D'Accord Guitar: an Innovative Guitar Performance System

Giordano Cabral, Izabel Santana, 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

Computers multimedia resources may be used to improve musical notation completeness, providing equally an intuitive user interface. An effort in this direction is the development of the so-called instrumental performance systems (IPS), which show the performance directly on a virtual instrument displayed on the computer screen. Unfortunately, the current IPS exhibit various limitations. For instance, they are hardly editable, or not editable at all, they are not enough interactive and their interface is not fully adequate for string instruments. 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 IPS and other musical tools in order to provide a user-friendly multi-function interface.

Keywords: Musical editing and publishing systems, musical performance modeling and simulation, formalization and representation of musical structures.

1. Introduction

Musical notations try to find the best trade-off between richness 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. In certain music styles, such as jazz, pop, bossa nova, rock'n roll, blues, 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 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 2-year designing and programming effort, which required the solution of several interesting problems ranging from chord fingering, to chord positioning, through musical structures representation.

However, the purpose of this paper is to give an overview of D'Accord Guitar. A detailed specification is out of the scope of this paper. Next section presents some current tools and their limitations. 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. Figure 1 shows the main interface of The Pianist.

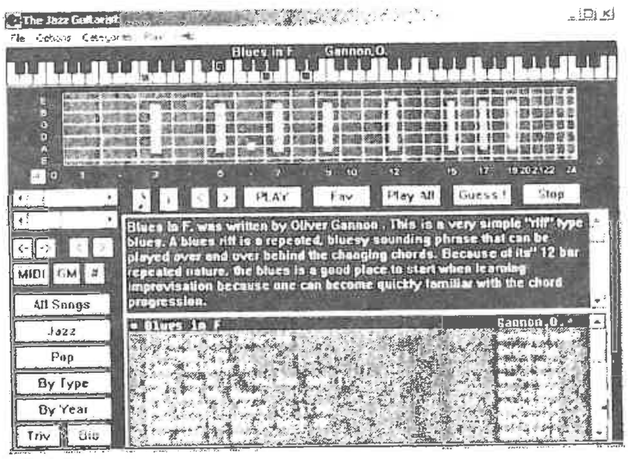

Figure 1 - PG Music The Guitarist

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,

---

[1] Inside Music - http://www.isong.com
[2] PG Music - http://www.pgmusic.com
[3] Desktop Music - http://www.desktopmusic.com

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 fretboard[6]. In fact, the current IPS stress the melodic (solo) information, hiding the underlying harmony (chords). For instance, Figure 2 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 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).

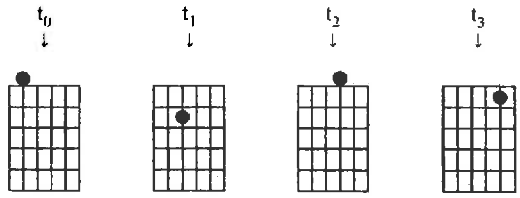

Figure 2 - Arpeggio of a Am7 in a conventional IPS

Third, in all current IPS, the music performance is not *editable*, not allowing users to create or edit their own or favorite songs. The exception is 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). Such systems may include an "IPS module", to exhibit the generated music on virtual instruments (usually a keyboard), and partially address some of the discussed problems. For instance, they display the guitar chords 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 their goal is basically to generate MIDI (Moog 1986) sequences 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 possible positions of a chord and its fingering on a guitar neck. This limitation inhibits 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

[4] Chord Wizard - http://www.chordwizard.com
[5] Violão Virtual - http://www.violaovirtual.com
[6] String instrument neck — in opposition to keyboard
[7] PG Music - http://www.pgmusic.com
[8] Myriad Software - http://www.myriad-online.com

positions. The user must accept the generated accompaniment without being able to perform adjustments.

3. Problems of modeling guitar performance

It is difficult to develop IPS for string instruments, due to its intrinsic ambiguities and spatial constraints. These difficulties are often found in guitar, used as the basis of this study. However, the solutions proposed can be extended for others string instruments, since most of the basic concepts are the same. The first fundamental problem in guitar modeling is that the same pitch may be generated using different combinations of positions[9]. For instance, the 3 positions shown in Figure 3 produces the same pitch.

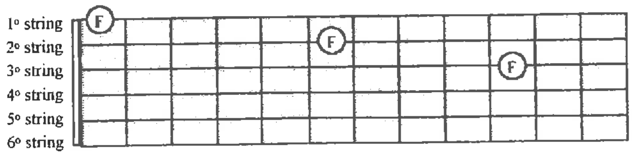

Figure 3 – 3 different positions for the same pitch

For this reason, a transcription from the conventional notations, such as scores, to 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.

The second fundamental problem is that there are a maximal number of strings to play and fingers to use. These spatial and anatomic constraints help to reduce the number of possible positions of a chord, by forcing the elimination or repetition of notes. However, they are difficult to be modeled.

3.1. Transcribing and playing solos

The conversion from selected positions in the fretboard into notes in a melodic notation is not considered a problem, since each of these positions have a unique correspondent note. However, the task of converting a melodic notation into an actual guitar performance is considered a problem. Fortunately, some algorithms (Sayegh 1989) may be used to satisfactorily solve this problem.

3.2. Recognizing and playing chords

If for single notes there are conversion difficulties, for chords there are much more. Even for a keyboard, there are more than one way of playing a chord: some notes may be doubled, some notes may be omitted, chords have different inversions. Such a problem is called voicing (Fowler 1984a). In the case of the guitar the problem is much harder, since there are also different positions to the same voicing.

In fact, there are two problems. One problem is "how to recognize a chord cipher based on an ensemble of notes selected in the fretboard?". In spite of the fact that there are still open cases (e.g. when to recognize a chord as a Em7(b5) or as a Gm6?), this problem is already well explored, as can be seen on several studies (e.g. Pachet, Ramalho & Carrive 1996).

On the other hand, the task of finding the best guitar performance based on a sequence of chord ciphers is much harder (Fowler 1984b). This is difficult because there is not a single

[9] In this paper, the word *position* indicates a fretboard position (a tuple <string, fret>). A *chord position*, however, indicates a set of positions that composes a chord.

possible best conversion, since there are multiple parameters that can be taken to choose the best chord positions. To a better understanding of this problem, it will be divided in subtasks. It is necessary to know all possible positions for a given chord and to evaluate them. To explore all possible chord positions for a given chord it is necessary to find out not only the frets and strings used, but also the fingers used. So, there are two distinct problems. The first is called positioning, and consists in finding which are the frets and strings used in each chord position. The second is called fingering and consists in find out what finger is used in each pressed string (Cordier 1995). In the following sections these problems will be described more deeply.

Positioning

The complexity involved in the positioning problem can be observed analyzing it combinatorially. Assuming a system that allows positions from fret 0 to fret 12, each string can be used to play any note of a chord (since each string covers a whole octave). Given a chord composite of *n* notes, there are at least 6*n* fretboard positions that may be used to form such a chord position. In Figure 4 can be seen the possible positions for a Fm7(11).

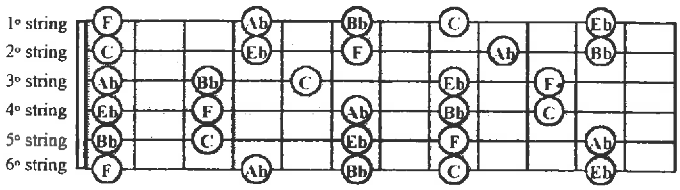

Figure 4- Possible positions for a Fm7(11)

Using all 6 strings, there are n^6 combinations of such notes (that may eventually compose the chord). Using only 5 strings there are $6n^5$ combinations of positions, and so on. This shows the large amount of data to deal with. For the Fm7(11) instance, there are a total of 43750 combinations of positions that may compose a chord. Such combinations must be filtered according to voicing and anatomic constraints.

The voicing constraint filters which of these combinations really compose the desired chord. In the previous case, only 5880 of the 43750 are in fact considered Fm7(11). Figure 5 shows some of these Fm7(11) positions, each one with a different voicing.

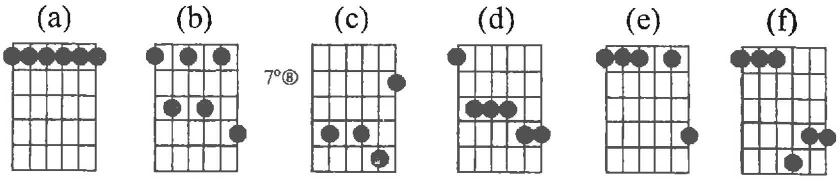

Figure 5 - Some positions of Fm7(11) in a guitar fretboard. 6th string on the left, 1st string on the right. Circles indicate the positions (fret 0 up). Strings without circles are unavailable.

Fingering

The complexity involved in the fingering problem is due to a difficult computational modeling of the anatomy of human hands. This modeling is necessary to find which fingers to use in each pressed string, and, consequently, to filter positions impossible to perform. This filtering is based on left-hand and right-hand constraints. The right-hand filter assumes that the player always plays his/her 3 fingers in consecutive strings, avoiding positions such as (e) of Figure 5. The left-hand filter avoids chord positions where there is a big distance between fingers (such as (f) of Figure 5) or that needs more than 4 fingers to be played (such as (d) of

Figure 5). In the Fm7(11) case, the right-hand filter eliminates 1296 of the 5880 chord positions. After the left-hand filter execution, only 283 chord positions remains.

After finding the positions, it is necessary to search the possible left-hand fingers to put at each one, including the possibility of using a bar chord. The amount of possibilities involved in fingering is extremely smaller, but the modeling is even harder. To demonstrate it, lets use the basic position of Dm7, shown in Figure 6. In such a case, there are 24 combinations of fingers that do not use a bar chord, plus 3 combinations of fingers that uses it. Figure 6 shows some of them.

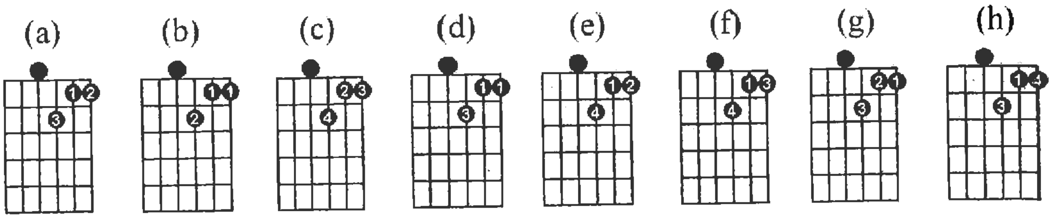

Figure 6 – Some possible fingerings of the Dm7 basic position. Numbers indicates the fingers used. Circles without number indicates a not pressed string. Strings without circles are unavailable.

This figure shows the difficulty of automatically finding values to its attributes. The Figure 6a and Figure 6b fingerings are the most commonly used (the first one does not use a bar chord while the second one does). The Figure 6c, Figure 6d and Figure 6e fingerings are less used, but their use is justifiable depending on next chord positions and guitar player styles. For instance, if next chord is the Em7 shown in Figure 7, Figure 6c fingering is justifiable.

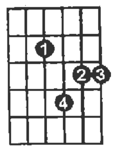

Figure 7 – Fingering for a Em7 position

Additionally, Figure 6f fingering is unjustifiable, Figure 6g fingering is unused and Figure 6h fingering is impossible to perform. Concluding, for the 27 possible fingerings of the Dm7 position shown in Figure 6, 7 are executable, but only 2 are used in most cases. To an IPS, the problem is how to find these 7 executable fingerings and how to evaluate them to choose the best one depending on the context.

Best positioning and fingering for chord chaining

After generating all possible chord positions and their possible fingerings, it is necessary to find and weight up parameters to evaluate them. Some of these parameters are context independent, such as comfort, commonness and flexibility, but it is hard to measure them in a deterministic way.

Nevertheless, this context free evaluation is not reasonable. The final objective is to play a chord grid. Then, the system must take into account all chords involved. Then, it is necessary to consider context dependents parameters, such as proximity of the bass note, similarity of chord positions, etc. The searching of these best positions and fingerings based on a chord grid is the hardest problem, and is called in this paper "*best positioning and fingering in chord chaining*".

To our knowledge, there is no available solution for this problem. The main innovation of D'Accord Guitar is to have developed a solution for it.

4. D'Accord Guitar

This section discuss the main design choices made in 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 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 2. 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 8.

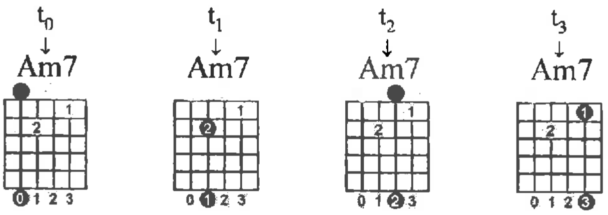

Figure 8 - Arpeggio of an Am7 in D'Accord Guitar. Circles at the top indicate left-hand fingers. Circles at the bottom indicate right-hand fingers. Black circles indicate the played string. Gray circles indicate finger that are not playing, but that are positioned. Strings without circles are unavailable.

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 9 shows the main interface of D'Accord Guitar's in such a mode. In the central part of the window, there is the virtual guitar neck 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 9, 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, such as WinOKE[10] and RealOrche[11]. D'Accord Guitar has generic functions for file manipulation and MIDI configuration. 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).

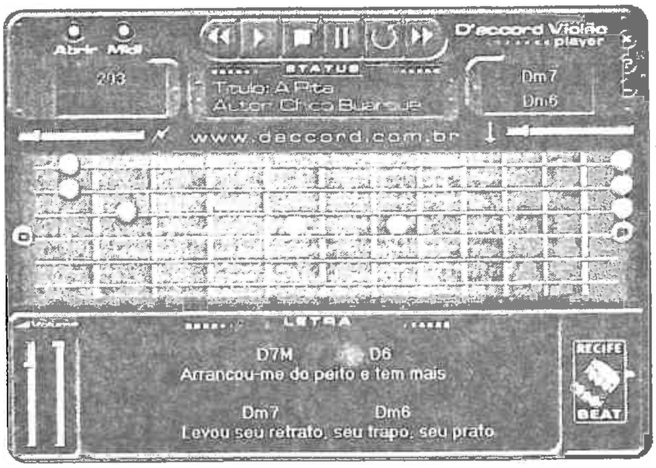

Figure 9 - D'Accord Guitar Interface in execution mode

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. Each element of the music is recorded or edited 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 9, the user may prefer a Dm7 in the fifth fret). The rhythm can be recorded from a MIDI instrument or from the computer keyboard (see Figure 10), or it 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.

Moreover, it is possible to create and store new chords and rhythmic patterns, and to assign a specific fingering and position to a chord. Finally, during edition, the user may apply a stored rhythmic pattern to a group of chords or, alternatively, directly record the rhythm of the whole song while he/she is playing on the MIDI keyboard, MIDI guitar or computer keyboard (see Figure 10).

---

[10] http://www.winoke.com
[11] http://www.realorche.com

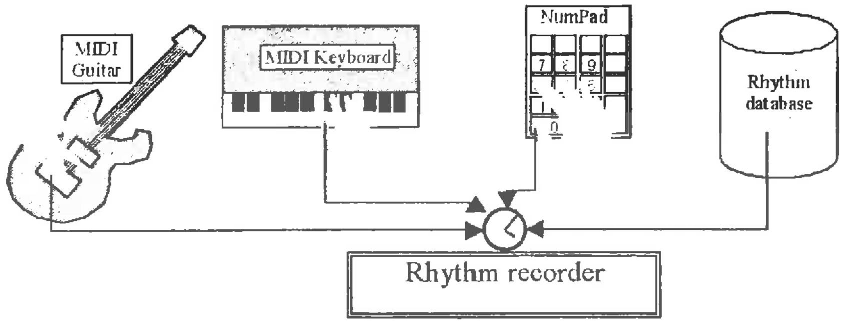

Figure 10 - General architecture of the rhythm recording module

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 chord the 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 11 shows D'Accord Guitar general architecture. Thick lines indicate the execution mode interactions, dotted lines indicate the edition mode interactions and the thin lines indicate the browser mode interactions.

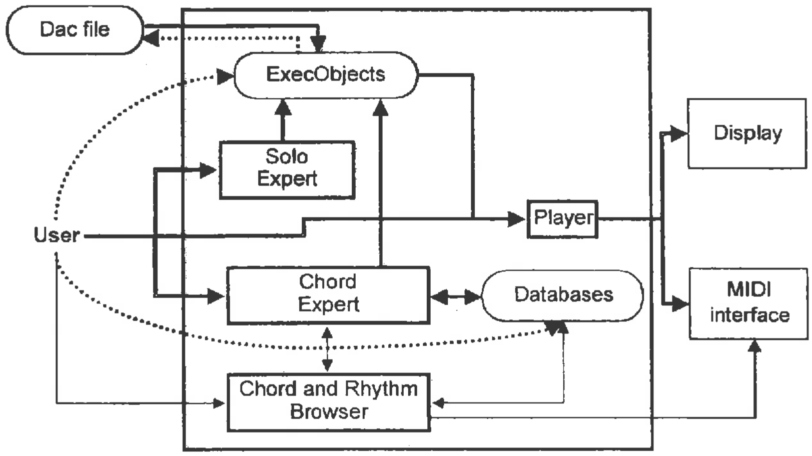

Figure 11 - D'Accord Architecture

The ExecObjects in Figure 11 are the executable musical objects. They are modeled as events, synchronized in order to play the song. There are harmonic, melodic, rhythmic, lyrics, solo and control events, each one with its own attributes. D'Accord Guitar has a proprietary file format to store or load these events, shown in Figure 11 as the Dac File.

The Browser module allows the user to browse the chord and rhythm databases and to interact directly with the fretboard.

The Chord 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 discussed in next section.

4.4. The Chord Expert

The Chord Expert is responsible for recognizing and playing chords. As explained previously, the main problem is the *best positioning and fingering in chord chaining*. In this section will be discussed a propose to solve this problem, partially implemented in D´accord Guitar.

Positioning and Fingering

Before finding the best chord position, Chord Expert needs to know all possible positions of each chord (enabling positions ranging from fret 0 to fret 12). D´accord Guitar performs an exhaustive search (Russel & Norvig 1994), filtering the results according to voicing and fingering constraints. The voicing constraint ensures that the chord position actually composes the chord (i.e. only the 5th may be omitted). The fingering constraints ensure 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 in the chord position must not exceed 4 frets.

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, and its solution is based on simple heuristics. For instance, if there are two strings pressed on different frets, the string using the lowest fret uses the lowest finger. If there are two strings pressed on the same fret, the upper string uses the lower finger. In Figure 12 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.

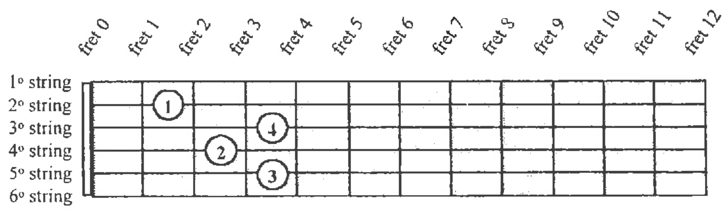

Figure 12 – a possible Db7 position in the fretboard

After the search, the system suggests values for the evaluation parameters like comfort and flexibility. These parameters are calculated based on the distance between fingers, the number of finger used and the hand orientation. These parameters together with the correspondent voicing, produces a ranking of the result.

The search done generates an amount of data impossible to deal with in a real-time application. Moreover, this data does not change after generated. For these reasons, this algorithm is executed offline and its results were stored in the Chord Database.

Best positioning and fingering in chord chaining

To find the best positions and fingerings taking into account the other chords, the system uses a hybrid model, joining a heuristic search (Rayward, Osman & Reeves 1996) with Case-Based Reasoning (Kolodner 1993). The search is based on a multi-attribute (Russel & Norvig 1994) 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.

This feature is still being validated. To illustrate the current state of it, we performed a simple test. Given the chord sequence: Em / C7M / D7 / G, the results when usual chord are preferred (**Erreur ! Source du renvoi introuvable.**) are different from those where a "smooth" bass line (possibly with unusual chord positions) is preferred (Figure 14).

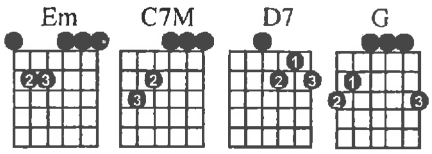

Figure 13– Most usual positions and fingering for the given chord grid

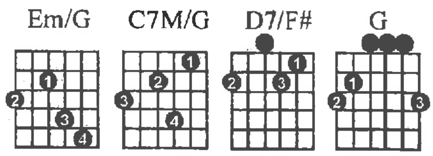

Figure 14 – Chord positions with the closest bass notes

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 problem presented.

5. Conclusion

This paper presented D'Accord Guitar, a learning, editing and performance tool, specific for guitar. D'Accord Guitar can be seen as an Instrumental Performance System with an original 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 the editing capabilities, D'Accord Guitar environment provides special mechanisms for helping amateurs 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 improvement and validation of the solutions proposed to some problems, such as transposition and fingering. Although most of the system has already been implemented and tested, it has not been concluded yet, in particular the editor modules. We intend to finish the implementation and perform a broad validation. Finally, we plan to adapt this system to other different string instruments, since their underlying representation structures are probably similar.

6. 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.

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.

Russel, S & Norvig, P. (1994) *Artificial Intelligence: A Modern Approach.* Englewood Cliffs: Prentice Hall.

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.

Todd, P & Loy, G. (1991) *Music and Connectionism*. Massachusetts: MIT Press.

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.