Playing along with D'Accord Guitar

Giordano Cabral, Izabel Zanforlin, Rodrigo Lima, Hugo Santana and Geber Ramalho

Centro de Informática - Universidade Federal de Pernambuco
Caixa Postal 7851 - CEP 50732-970 - Recife (PE) - Brasil
{grec,izs,rqcl,hps,glr}@cin.ufpe.br

Abstract

This paper presents D'Accord Guitar, an innovative environment for learning, editing and performing music. D'Accord Guitar can be seen as an Instrumental Performance System (IPS). In order to improve musical notation completeness, providing equally an intuitive user interface, this kind of system shows the music performance directly on a virtual instrument displayed on the computer screen. Unfortunately, the current IPS exhibit various limitations: they are hardly editable, or not editable at all, they are not enough interactive and their interface is not fully adequate for string instruments. D'Accord Guitar combines various features of the current IPS and other musical tools in order to provide a user-friendly multi-function interface.

1. Introduction

Musical notations try to find the best trade-off between completeness and simplicity. The more complete is the notation, the more accurate it is. Rich notations are suitable for situations where the execution precision is essential. This is the case of classical western music, which is based on a score, a notation containing several performance details. On the other hand, the richer is the notation, the less readable it is. Amateur public usually demands simple and intuitive notations for making the learning process faster and easier. Trying to reach this public, notations that represent each music element separately (melody, lyrics, rhythm, harmony) have come out (Wet, Howel & Cross 1991). Using these notations, musicians can focus on each element independently, facilitating learning. Sometimes some of these elements are even omitted, supposing that the musicians already know them. For instance, jazz chord grids represent only harmony and real/fake books represent harmony plus melody (Sher 1991).

Multimedia resources, however, can improve the notation completeness without loosing too much simplicity (Roads 1996). An effort in this direction is the development of the so-called instrumental performance systems (IPS). These systems show directly the song performance on a virtual instrument displayed on the screen.

Unfortunately, the current IPS exhibit various limitations. First, they are hardly editable, or not editable at all, not allowing the users to write and play their own or favorite songs. Second, they are not enough interactive, considering the possibilities of teaching musical concepts. Third, the current IPS interface of the string instruments is not fully adequate, since problems such as chord positioning and fingering are not well addressed. Chord Dictionary and Automatic Accompaniment Systems overcome some of these IPS limitations. However, since they have other purposes, they do not fulfill IPS requirements.

This paper presents D'Accord Guitar, an innovative environment for learning, editing and performing music for guitar. D'Accord Guitar combines various features of the current Instrument Performance, Automatic Accompaniment, Chord Dictionary and Karaoke Systems in order to provide a user-friendly multi-function interface. D'Accord Guitar represents a two-year designing and programming effort, which required the solution of several interesting problems ranging from chord fingering, to chord positioning, through musical structures representation.

The purpose of this paper is to give an overview of D´Accord Guitar. A detailed specification of the problems of chord/solo positioning and fingering on a guitar neck is discussed elsewhere (Cabral et al. 2001a; Cabral et al. 2001b). Next section presents the state of art. Section 3 shows the guitar specific difficulties and constraints. Section 4 presents the principles, general architecture and enumerates the main difficulties in designing and implementing the system, sketching the proposed solutions. The final section draws some conclusions and presents directions for future work.

2. State of art

The most direct way of learning how to play a musical instrument is probably by observing a teacher playing it. Based on this hypothesis, some systems, generically classified as IPS, have been developed. The following sections analyze them and other related systems, explaining their benefits and limitations.

2.1.Instrumental Performance Systems

IPS simulate a human instrumental performance, showing it on a virtual instrument on the computer screen. They also offer some functions for changing the key and the tempo, for searching music on the Internet and for exhibiting extra information about the music (e.g., authors, publisher, etc.). Such systems are built to teach specific songs to people that already have some experience with the instrument, without caring about music theory or instrumental techniques. Examples of IPS are: *iSong*[1], *The Guitarist*[2] and *Desktop Guitarist*[3]. These systems show how multimedia resources can help musical learning process, improving notation accuracy, completeness and simplicity through a direct exhibition of the music performance on virtual instruments.

Unfortunately, various criticisms may be addressed to the current IPS. First, the *interactivity* is poor since they do not explore the huge possibility of teaching musical concepts, such as chord construction, recognition, ciphering and fingering (Birmingham & Pardo 2000). The user acts like a spectator, executing only macro commands, such as play, stop, rewind, fast forward, change key and change tempo. A higher level of interactivity can be found in chord dictionary systems (like Chord Wizard[4] and Violão Virtual[5]). However, they are not meant to show music performance on a virtual instrument.

Second, the IPS interface for *string instruments,* in particular guitar, does not show the information appropriately, complicating the interpretation of the song being played on the fretboard[6]. In fact, the current IPS stress the melodic (solo) information, hiding the underlying harmony (chords). For instance, Figure 1 shows a sequence of notes of an Am7 arpeggio as displayed on a guitar fretboard by a conventional IPS. Since neither the chord cipher is synchronically shown nor the position of all fingers involved in playing sequentially the chord

---

[1] Inside Music - http://www.isong.com
[2] PG Music - http://www.pgmusic.com
[3] Desktop Music - http://www.desktopmusic.com
[4] Chord Wizard - http://www.chordwizard.com
[5] Violão Virtual - http://www.violaovirtual.com
[6] String instrument neck — in opposition to keyboard

notes are exhibited, it is hard to recognize the harmony. This recognition task is almost impossible when various arpeggios are chained and played fast (Holdsworth 1998).

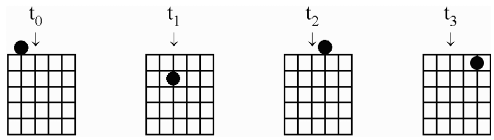

Figure 1 - Arpeggio of a Am7 in a conventional IPS

Third, in all current IPS, the music performance is not *editable*, not allowing users to create or edit their own or favorite songs. The exceptions are Desktop Music's systems, which are editable but are impractical since the user is obliged to specify every note to be played.

Fourth, several IPS do not include the songs' *lyrics*. Even in the cases where the lyrics are shown, they are not temporally synchronized with the chord ciphers and the melody. As observed in conventional songbooks (Chediak 1999), the fact of displaying lyrics, chord ciphers and melody synchronically, can in principle help users to learn to play a song.

2.2. Automatic Accompaniment Systems

Automatic Accompaniment Systems (AAS), such as Band-In-A-Box[7] and Harmony Assistant[8], are systems that automatically generate melodic lines (e.g., saxophone and bass), rhythmic lines (e.g., drums) and/or chord voicing chaining (e.g., piano) in a particular music style according to a given chord grid (Ramalho, Rolland & Ganascia 1999). Since these systems may include an "IPS module" in order to exhibit the generated music on virtual instruments (usually a keyboard), they partially address some of the discussed problems. For instance, they display the guitar chord ciphers synchronically, facilitating interpretation of notes played on the fretboard. The user can edit song ciphers on a chord grid and associate melody and lyrics to it.

However, since the goal of AAS is basically to generate MIDI sequences (Moog 1986) to be used by composers, arrangers and musicians in general, they do not care about performance details. For instance, there is no indication of the fingering of a chord on a guitar neck. In most systems, there is no indication of the possible positions of a chord neither. When this indication exists, the number of such positions is minimal (typically less than 5). These limitations inhibit a suitable use of these systems for string instruments, despite their popularity. Another problem found is the AAS weak flexibility, since, in the majority of these systems, it is impossible to edit the rhythm and/or the chord positions. The user must accept the generated accompaniment without being able to adjust it.

3. Problems of modeling guitar performance

It is difficult to develop IPS for guitar due to its intrinsic ambiguities and spatial constraints (Cabral at al 2001a; Cabral at al 2001b). The first fundamental problem in guitar modeling is that the same pitch may be generated using different guitar positions[9]. The second fundamental problem is that there are a maximal number of strings to play (six) and maximal number of fingers to use (four). On one hand, these spatial and anatomic constraints help to

[7] PG Music - http://www.pgmusic.com
[8] Myriad Software - http://www.myriad-online.com
[9] In this paper *position* means a single tuple <string, fret> whereas c*hord position* means a group of positions that composes a chord (i.e., a position for each chord note).

reduce the number of possible positions of a chord, by forcing the elimination or repetition of notes. On the other hand, these constraints are difficult to be modeled.

For these reasons, the transcription of a music conventional notation, such as a score, into the actual performance is not straightforward. This transcription problem is even more complicated when converting chord ciphers into actual fretboard positions, since ciphers introduce more ambiguity (i.e., they can be mapped into different chord positions).

Concerning solos transcriptions, the task of converting a notation into an actual guitar performance can be satisfactorily solved by some algorithms (Sayegh 1989). However, to our knowledge, there is no available solution for the task of finding the best guitar performance based on a sequence of chord ciphers (Fowler 1984b). Even for a keyboard, there is more than one way of playing a chord. In other words, there is more than one voicing (Fowler 1984a). Some notes may be doubled, some notes may be omitted, and chords have different inversions. In the guitar case the problem is much harder. To find the best chord position, the system must first know all possible chord positions, in where a chord position can be seen as an ensemble of fretboard positions and the right-hand and left-hand fingers used. Finding these possible chord positions is, per se, a problem, since there are musical, anatomic and stylish constraints to be respected. Finding the best chord positions is the hardest problem, involving different and often-contradictory parameters, as will be seen in section 4.4. Such section briefly discuss a solution for this problem, called *best positioning and fingering in chord chaining*.

4. D'Accord Guitar

This section discuss the main design choices made when developing D'Accord Guitar, as well as the proposed solutions for some of the previously discussed problems. However, a detailed explanation about these solutions is not in the scope of this paper.

4.1.Principles

The basic assumption in D'Accord Guitar is that a proper separation of melody, harmony, rhythm and lyrics information improves user learning, letting him/her focuses on a particular element at once. For example, a chord accompaniment can be seen not just as an ensemble of notes sequentially played, but also as a rhythm being applied to a sequence of chords, generating such notes. Instead of learning a sequence of notes, the user could then focus only on a chord sequence associated with a known rhythmic pattern.

This separation is also of great help in the song edition process, since it will be possible to edit each element separately, omit some elements and generate some elements automatically. For instance, the user can specify the chord grid, import the melody from a MIDI file, and specify a rhythmic pattern to be applied on the chords. In particular, the explicit representation of the underlying harmony can solve the problem illustrated in Figure 1. In fact, knowing the chord, it is simple to show the position of all fingers involved in playing it. This separation can be a basis for an adequate approach of problems such as chord positioning, fingering and chaining. For instance, when playing an arpeggio of a chord, D'Accord Guitar does not show an animation note by note, as a traditional IPS do. The position of every finger is indicated in the fretboard during the chord time span. At the moment that a note is played, the correspondent fingers (of the left and right hands) are highlighted, as illustrated in Figure 2.

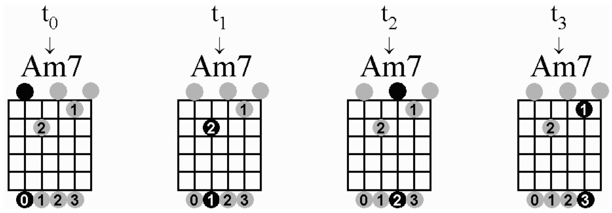

Figure 2 - Arpeggio of an Am7 in D'Accord Guitar. Circles at the top indicate left-hand fingers. Circles at the bottom indicate right-hand fingers. Black circles indicate the played string. Gray circles complete the chord position. Strings without circles are unavailable.

4.2. User interface

Regarding the problems of interactivity, information displaying, editing capabilities, and flexibility discussed in Sections 2.1, D'Accord Guitar proposes some solutions and/or improvements.

As all the IPS, the didactic concern of D'Accord Guitar is reflected by some controls, like loop and tempo change, and by a visible fretboard. However, the interactivity provided by our system goes beyond these simple features. In D'Accord Guitar, there are three different user interaction modes, for executing a song, editing a song and browsing chords/rhythms.

In the *executing mode*, the user can perform executing (playing) commands over the song (such as play, stop, pause, loop, fast forward, rewind, change tempo and change song position). Figure 3 shows the main interface of D'Accord Guitar's in such a mode. In the central part of the window, there is the fretboard indicating the left and right fingers positions from the guitar player standpoint (guitar's nut on the left, 1st string up, 6th string down). When a string is played, the correspondent fingers pictures are highlighted (in Figure 3, fingers 1, 2 and 3 are playing simultaneously). The bottom of the window exhibits song's ciphered chords and lyrics, animated as in Karaoke systems. Users can also transpose the song, see other positions of a given chord and toggle between the harmony exhibition mode (where chords and all fingers involved in playing them are shown) and the solo one (where the song melody is played on the fretboard as a solo).

Figure 3 - D'Accord Guitar Interface in execution mode

In the *edition mode*, the user can create a song from scratch or edit an existent one, writing the chord grid, lyrics and recording each song element separately. The melody can be recorded via a MIDI instrument or obtained via a MIDI file. The harmony is obtained from a chord grid (like in Band-In-A-Box), with the flexibility of changing the chord position according to the user demand (e.g., in the case of Figure 3, the user may prefer a Dm7 in the fifth fret). The rhythm can be recorded from a MIDI instrument or can be chosen from the rhythm base. Since each element is recorded separately, the system implements studio recording mechanisms of overdubbing, punch in and punch off (Keating & Anderton, 1998), so that a new element can be recorded in real time while the other already stored elements are played synchronically.

In the *browsing mode*, the user can interact directly with the fretboard and browse the chord and rhythm databases. The user can stop the system performance in order to browse musical concepts (typically chords) by interacting directly with the fretboard. For example, while the user chooses chord notes in the fretboard, the program is able to show the notes names, the chord cipher, the intervals between the notes and the chord root, and the fingers that must be used to play these notes. The user can also navigate through chord and rhythm databases, as in a chord dictionary system.

4.3.Architecture

Figure 4 shows D'Accord Guitar general architecture. Thick lines indicate the execution mode interactions, dotted lines show the edition mode interactions and thin lines indicate the browser mode interactions.

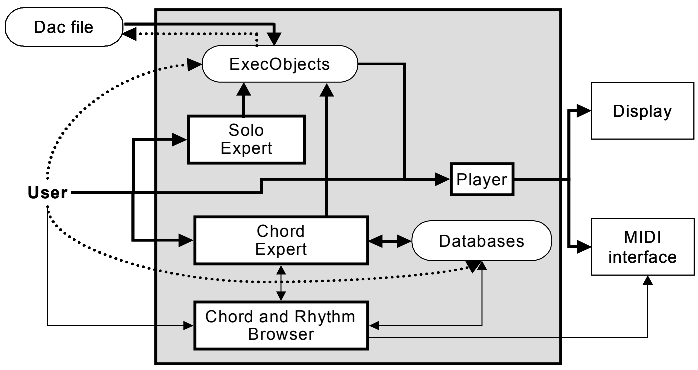

Figure 4 - D'Accord Architecture

D´Accord Guitar uses a particular musical representation, rich enough to describe appropriately the song. The idea is to include information about harmony, melody, rhythm, lyrics and solo, providing different granularity levels of representation. This allows simultaneously the specifications of detailed information (e.g. solos, specific chord positions) and generic one (e.g. ciphers and rhythm patterns). For instance, in editing a song, the user can only write the ciphers, letting the system to choose the positions and the rhythmic patterns. Alternatively, the user can choose particular chord positions and record rhythmic patterns for some song's passages or for the entire song.

The song is modeled as synchronized events. There are harmonic, melodic, rhythmic, lyrics, solo and control events, each one with its own attributes. At runtime, these events can be seen as executable musical objects (the `ExecObjects` in Figure 4). They are synchronized by the `Player`, which shows the performance on the virtual instrument on the display, and plays the notes via MIDI interface. D'Accord Guitar has a proprietary file format to store or load these events, shown in Figure 4 as the `Dac File`.

There are two `databases` (see Figure 4), one for chord positions and other for rhythms. They are used both to improve the system performance and to store information that cannot be automatically computed (e.g. some cultural-grounded judgements such as whether a given chord position is usual). The `Browser` module allows the user to browse the chord and rhythm databases and to interact directly with the fretboard.

The `Chord Expert` and `Solo Expert` modules are designed to solve problems such as right and left hand fingering, chord voicing and positioning. The `Solo Expert` deals with transcribing and playing solos, and uses existent algorithms (Sayegh, 1989; Cordier, 1995). In contrast, the `Chord Expert` presents innovative solutions, which will be briefly discussed in the next section.

4.4. The Chord Expert

The Chord Expert is responsible for recognizing and playing chords. As explained previously, the main problem in D´Accord Guitar is to find the best chord positions based on a sequence of chord ciphers. This section will briefly introduce the solution we are proposing to this problem in D´Accord Guitar.

Positioning and Fingering

Before finding the best chord position, the Chord Expert find the possible positions of each chord. D´accord Guitar enables positions ranging from fret 0 to fret 12. Since a whole octave is covered, for each string, all notes of a chord can be played. Given a chord composite of n notes, there are $6n$ fretboard positions to use. D´Accord Guitar performs an exhaustive search on these fretboard positions, filtering the results according to voicing and fingering constraints. The voicing constraints ensure that the chord position actually corresponds to the chord. For instance, Figure 5 shows six possible Fm7(11) positions. The fingering constraints guarantee that the chord position is executable. For instance, the chord position must not need more than 4 fingers to be placed and the maximum distance between fingers must not exceed 4 frets. Thus, the positions shown in Figure 5d, Figure 5e and Figure 5f are disqualified by fingering constraints.

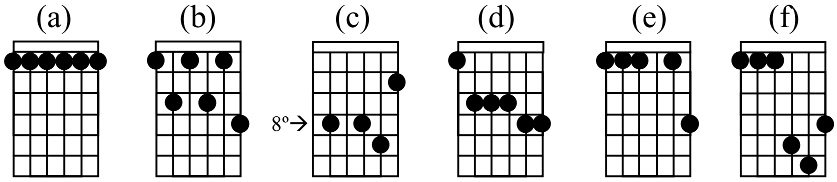

Figure 5 – Some possible Fm7(11) positions

For each chord position generated, D´accord Guitar calculates the possible left-hand fingerings, including the possibility of using a bar chord. This is a constraint satisfaction problem, whose solution is based on simple heuristics. For instance, if there are two strings pressed on different frets, the note placed on the lower fret must be played by the finger with the lower number. If there are two strings pressed on the same fret, the not of the upper string must be played by the lower finger. In Figure 6 the finger 1 is placed on the 2nd string and the finger 2 is placed on the 4th string by the first rule. The finger 3 is placed on the 5th string and the finger 4 is placed on the 3rd string by the second rule.

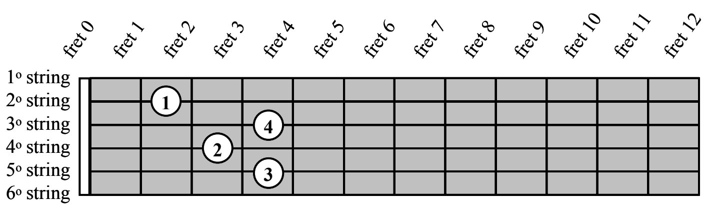

Figure 6 – A possible Db7 position in the fretboard

Best positioning and fingering in chord chaining

After generating all possible chord positions, it is necessary to evaluate them, taking into account the other chords. D´Accord Guitar uses a hybrid model to solve this problem, joining a heuristic search (Rayward, Osman & Reeves 1996) with Case-Based Reasoning (Kolodner 1993).

The search is based on a multi-attribute function using different and often-contradictory parameters that can be weight up by the user. There are context-free and context-dependent parameters. Context-free parameters refer to questions like: "how usual is the chord position/fingering?" and "how easy is to play the chord position/fingering?". Context-dependent parameters refer to questions like: "what are the chord positions where there is the smoothest bass line?".

The Case-Based Reasoning measures the similarity between chord positions, and is also useful to transpose a song, trying to keep the chord positions as similar as possible with the original ones.

5. Results

The *search of all possible position and fingerings*, and the *best positioning and fingering without chord chaining*[10] are already implemented and tested. The *best positioning and fingering in chord chaining*[11] feature is still being validated and refined.

To illustrate the current state of the *best positioning and fingering in chord chaining,* we performed a simple test. Given the chord sequence: Em / C7M / D7 / G, the results when usual chords are preferred (Figure 7) are different from those where a "smooth" bass line (possibly with unusual chord positions) is preferred (Figure 8).

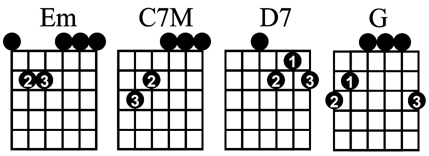

Figure 7– Most usual positions and fingering for the given chord grid

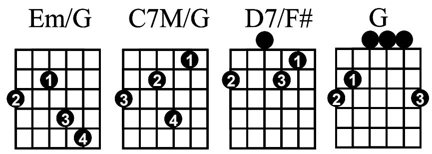

Figure 8 – Chord positions with the closest bass notes

These results are not conclusive yet, since we are assessing just one of the evaluation parameters at each time. However, the first results are encouraging, indicating that this approach can lead us to a complete and flexible solution to the presented problem.

6. Conclusion

In this paper, we have described the main features of D´Accord Guitar, a multi-featured tool, useful to improve the musical learning, editing and reading processes. D'Accord Guitar can be seen as a guitar IPS that incorporates features from Automatic Accompaniment Systems, Karaoke Systems and Chord Dictionary Systems to provide a more complete solution. It has an underlying representation of guitar musical structures. Based on this representation, some editing functions existing in Automatic Accompaniment Systems have been adapted and integrated. Beyond the proposed representation and some editing capabilities, D'Accord Guitar environment provides special mechanisms for helping amateurs

[10] Best positioning and fingering without chord chaining is a search using only context-free parameters
[11] Best positioning and fingering in chord chaining is a search using only context-free parameters

to learn musical concepts. These learning mechanisms and the advanced editing functions qualify this system to be used by both beginners and skilled guitar players.

Our future works include the validation of the solutions proposed to some problems, such as transposition and fingering. We also intend to improve, and add some new, solutions to simplify the edition of songs. For instance, we are now improving the module that automatically transcribes solos and chord sequences played in a MIDI guitar. Finally, we plan to adapt this system to other different string instruments, since their underlying representation structures are probably similar.

7. References

Birmingham, W. & Pardo, B. (2000) On The Computational Properties of Harmonic Analysis. In the *Proceedings of the Workshop on Artificial Intelligence and Music*. AAAI'2000.

Cabral, G., Santana, I., Lima, R., Santana, H. & Ramalho, G. (2001a) D'Accord Guitar: an Innovative Guitar Performance System. In Proceedings of *VIII Journées Françaises d'Informatique Musicale (JIM '2001)*, Bourges (forthcoming).

Cabral, G., Zanforlin, I., Lima, R., Santana, H. & Ramalho, G. (2001b) Da Cifra para o Braço: Estudo dos Problemas de Execução Musical em Violão e Guitarra. In Proceedings of *VIII Brazilian Symposium on Computer Music* (discussion paper). Fortaleza (forthcoming).

Chediak, A. (1999). *Songbook of Chico Buarque* (vol 1-4). Rio de Janeiro: Lumiar Ed.

Cordier, M.-O. (October, 1995) Doigtage intelligent d'une partition de guitare. N° 23, 46-48. Bulletin de L´AFIA.

Fowler, W. (1984a) *Chord Voicing Systems*. Fowler Music Enterprises.

Fowler, W. (1984b) *Chord Progression Systems*. Fowler Music Enterprises.

Holdsworth, A. (1998) *Melody Chords for Guitar*. New York: Hal Leonard.

Keating, C. & Anderton, C. (1998) *Digital Home Recording - Tips, Techniques, and Tools for Home Studio Production*. Backbeat Books.

Kolodner, J. (1993) *Case-Based Reasoning*. San Mateo: Morgan Kaufmann.

Moog, B. (1986) MIDI: Musical Instrument Digital Interface. *Journal of the Audio Engineering Society*, 34(5), 394-404.

Pachet, F., Ramalho, G., & Carrive, J. (1996). Representing Temporal Musical Objects and Reasoning in the MusES System. *Journal of New Music Research,* 25(3), 253-73. Swets & Zeitlinger.

Ramalho, L., Rolland, P-Y., Ganascia, J-G. (1999). An Artificially Intelligent Jazz Performer. *Journal of New Music Research,* 28(2), 105-129*. Swets & Zeitlinger.*

Rayward-Smith, V., Osman, L & Reeves, R. (1996) *Modern Heuristic Search Methods.* John Wiley & Son.

Roads, C. (1996) *The Computer Music Tutorial*. Massachusetts: MIT Press.

Sayegh, S. (1989) Fingering for String Instruments with the Optimum Path Paradigm. *Computer Music Journal*, 13(3).

Sher, C. (1991) *The New Real Book* (vol. 1 and 2). Berkeley: Sher Music.

West, R., Howell, P., & Cross, I. (1991). Musical Structure and Knowledge Representation. In P. Howell, R. West, & I. Cross (Eds.), *Representing Musical Structure* (pp. 1-30). London: Academic Press.