .10222)
  STRONG AGREEMENT RULE  -->  treat  2.144  as 2/1 units

AT 29.824, bridge of length  2.160
  PU is  .999,  guesses for 2.160 are  2/1  ( .07662)
  BU is 1.011,  guesses for 2.160 are  2/1  ( .06498)
  STRONG AGREEMENT RULE  -->  treat  2.160  as 2/1 units
```

Figure 10a: Tempo Line Decisions

The combined cost rule, used only when all else fails, is not needed in the Chopin examp
Figure 10b contains some excerpts from the analysis of a different example, the Moza
Sonata discussed in the CMJ paper, which illustrate the operation of the combined cost ru
This rule selects an hypothesis with minimum *combined cost* in CH, the set obtained from t
compromise estimate. The combined cost of hypothesis H for duration D is the sum of separa
costs, expressing the fact that one wants both a good relationship to CU, the duration of t
compromise reference unit, and to the duration and metric value of the previous brid
Note that the hypothesis selected is not always the one preferred by BU, PU or even CU ratin
and that it does not always have the simplest relationship to the previous bridge eith
Thus all these criteria carry some weight in the decision, but one resorts to numeri
combinations of ratings only when stronger forms of selection have all failed.

```
      ...
      ...
      ...

AT 22.158, bridge of length  1.701
  PU is  .235,  guesses for 1.701 are  8/1  ( .07007),   7/1  ( .07026)
  BU is  .249,  guesses for 1.701 are  7/1  ( .06605),   6/1  ( .08751)
  CU is  .242,  guesses for 1.701 are  7/1  ( .05559),   8/1  ( .08404)

  cost combinations                Left   Right  Ratio   Sum of costs
                                   .287  1.701    .169

                                   1/1     7/1     1/7
                                  .070    .056    .105    .230

                                   1/1     8/1     1/8
                                  .070    .084    .152    .306

  COMBINED COST RULE  -->  treat  1.701  as 7/1 units

      ...
      ...
      ...

AT 25.859, bridge of length  2.084
  PU is  .223,  guesses for 2.084 are 10/1  ( .09674),   9/1  ( .09732)
  BU is  .247,  guesses for 2.084 are  8/1  ( .04860),   9/1  ( .11472)
  CU is  .235,  guesses for 2.084 are  9/1  ( .08068),   8/1  ( .08163)

  cost combinations                Left   Right  Ratio   Sum of costs
                                  2.000  2.084    .960

                                   8/1     9/1     8/9
                                  .015    .081    .130    .225

                                   8/1     8/1     1/1
                                  .015    .082    .054    .151

  COMBINED COST RULE  -->  treat  2.084  as 8/1 units

      ...
      ...
      ...

AT 31.022, bridge of length  2.862
  PU is  .265,  guesses for 2.862 are 11/1  ( .08466),  12/1  ( .09649)
  BU is  .257,  guesses for 2.862 are 12/1  ( .07357),  11/1  ( .08167)
  CU is  .261,  guesses for 2.862 are 11/1  ( .07297),  12/1  ( .08510)

  cost combinations                Left   Right  Ratio   Sum of costs
                                  2.061  2.862    .720

                                   8/1    11/1    8/11
                                  .023    .073    .044    .140

                                   8/1    12/1     2/3
                                  .023    .085    .121    .229

  COMBINED COST RULE  -->  treat  2.862  as 11/1 units
```

Figure 10b: Tempo Line Decisions (use of combined cost rule)

The result of the successive tempo line decisions, for the Chopin example, is shown in Fi 11, in the UNITS ADDED column. The UNITS TOTAL column shows the chosen metric u summed from the beginning of the piece. One can track the evolution of the tempo with UNIT LENGTH, where the reference unit is given both in seconds and in metronomic m (number of units per minute).

TEMPO LINE

REAL TIME (SEC)	UNITS (TOTAL)	(ADDED)	UNIT LENGTH (SEC)	(MM)
.000	0/1	1/1	1.056	57
1.056	1/1	3/1	.928	65
3.840	4/1	5/1	.954	63
8.608	9/1	3/1	.971	62
11.520	12/1	2/1	.992	60
13.504	14/1	5/2	.973	62
15.936	33/2	1/1	1.184	51
17.120	35/2	3/1	.949	63
19.968	41/2	5/1	.960	63
24.768	51/2	3/1	.971	62
27.680	57/2	2/1	1.072	56
29.824	61/2	2/1	1.080	56
29.824	65/2			

Figure 11: Tempo Line (before modifications)

2.7 Tempo Line Modification

The tempo line decisions made so far are not final. In fact, since they were based on l criteria, no attempt had been made to cast them in the larger context of global me regularities. This will be now be done.

First of all, the reference unit chosen is typically a rather small unit, e.g. an eight-note quarter-note. A primitive form of metric regularity is represented by the choice of a la grouping unit -- let us call it the *base unit* -- which captures periodicities roughly at level of a bar (it could be also be two bars, or a half of one).

The base unit is determined by looking for periodicities in the tempo line, i.e. in distribution of metric bridge lengths. In order to collect possible periods (candidate l units), the program uses both individual bridge lengths, and the lengths resulting f associating up to three successive bridges. Local syncopation in the metric accents, whet truly present in the music or resulting from erroneous decisions in the program, cre strange trees which obscure the forest (the metric hierarchy). These potential difficul are alleviated to some degree by the use of local grouping, together with statistics mutual support of simply related metric lengths. The method below, first described us the current example, gives a first approximation of the metric hierarchy.

As shown in Figure 12, successive metric bridge lengths are collected, and the sums of u 3 successive lengths are formed, leaving out fractions which do not represent an inte

The statistic of how many times each number of units occurs is only mildly interesting. the Chopin example, the three highest counts are obtained by 3, 5 and 8 metric units, e with a count of 4. Next come 2 and 4 units, with counts of 3. A much more interest statistic is obtained by the use of *multiplier support*. The idea is that 4 units, for exam becomes a better choice of the base unit if simple multiples (8, 12 or 16 units) and sim divisors (1 or 2 units) also have high counts. Thus, we arrange for multipliers and divis to contribute to candidate units, in proportion of their own count, but with a weight t depends on the multiplier or divisor chosen. Only powers of 2 and 3 are used h representing a definite bias towards the commonly useful hierarchical relationships.

```
STATISTICS FOR CHOICE OF BASE UNIT

bridge lengths (in reverse temporal order)
2/1, 2/1, 3/1, 5/1, 3/1, 1/1, 5/2, 2/1, 3/1, 5/1, 3/1, 1/1

counts of integral bridge lengths, cumulating up to 3 successive lengths

 7/1 ( 1.000), 10/1 ( 2.000),  1/1 ( 2.000),  9/1 ( 2.000),
11/1 ( 2.000),  4/1 ( 3.000),  2/1 ( 3.000),  5/1 ( 4.000),
 8/1 ( 4.000),  3/1 ( 4.000)

counts augmented by multiplier support

 7/1 ( 1.000), 11/1 ( 2.000),  9/1 ( 3.333), 10/1 ( 4.000),
 5/1 ( 5.000),  3/1 ( 5.333),  1/1 ( 6.083),  8/1 ( 6.500),
 2/1 ( 6.500),  4/1 ( 7.000)

 selected base unit: 4/1
```

Figure 12: Choice of Base Unit

In the example, the highest rating (with multiplier support) is obtained by the candid consisting of 4 reference units, and the program makes this the base unit. The next lov ratings are 2 units, 8 units, and 1 unit. These are excellent choices for the piece, since base unit (4 units) turns out to be the bar, and the 2- and 1-unit divisions are the pro subdivisions (the piece is in 4/4). Furthermore, the 8-unit grouping reflects the two- structure present in the piece.

The `base unit` algorithm has been successful with a variety of pieces, distinct in meter (4 3/4 or 6/8) and style (common practice, ragtime, afro-cuban). The algorithm explained far essentially produces a first draft of the metric hierarchy, to be refined by furt examination of the macro-periodicities. We have not yet extended the approach to explici spell out decisions about meter, but a major step is this direction has clearly been made.

For now, we use the chosen base unit for something rather mundane but quite useful. idea is to rearrange the tempo line into a more regular pattern, which reflects whene possible the desired groupings. Figure 13 is a trace of the rearrangement rules in action. idea is to rearrange structural anchors from left to right as follows. If the metric length bridge exceeds the base unit, the program tries to find a structural anchor at a dividing po that correspond to some multiple of the base unit. If several options are present, the b

match is used. The division process may be repeated recursively. It stops either when bridge length becomes less than or equal to the base unit, or when no good match is fo that is, when there is no note close enough to the target attack time. On the other h when the length of a bridge is strictly less than the base unit, and the next length is a multiple of the base unit, the rearrangement algorithm considers eliminating intermediate anchor. This is done if another anchor point can be found, at a whole numb base units ahead of the current position.

```
BRIDGE MODIFICATIONS

remove anchor at   1.056 to get a 4/1 unit bridge from   .000 to  3.840

attempting to divide 4/1 units between    .000 and   3.840

attempting to divide 5/1 units between   3.840 and   8.608
(    look for a break near  7.654 )
(   candidate at  7.584 ratio =  1.094 )

divide the 5/1 unit bridge between   3.840 and   8.608 at 4/1 units
 bridge from  3.840 to  7.584

    attempting to divide 4/1 units between   3.840 and   7.584
 bridge from  7.584 to  8.608

remove anchor at   8.608 to get a 4/1 unit bridge from  7.584 to 11.520

attempting to divide 4/1 units between   7.584 and  11.520

remove anchor at  17.120 to get a 4/1 unit bridge from 15.936 to 19.968

attempting to divide 4/1 units between  15.936 and  19.968

attempting to divide 5/1 units between  19.968 and  24.768
(    look for a break near 23.808 )
(   candidate at 23.680 ratio =  1.172 )

divide the 5/1 unit bridge between  19.968 and  24.768 at 4/1 units
 bridge from 19.968 to 23.680

    attempting to divide 4/1 units between  19.968 and  23.680
 bridge from 23.680 to 24.768

remove anchor at  24.768 to get a 4/1 unit bridge from 23.680 to 27.680

attempting to divide 4/1 units between  23.680 and  27.680

remove anchor at  29.824 to get a 4/1 unit bridge from 27.680 to 31.984

attempting to divide 4/1 units between  27.680 and  31.984
```

Figure 13: Rearrangement of Tempo Line

Many of the situations covered by the rearrangement algorithm do not actually come this example. Nevertheless, the result, shown in Figure 14, is worth considering.

TEMPO LINE

REAL TIME (SEC)	UNITS (TOTAL)	(ADDED)	UNIT LENGTH (SEC)	(MM)
.000	0/1	4/1	.960	63
3.840	4/1	4/1	.936	64
7.584	8/1	4/1	.984	61
11.520	12/1	2/1	.992	60
13.504	14/1	5/2	.973	62
15.936	33/2	4/1	1.008	60
19.968	41/2	4/1	.928	65
23.680	49/2	4/1	1.000	60
27.680	57/2	4/1	1.076	56
29.824	65/2			

Figure 14: Tempo Line (after modifications)

It is easy to see the error that was made earlier, in the evaluation of bridge lengths. The 5 bridge length is clearly wrong, and changing it to 2/1 would allows everything to fall place. In Figure 11, we might have noticed that the metric position of accents lost simplicity after the 5/2 bridge, but the nature of the error was not as obvious as it is now

How did the program make this mistake? We have to reexamine Figure 10a, at the po where the 5/2 length was decided upon (look for AT 13.504 in the printout). We see t the bridge length D is 2.432. With the local pulse Pu = 1.000, approximations for 2.432, cal PH earlier, are 5/2 and 2/1. Since 5/2 is much closer to 2.432 than 2/1, even though latter is simpler, the cost (combination of fit and simplicity) of 5/2 turns out to be lov than that of 2/1.

With the estimate based on previous bridges, things are rather worse, since the estimate I .979 is lower than Pu, making 2.432 look even larger. The only hypothesis returned by generator is 5/2 this time. Since this is a unique answer, and it agrees with the l hypothesis according to the other estimate, the strong agreement rule fires, and the progr chooses the 5/2 unit length for this bridge.

This is how the *error* happened, and it is difficult to see how it could have been avoided. the performance of the piece, the septuplet was played at a much slower tempo than the r of the piece, so slow in fact, that local criteria cannot make this appear as mere ten fluctuation. Changes to the generator or the production system that would make 2/1 oc ahead here would result in numerous errors in other contexts.

It thus seems adequate behavior for the program to first choose 5/2 on the basis of lc information, and to question that choice later using a more global perspective. This type reasoning, where feedback from higher levels is used to revise decisions made in a bottom

manner, seems necessary to achieve the desired robustness. We see here another instance control pattern used in many places: make some decisions in a bottom-up (or feed-forw manner, gather the results in a more global context, extract patterns at the higher le rearrange the lower level on the basis of the higher pattern, and reevaluate some of earlier decisions.

In fact, the reevaluation part of the program has not been implemented yet. This is bec the base unit algorithm was introduced quite recently, and the improvements that depen it have not been made yet. In the example, this means that the 5/2 bridge length is actu not corrected to 2/1. The tempo line corrections have done a useful task, but have not g as far as they could have gone. The program may now use the revised tempo line to add the assignment of metric values to individual notes.

2.8 Note Value Assignment

Once the tempo line has been determined, the problem of assigning rhythmic value individual notes becomes manageable, since it breaks down into independent problems, for each bridge.

The first step is to generate rational approximations for each note duration. We use estimates of the reference unit obtained from the revised tempo line.

Next, we consider bridges one at a time. If a particular bridge consists of, say, 5 notes, has a metric length of 3 units, the problem is to distribute the 3 units over the 5 note such a way that the cost of approximation of the notes is minimized. In general, involves a combinatorial search, since each note will usually have several poss approximations.

Before carrying out the combinatorial search, however, the program tries to take advan of the obvious cases. The least cost approximations for every note in a bridge are added and if the sum turns out to be equal to the number of units in the bridge, we have obtain cheap solution.

The note list in Figure 15 summarizes the state of affairs that results after this weak economical hypothesis generator has been tried. Let us examine the various fields displa

The Sdur column (Smoothed duration) is usually equal to the note duration (DUR). differences between Sdur and DUR occur within pulse trains, where Sdur is the average n duration in the pulse train average. Here, there is one pulse train in bar 4. The numbers blank in the Sdur column are all equal to the previous number, .173, which is the aver duration in the pulse train. This is part of the septuplet in the score.

In the Bu column (Bridge units) we find the metric length of each bridge, at the note wh begins the bridge. The next column (Lu, for local unit) gives the duration of the refere unit. Its value is printed at the beginning and end of the bridge, to indicate its scope. I equivalent to a local tempo setting.

The next column (Rdur for Relative duration) is the ratio Sdur/unit, that is, the (smooth note duration divided by the local reference unit. This is the target number for ratio

approximations. The rational approximations themselves (there is a variable number them) are found in the last field, approx. As with Sdur, numbers left blank indicate repetiti of the previous number (within a pulse train).

```
TITLE   Chopin Prelude (Opus 28, No15) ;

BARS    4/1     0/1 ;


KEY  Df major ;
SIGNATURE  Df: Bf Ef Af Df Gf  ;

PARS
    BEG   DUR  PIT TRS Sdur   Bu   Lu   Rdur  VAL  OK UVAL UMPOS approx;


ð BAR 1;
   .000  .800   F5  ~  .800  4/1  .960  .833   ?   ?  3/4    0/1 (5/6, 3/4, 7/8, 1/1, 2.
   .800  .256   D5  ~  .256   /   .     .267   ?   ?  1/4    3/4 (1/4, 1/3, 7/24, 1/2);
  1.056 1.824   A4  R 1.824   /   .    1.900   ?   ?  2/1    1/1 (2/1);
  2.880  .960   B4  ~  .960   /  .960 1.000   ?   ?  1/1    3/1 (1/1);
ð BAR 2;
  3.840 2.752   C5  R 2.752  4/1  .936 2.940  3/1  /  3/1    4/1 (3/1);
  6.592  .992   D5  ~  .992   /  .936 1.060  1/1  /  1/1    7/1 (1/1);
ð BAR 3;
  7.584  .768   E5  ~  .768  4/1  .984  .780  3/4  /  3/4    8/1 (3/4, 5/6, 2/3, 1/1, 1.
  8.352  .256   F5  ~  .256   /   .     .260  1/4  /  1/4   35/4 (1/4, 1/3, 7/24);
  8.608 1.920   G5  R 1.920   /   .    1.951  2/1  /  2/1    9/1 (2/1);
 10.528  .992   F5  ~  .992   /  .984 1.008  1/1  /  1/1   11/1 (1/1);
ð BAR 4;
 11.520 1.408   F5  R 1.408  2/1  .992 1.419   ?   ?  3/2   12/1 (3/2, 4/3, 2/1, 1/1);
 12.928  .576   E5  ~  .576   /  .992  .581   ?   ?  1/2   27/2 (7/12, 1/2, 2/3, 5/8);
 13.504 1.088 Eff5  R 1.088  5/2  .973 1.118   ?   ?  1/1   14/1 (1/1, 7/6, 5/4, 4/3);
 14.592  .224   E5  ~  .224   /   .     .230   ?   ?  1/7   15/1 (1/4, 5/24, 1/3);
 14.816  .192   F5  ~  .173   /   .     .178   ?   ?  1/7  106/7 (1/6, 5/24, 1/4, 1/3);
 15.008  .160   E5  ~   .     /   .      .     ?   ?  1/7  107/7 ( same );
 15.168  .160   D5  ~   .     /   .      .     ?   ?  1/7  108/7 ( same );
 15.328  .192   E5  ~   .     /   .      .     ?   ?  1/7  109/7 ( same );
 15.520  .160   F5  ~   .     /   .      .     ?   ?  1/7  110/7 ( same );
 15.680  .256   G5  ~  .256   /  .973  .263   ?   ?  1/7  111/7 (1/4, 1/3, 7/24, 1/2);
ð BAR 5;
 15.936  .864   F5  R  .864  4/1 1.008  .857   ?   ?  3/4   16/1 (7/8, 5/6, 1/1, 3/4, 2.
 16.800  .320   D5  ~  .320   /   .     .317   ?   ?  1/4   67/4 (1/3, 7/24, 1/2);
 17.120 1.824   A4  R 1.824   /   .    1.810   ?   ?  2/1   17/1 (2/1, 7/4, 5/3);
 18.944 1.024   B4  ~ 1.024   /  1.008 1.016   ?   ?  1/1   19/1 (1/1);
ð BAR 6;
 19.968 2.624   C5  R 2.624  4/1  .928 2.828   ?   ?  3/1   20/1 (3/1, 8/3);
 22.592 1.088   D5  ~ 1.088   /  .928 1.172   ?   ?  1/1   23/1 (7/6, 5/4, 1/1, 4/3, 3.
ð BAR 7;
 23.680  .800   E5  ~  .800  4/1 1.000  .800  3/4  /  3/4   24/1 (3/4, 5/6, 7/8, 2/3, 1.
 24.480  .288   F5  ~  .288   /   .     .288  1/4  /  1/4   99/4 (1/4, 1/3, 7/24, 1/2);
 24.768 1.888   G5  R 1.888   /   .    1.888  2/1  /  2/1   25/1 (2/1);
 26.656 1.024   F5  ~ 1.024   /  1.000 1.024  1/1  /  1/1   27/1 (1/1);
ð BAR 8;
 27.680 1.536   F5  R 1.536  4/1 1.076 1.428   ?   ?  3/2   28/1 (3/2, 4/3, 2/1);
 29.216  .608   E5  ~  .608   /   .     .565   ?   ?  1/2   59/2 (1/2, 7/12, 2/3, 5/8);
 29.824 2.160   D5  R 2.160   /   .    2.007  2/1  /  2/1   30/1 (2/1);

ð PARS
    BEG   DUR  PIT TRS Sdur   Bu   Lu   Rdur  VAL  OK UVAL UMPOS approx;
```

Figure 15: Note List with Values Based on Straight Sums

We recall that UVAL is an optional field taken directly from the input file, and that UMPOS obtained by cumulating the UVAL values, starting at 0/1. This allows the program to print t bar lines. Again, these are only for debugging purposes.

At this point, the VAL column contains mostly question marks, indicating that an hypothe was not found using the cheap method outlined above. When an hypothesis is found, that when the least cost approximations for the notes in a bridge add up to the bridge length, t metric values used are shown in the VAL column. We see that bars 2, 3 and 7, and part of b 8, fell in place that way. We can check these values against the user-provided values (t score) and see that they are correct.

One reason to carry out a cheap analysis before a full-blown search of all possi approximations is that we can use partial results in order to build contextual informati for the *peer pressure* strategy. The idea is to obtain statistics from the "obvious" solutic obtained using weaker methods, in order to bias the rational approximation genera towards finding more of the same values. Figure 16 shows the sort of statistics used he In these statistics, all the values that are the first choice of the rational approximati generator for some note receive a small weight, but the values retained in the VAL fi receive a high weight.

METRIC CONTEXT

VALUE	COUNT	COST
1/1	9	.100
2/1	8	.111
1/4	7	.125
1/6	5	.167
3/4	4	.200
3/1	3	.250
3/2	2	.333
5/6	1	.500
7/12	1	.500
7/8	1	.500
1/2	1	.500
7/6	1	.500
1/3	1	.500

Figure 16: Note Value Statistics

These statistics are used to alter the approximation generator's a priori idea of the simplic of fractions. The fractions that receive a high weight in the computed statistic tend become simpler than those with a low weight, even though the a priori criteria are r forgotten altogether. The rational approximation generator may then be invoked ane with the a posteriori, or context-driven simplicity criterion which now takes *peer press* into account. The cheap method is tried again. The results are shown in Figure 17, whicl to be compared with Figure 16. There are subtle differences in the approximatic generated, but the important fact is that bar 1 has now become *obvious*, as reflected in VAL fields. This is because the lowest cost approximation for the first note is now 3/4 inst of 5/6, due to peer pressure, allowing the sum of lowest cost approximations for this bric to come out right. The *peer pressure* statistics are then updated to reflect the new *islar of confidence*, for the benefit of the next step.

```
TITLE   Chopin Prelude (Opus 28, No15) ;

BARS    4/1     0/1 ;


KEY  Df major ;
SIGNATURE  Df: Bf Ef Af Df Gf  ;

PARS
    BEG   DUR  PIT TRS Sdur   Bu   Lu   Rdur  VAL  OK UVAL UMPOS approx;


ð BAR 1;
   .000  .800   F5  ~  .800  4/1  .960  .833  3/4  /  3/4   0/1 (3/4, 5/6, 1/1, 7/
   .800  .256   D5  ~  .256   /    .     .267  1/4  /  1/4   3/4 (1/4, 1/3, 7/24);
  1.056 1.824   A4  R 1.824   /    .    1.900 2/1  /  2/1   1/1 (2/1);
  2.880  .960   B4  ~  .960   /   .960 1.000  1/1  /  1/1   3/1 (1/1);
ð BAR 2;
  3.840 2.752   C5  R 2.752  4/1  .936 2.940  3/1  /  3/1   4/1 (3/1);
  6.592  .992   D5  ~  .992   /   .936 1.060  1/1  /  1/1   7/1 (1/1);
ð BAR 3;
  7.584  .768   E5  ~  .768  4/1  .984  .780  3/4  /  3/4   8/1 (3/4, 1/1);
  8.352  .256   F5  ~  .256   /    .     .260  1/4  /  1/4  35/4 (1/4, 1/3, 7/24);
  8.608 1.920   G5  R 1.920   /    .    1.951 2/1  /  2/1   9/1 (2/1);
 10.528  .992   F5  ~  .992   /   .984 1.008  1/1  /  1/1  11/1 (1/1);
ð BAR 4;
 11.520 1.408   F5  R 1.408  2/1  .992 1.419   ?   ?  3/2  12/1 (3/2);
 12.928  .576   E5  ~  .576   /   .992  .581   ?   ?  1/2  27/2 (7/12, 1/2, 3/4, 5
 13.504 1.088 Eff5  R 1.088  5/2  .973 1.118   ?   ?  1/1  14/1 (1/1, 7/6);
 14.592  .224   E5  ~  .224   /    .     .230  ?   ?  1/7  15/1 (1/4, 1/3, 5/24);
 14.816  .192   F5  ~  .173   /    .     .178  ?   ?  1/7 106/7 (1/6, 1/4);
 15.008  .160   E5  ~   .     /    .      .    ?   ?  1/7 107/7 ( same );
 15.168  .160   D5  ~   .     /    .      .    ?   ?  1/7 108/7 ( same );
 15.328  .192   E5  ~   .     /    .      .    ?   ?  1/7 109/7 ( same );
 15.520  .160   F5  ~   .     /    .      .    ?   ?  1/7 110/7 ( same );
 15.680  .256   G5  ~  .256   /   .973  .263   ?   ?  1/7 111/7 (1/4, 1/3, 7/24);
ð BAR 5;
 15.936  .864   F5  R  .864  4/1 1.008  .857   ?   ?  3/4  16/1 (1/1, 7/8, 3/4, 5/
 16.800  .320   D5  ~  .320   /    .     .317  ?   ?  1/4  67/4 (1/3, 1/4, 7/24);
 17.120 1.824   A4  R 1.824   /    .    1.810  ?   ?  2/1  17/1 (2/1);
 18.944 1.024   B4  ~ 1.024   /  1.008 1.016  ?   ?  1/1  19/1 (1/1);
ð BAR 6;
 19.968 2.624   C5  R 2.624  4/1  .928 2.828   ?   ?  3/1  20/1 (3/1);
 22.592 1.088   D5  ~ 1.088   /   .928 1.172   ?   ?  1/1  23/1 (7/6, 1/1);
ð BAR 7;
 23.680  .800   E5  ~  .800  4/1 1.000  .800  3/4  /  3/4  24/1 (3/4, 5/6, 1/1);
 24.480  .288   F5  ~  .288   /    .     .288  1/4  /  1/4  99/4 (1/4, 1/3, 7/24);
 24.768 1.888   G5  R 1.888   /    .    1.888  2/1  /  2/1  25/1 (2/1);
 26.656 1.024   F5  ~ 1.024   /  1.000 1.024  1/1  /  1/1  27/1 (1/1);
ð BAR 8;
 27.680 1.536   F5  R 1.536  4/1 1.076 1.428   ?   ?  3/2  28/1 (3/2);
 29.216  .608   E5  ~  .608   /    .     .565  ?   ?  1/2  59/2 (7/12, 1/2, 3/4);
 29.824 2.160   D5  R 2.160   /    .    2.007  2/1  /  2/1  30/1 (2/1);

ð PARS
    BEG   DUR  PIT TRS Sdur   Bu   Lu   Rdur  VAL  OK UVAL UMPOS approx;
```

Figure 17: Note List with Revised Values Based on Straight Sums

The next step attempts to finalize choices of metric values for the entire piece. Sinc have hypotheses for the metric duration of each bridge, the choice of note values is lo each bridge. For each note, we have a number of possible choices, which are the ra approximations formed earlier. The "cheap" method which paid attention only to the cost approximations obtained only a few answers. Now we need to consider all the ava approximations.

A branch and bound procedure is used to determine the optimal combination of hypoth note values inside the bridge. Partial sums of the available note value hypotheses rational approximations) are formed during a recursive search (from left to right i bridge). An answer is found when recursion hits the end of the bridge, with a partia equal to the metric length of the bridge. The answers obtained hypotheses whic evaluated according to the same two criteria used in the approximation generator, simp and degree of fit. Of course, the way these criteria are computed is quite different here.

Figure 18a gives a first example of the operation of the recursive search algorithm. bridge consists of the notes in bar 6. There are 4 approximations for the first note, 3 fc second, and 1 for the other two notes. Thus, the number of candidate partial su 4x3x1x1 = 12. Of these, only one gives the desired sum of 4 units, so this is the an retained.

```
SEARCH IN BRIDGE FROM  15.936 TO  19.968 (GOAL SUM: 4/1)

PARS
    BEG   DUR  PIT TRS Sdur   Bu   Lu   Rdur  VAL  OK UVAL UMPOS approx;

0 BAR 5;
 15.936  .864   F5  R  .864  4/1 1.008  .857   ?   ?  3/4   16/1 (1/1, 7/8, 3/4, 5/6);
 16.800  .320   D5  ~  .320   /   .     .317   ?   ?  1/4   67/4 (1/3, 1/4, 7/24);
 17.120 1.824   A4  R 1.824   /   .    1.810   ?   ?  2/1   17/1 (2/1);
 18.944 1.024   B4  ~ 1.024   /  1.008 1.016   ?   ?  1/1   19/1 (1/1);

SEARCH STARTED

 1/1   ...   ...   ...   Cutoff (large psum)
 7/8   ...   ...   ...   Cutoff (large psum)
 3/4   1/3   ...   ...   Cutoff (large psum)
  =    1/4   2/1   1/1   cd= 4 err= .449
  =    7/24  ...   ...   Cutoff (large psum)
 5/6   ...   ...   ...   Cutoff (large psum)

SOLUTIONS RETAINED
>>>  cd= 4 err= .449 cost= 1.048 3/4, 1/4, 2/1, 1/1
```

Figure 18a: A Simple Example of Recursive Search

In most cases, search paths are not explored all the way to the last note of the bridge. are cutoff rules which prune the search tree by interrupting the recursion. Whenever found in the figure, there has been such a cutoff, and an explanation of what caused it.

The cutoff rules are variants of the Q* algorithm. The first rule asks for a cutoff if tl partial sum of approximations formed thus far, plus the minimum partial sum we may g from the rest of the notes, exceed the target sum. This *large partial sum* rule is responsib for cutoffs on all search paths in Figure 18a, except of course on the path leading to tl solution. There is a *small partial sum* which is analogous to the previous rule, except th the inequality is reversed, and it uses the maximum partial sum obtainable for the rest of tl notes. The minimum and maximum needed for this are computed once and for all before tl search, in time linear with the number of notes in the bridge. Finally, there is a third cuto rule which involves a pre-computed lower bound on the remaining cost. The rule prescrib a cutoff if the costs so far, plus the minimum expected cost to completion (just mentione exceed the cost of the best solution so far.

```
SEARCH IN BRIDGE FROM  13.504 TO  15.936 (GOAL SUM: 5/2)

PARS
    BEG    DUR   PIT TRS Sdur    Bu    Lu    Rdur  VAL    OK    UVAL UMPOS approx;

 13.504 1.088    D5  R 1.088   5/2  .973 1.118   ?      ?     1/1   14/1 (1/1, 7/6);
 14.592  .224    E5  ~  .224   /    .      .230   ?      ?     1/7   15/1 (1/4, 1/3, 5/24);
 14.816  .192    F5  ~  .173   /    .      .178   ?      ?     1/7  106/7 (1/6, 1/4);
 15.008  .160    E5  ~  .      /    .      .      ?      ?     1/7  107/7 ( same );
 15.168  .160  Eff5  ~  .      /    .      .      ?      ?     1/7  108/7 ( same );
 15.328  .192    E5  ~  .      /    .      .      ?      ?     1/7  109/7 ( same );
 15.520  .160    F5  ~  .      /    .      .      ?      ?     1/7  110/7 ( same );
 15.680  .256    G5  ~  .256   /    .973   .263  ?      ?     1/7  111/7 (1/4, 1/3, 7/24);

SEARCH STARTED

 1/1    1/4    1/6    1/6    1/6    1/6    1/6    ...    Cutoff (small psum)
  ≡      ≡     1/4    1/4    1/4    ...    ...    ...    Cutoff (large psum)
  ≡     1/3    1/6    1/6    1/6    1/6    1/6    1/4    wrong psum
  ≡      ≡      ≡      ≡      ≡      ≡      ≡     1/3    cd=  6 err=  .674
  ≡      ≡      ≡      ≡      ≡      ≡      ≡     7/24   wrong psum
  ≡      ≡     1/4    1/4    ...    ...    ...    ...    Cutoff (large psum)
  ≡     5/24   1/6    1/6    1/6    1/6    ...    ...    Cutoff (small psum)
  ≡      ≡     1/4    1/4    1/4    ...    ...    ...    Cutoff (large psum)
 7/6    1/4    1/6    1/6    1/6    1/6    1/6    1/4    cd= 12 err=  .368
  ≡      ≡      ≡      ≡      ≡      ≡      ≡     1/3    wrong psum
  ≡      ≡      ≡      ≡      ≡      ≡      ≡     7/24   wrong psum
  ≡      ≡     1/4    ...    ...    ...    ...    ...    Cutoff (large psum)
  ≡     1/3    ...    ...    ...    ...    ...    ...    Cutoff (large psum)
  ≡     5/24   1/6    1/6    1/6    1/6    1/6    1/4    wrong psum
  ≡      ≡      ≡      ≡      ≡      ≡      ≡     1/3    wrong psum
  ≡      ≡      ≡      ≡      ≡      ≡      ≡     7/24   cd= 24 err=  .822
  ≡      ≡     1/4    ...    ...    ...    ...    ...    Cutoff (large psum)


SOLUTIONS RETAINED
>>>  cd= 12 err=  .368 cost= 1.300 7/6, 1/4, 1/6, 1/6, 1/6, 1/6, 1/6, 1/4
     cd=  6 err=  .674 cost= 1.389 1/1, 1/3, 1/6, 1/6, 1/6, 1/6, 1/6, 1/3
```

Figure 18b: Another Example of Search

Examples of the second cutoff rule are found in Figure 18b, but the third rule does not c into play in the current example. In Figure 18b, the bridge under consideration correspc to the latter part of bar 4, where the score contains a septuplet. Since the current progra not willing to consider denominators with prime factors other than 2 and 3, we expect s difficulties. The trace of the search shows that 3 solutions are found:

```
cd= 6 err= .674 solution= 1/1, 1/3, 1/6, 1/6, 1/6, 1/6, 1/6, 1/3

cd= 12 err= .368 solution= 7/6, 1/4, 1/6, 1/6, 1/6, 1/6, 1/6, 1/4

cd= 24 err= .822 solution= 7/6, 5/24, 1/6, 1/6, 1/6, 1/6, 1/6, 7/24
```

However, the summary after the trace of the search gives only two retained solutions. ' is due to selection by dominance relations: the third solution is dominated (by both o solutions). Here, we use the two criteria of fit to the data (err= ...) and complexity (cd= The error criterion for an answer is the sum of the error criteria of the individual fractio

Simplicity ratings are currently obtained from a rather naive notion of notati complexity. Suppose a candidate answer for a target sum of 1/1 consists of the fractions 1/4, 1/6, 1/6 and 1/6. The largest unit that evenly divides all these fractions is 1/12. ' unit, often called the density referent in musical analyses, may be mathematically defi using the common denominator of the given fractions, cd = 12. The simple comple measure based on the cd value is often adequate, but it does not capture the subtletie notation. The main problem is that it is independent of the order of the fractions. A be measure would consider 1/4, 1/6, 1/4, 1/6, 1/6 to be more complex than the prev example, which allows grouping as (1/4 + 1/4) + (1/6 + 1/6 + 1/6) and is thus easie notate. For a detailed analysis, though, one must take into account the position of current fragment in the global metric grid. An answer such as 1/4 + 1/2 + 1/4 is actu more complex than 1/6 + 1/2 + 1/3 if we are already 1/3 after the bar. This partly expl why we use the simpler measure: the more accurate one needs the precise knowledge of global metric position, which in turn depends on having obtained correct metric answe the left of the current position, a strong requirement that hinders the locality of decisions.

In the example of Figure 18b, we get common denominators of 1/12, 1/6 and 1 respectively. The use of dominance relations eliminates one of the answers, but we still h two answers (neither of which is very good). In order to make a decision, we combine two criteria into a single cost measure, obtaining:

```
cd= 12 err= .368 cost= 1.300 for 7/6, 1/4, 1/6, 1/6, 1/6, 1/6, 1/6, 1/4

cd = 6 err= .674 cost= 1.389 for 1/1, 1/3, 1/6, 1/6, 1/6, 1/6, 1/6, 1/3
```

The weighting scheme placed a high enough weight on the fit to the data to make musically absurd solution look better. Of course, the scheme had been trained using exam where there was a correct answer to be found, which is not the case here, due to septuplet.

A further cause of trouble at this spot, which was discussed in a previous section, is that length of the bridge should be 2/1, instead of the 5/2 length we are currently using.

Maybe the above discussion focused too heavily on a situation which appeared qu desperate (if only because of the septuplet, and the fact it was played far from evenly, as duration data show). However, this situation provided a way to bring many of the issues the surface.

If we consider the entire excerpt, however, the results are quite good. In the final note (see Figure 19) the only trouble spot is the septuplet (with the note immediately preced it). It seems quite certain that a trained music student who does not know the piece wo also have difficulties at this spot, even though he/she would probably find a better w around it that the program did.

All other notes in the piece received correct metric values. Furthermore, all pitches w correctly evaluated. The key of D flat major is correct, and a single error was made in naming of the notes. (The E double flat should have been called D natural, but the *n neighbor rule* responsible for this has not yet been implemented).

```
TITLE   Chopin Prelude (Opus 28, No15) ;

BARS    4/1     0/1 ;

KEY  Df major ;
SIGNATURE  Df: Bf Ef Af Df Gf  ;

PARS
    BEG    DUR   PIT TRS Sdur    Bu    Lu    Rdur  VAL  OK UVAL UMPOS approx;

∂ BAR 1;
    .000  .800    F5  ~  .800  4/1  .960  .833  3/4  /  3/4   0/1 (3/4, 5/6, 1/1, 7/8);
    .800  .256    D5  ~  .256   /    .     .267  1/4  /  1/4   3/4 (1/4, 1/3, 7/24);
   1.056 1.824    A4  R 1.824   /    .    1.900 2/1  /  2/1   1/1 (2/1);
   2.880  .960    B4  ~  .960   /   .960 1.000 1/1  /  1/1   3/1 (1/1);
∂ BAR 2;
   3.840 2.752    C5  R 2.752  4/1  .936 2.940  3/1  /  3/1   4/1 (3/1);
   6.592  .992    D5  ~  .992   /   .936 1.060  1/1  /  1/1   7/1 (1/1);
∂ BAR 3;
   7.584  .768    E5  ~  .768  4/1  .984  .780  3/4  /  3/4   8/1 (3/4, 1/1);
   8.352  .256    F5  ~  .256   /    .     .260  1/4  /  1/4  35/4 (1/4, 1/3, 7/24);
   8.608 1.920    G5  R 1.920   /    .    1.951 2/1  /  2/1   9/1 (2/1);
  10.528  .992    F5  ~  .992   /   .984 1.008  1/1  /  1/1  11/1 (1/1);
∂ BAR 4;
  11.520 1.408    F5  R 1.408  2/1  .992 1.419  3/2  /  3/2  12/1 (3/2);
  12.928  .576    E5  ~  .576   /   .992  .581  1/2  /  1/2  27/2 (7/12, 1/2, 3/4, 5/8, 2/
  13.504 1.088    D5  R 1.088  5/2  .973 1.118  7/6 7/6 1/1  14/1 (1/1, 7/6);
  14.592  .224    E5  ~  .224   /    .     .230  1/4 7/4 1/7  15/1 (1/4, 1/3, 5/24);
  14.816  .192    F5  ~  .173   /    .     .178  1/6 7/6 1/7 106/7 (1/6, 1/4);
  15.008  .160    E5  ~  .      /    .     .      1/6  /  1/7 107/7 ( same );
  15.168  .160 Eff5  ~  .      /    .     .      1/6  /  1/7 108/7 ( same );
  15.328  .192    E5  ~  .      /    .     .      1/6  /  1/7 109/7 ( same );
  15.520  .160    F5  ~  .      /    .     .      1/6  /  1/7 110/7 ( same );
  15.680  .256    G5  ~  .256   /   .973  .263  1/4 7/4 1/7 111/7 (1/4, 1/3, 7/24);
∂ BAR 5;
  15.936  .864    F5  R  .864  4/1 1.008  .857  3/4 1/1 3/4  16/1 (1/1, 7/8, 3/4, 5/6);
  16.800  .320    D5  ~  .320   /    .     .317  1/4  /  1/4  67/4 (1/3, 1/4, 7/24);
  17.120 1.824    A4  R 1.824   /    .    1.810 2/1  /  2/1  17/1 (2/1);
  18.944 1.024    B4  ~ 1.024   /  1.008 1.016  1/1  /  1/1  19/1 (1/1);
∂ BAR 6;
  19.968 2.624    C5  R 2.624  4/1  .928 2.828  3/1  /  3/1  20/1 (3/1);
  22.592 1.088    D5  ~ 1.088   /   .928 1.172  1/1  /  1/1  23/1 (7/6, 1/1);
∂ BAR 7;
  23.680  .800    E5  ~  .800  4/1 1.000  .800  3/4  /  3/4  24/1 (3/4, 5/6, 1/1);
  24.480  .288    F5  ~  .288   /    .     .288  1/4  /  1/4  99/4 (1/4, 1/3, 7/24);
  24.768 1.888    G5  R 1.888   /    .    1.888 2/1  /  2/1  25/1 (2/1);
  26.656 1.024    F5  ~ 1.024   /  1.000 1.024  1/1  /  1/1  27/1 (1/1);
∂ BAR 8;
  27.680 1.536    F5  R 1.536  4/1 1.076 1.428  3/2  /  3/2  28/1 (3/2);
  29.216  .608    E5  ~  .608   /    .     .565  1/2  /  1/2  59/2 (7/12, 1/2, 3/4);
  29.824 2.160    D5  R 2.160   /    .    2.007 2/1  /  2/1  30/1 (2/1);

∂ PARS
    BEG    DUR   PIT TRS Sdur    Bu    Lu    Rdur  VAL  OK UVAL UMPOS approx;
```

Figure 19: Final Note List

The program does not currently make a formal decision regarding meter. The previ discussion shows that we have essentially obtained the metric hierarchy, but we n additional rules for labeling particular levels in the hierarchy as being the preferred bar beat level. Methods to resolve classic ambiguities (such as 3/4 vs 6/8 meter) are availa but in some cases the choice is ultimately a matter of taste.

3. Conclusions and Future Research

This project has been concerned with combining signal processing and knowledge-bas systems to analyze digitized acoustic sound. We have developed a set of signal processi tools which includes implementations of standard techniques as well as new algorith specifically designed for the project. This tool kit is used by the acoustic analysis compone of our system to process the input signals and produce an event list. Each event is describ by an attack time and a set of other parameters, such as pitch, amplitude and sour identification. Knowledge-based techniques are used in the musical analysis component the system to produce an annotated score of the original sound.

Tests carried out with a variety of instruments and a variety of musical pieces, monophonic, have shown the system to perform quite well over a range of situations. intuitive description of the level of performance achieved is that the system mak relatively few errors, and more importantly, the errors made can be explained by t presence of a real difficulty in the piece, as opposed to some obscure quirk of processing. Th indicates that the system has the potential to handle reasonably well a wide variety transcription tasks.

Although much of the research has been directed to the particular domain of musi analysis, the task is basically a perception task. The overall structure of the system, as w as many of the methods grouped under the name of *control strategies*, deal with gene properties of perception systems rather than being specifically musical. The main property that context is critical in analyzing the signals at all levels of description. Speci knowledge about the domain is necessary to establish context, and use it, but many of t methods we have devised, or the conclusions we have reached, apply to a broader class problems.

In fact, we have confirmed our initial intuition that music is a particularly good domain experiment with the concept of intelligent signal processing. Signals have a high signal-t noise ratio, permitting a great detail of analysis; inputs of graded difficulty are easy generate, and we have a great degree of control over the various parameters, from t characteristics of the instruments to those of the performance, and the musical style; t domain knowledge has already received considerable attention, both at the acoustic level a at the various structural levels, and in some cases this knowledge has already been cast i form suitable for computer implementation.

One major limitation in the current system is that there is little interaction between t acoustic analysis and the musical analysis subsystems. Two examples of major improveme that can be expected from a more integrated system are a more precise analysis monophonic signals and the capability to process polyphonic material. Even w monophonic data and a high speed special-purpose signal processor, there are many attracti algorithms which cannot be used indiscriminately due to the computational cost. Furth this cost varies directly with such variables as the effective sampling rate, the number partials being processed, the time and frequency resolution required, etc. The selection of proper front-end algorithms and parameters to be used at each point in the data is important resource allocation problem. With polyphonic inputs this resource limitat becomes so critical that one needs elaborate methods for automatically allocating resourc In the current system, control strategies already carry out such a task within a sin subsystem, but it will be necessary to extend their influence across the entire system.

The current system's ability to process polyphonic signals is quite limited, and we do expect that the current *pipeline* front-end would successfully analyze polyphonic exam of moderate difficulty. Moorer's analysis program described in his thesis (1977) was cap of handling a duo of guitars, provided that the lower voice and the higher voice w always kept very clearly separated, and consonant intervals such as fifths and octaves w simultaneous attacks were avoided. Ultimately, the removal of such limitations se possible only in a system in which signal processing and musical context have an effect flexible mode of interaction.

Our current research indicates that the problems of source segregation and identificat will not be solved without the inclusion of techniques for uncovering patterns in various partial descriptions of the data manipulated by the system. Thus, future researc aimed at providing an extended framework for processing and representation (related those developed for speech and vision tasks) which will support an improved ability adapt to the patterns of any given example. We have already found useful conte generating patterns at different levels in music, giving us confidence in the power of approach, but there is much more to be done to systematize the control strategies and ap them effectively across the entire span of levels of description, including feedback f high-level pattern recognition processes into signal analysis.

The current work continues the research effort described in this report towards integration of a flexible and powerful framework for machine perception studies. Bey the pursuit of performance goals on the specific task of transcribing polyphonic music, are looking into ways of achieving a better separation of domain-dependent knowledge general resource allocation strategies.

4. Appendices

1. Chafe, C., B. Mont-Reynaud and L. Rush, *Toward an Intelligent Editor of Digital Aud Musical Construct Recognition*, Computer Music Journal 6:2 (30-41), March 1982.

2. Foster, S., J. Rockmore and W. Schloss, *Toward an Intelligent Editor of Digital Aud Signal Processing Methods*, Computer Music Journal 6:2 (42-51), March 1982.

3. Green, C., and J. Rockmore, *An Information-Theoretic Approach to Search Strategies Expert Systems*, unpublished technical report (1982)