Constraints and Multimedia, F. Pachet, *Practical Applications of Constraint Logic Programming*, London, March 1999, pp. 3 -13

Constraints and Multimedia

François Pachet
Sony CSL-Paris
6, rue Amyot, 75005 Paris
pachet@csl.sony.fr
http://www.csl.sony.fr/Music

Abstract:

The developing field of multimedia opens up a whole new class of applications for constraint techniques. The aim of this talk is twofold: show innovative applications of existing constraints techniques, and also show how the new landscape of multimedia raises new issues pertaining to constraint research and development. I will first outline the main results in so-called automatic harmonization, i.e. the use of constraints to generate music according to given inputs such as melodies or bass lines. I will then show how constraint perturbation techniques can be used to build interactive listening systems, especially for controlling sound source spatialization. Then I show an application of constraint satisfaction for creating music programs satisfying user preferences. I illustrate this idea on a prototype music program generation system.

1. Looking for new Combinatorial Problems

Operation research and combinatorial optimization have traditionally been the main application field of constraint techniques, and is indeed today probably the main one for software industries selling constraint technology. However, we argue here that new domains should be investigated for applying constraint techniques, both for enlarging the impact of constraints, and also to boost research in new areas. This talk stresses on the interest of Multimedia in this context, as a promising field for new types of applications of constraints. Starting with the long tradition of works in constraint programming and music, we will then describe the landscape of multimedia technology, and outline three main facts which make multimedia a potentially huge field for constraints. We illustrate our claim by ongoing work conducted at our research lab in the field of interactive musical systems and music selection.

1.1 Constraints and Music

Constraints have been used in the last decade for solving musical problems with some success. The typical musical problem solved is the *automatic harmonization problem*: given a melody, the task is to compute a four-voice harmonization of the melody, which satisfy the rules of harmony and counterpoint. This problem can be seen as a constraint satisfaction problem because of the very nature of the rule of harmony, which typically state incompatibilities. A typical rule is the "parallel fifth rule"; which states that two consecutive chords should not have a parallel fifth (see Figure 1).

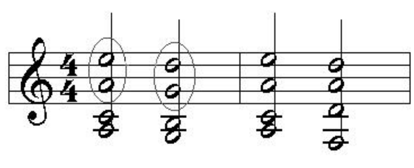

Figure 1. The first two chords violate the parallel fifth rule (the two parallel fifth intervals are represented by an oval). The two last chords do satisfy the parallel fifth rule.

Although many systems have been proposed to solve this problem (Tsang, 1991, Ebcioglu, 1992), see Pachet, 1999 for a survey), it is only recently that we can consider the pure combinatorial problem of automatic harmonization as solved, using arc-consistency-based algorithms, and an adequate representation of chords as groups of notes in the solving process (Pachet & Roy, 1998). A typical solution found by such a system is given in Figure 2.

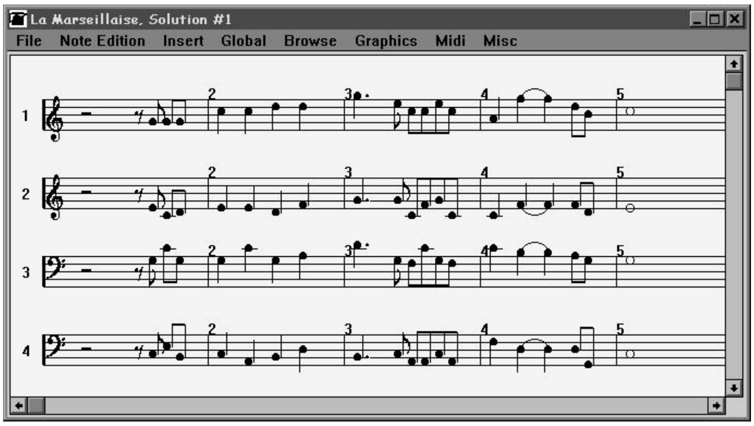

Figure 2. A Typical melody (French anthem *La Marseillaise*) and a 4-voice harmonization computed with a constraint satisfaction system.

However, we can draw many lessons from the history of constraint-based harmonization systems. On the one hand, it can be said that the pure underlying combinatorial problem is now solved: solutions can be found practically in real time for any kind of melodies, and for all sorts of musical constraints. On the other hand, the problem is now even more open than before. The first reason is that it is not enough to be able to compute all the solutions of the problem: the user usually wants the “best” solution. But specifying what means a “best” musical harmonization is impossible in terms of constraints. This would require the ability to formulate explicitly what makes a melody “nice”, in operational terms, such as multi-criteria optimization function, which is also a research issue today for constraint systems. The second reason is that the combinatorial approach somehow defines away the intentional aspect of musical composition: solutions are computed respectless of any harmonic *intention*, which is also a major drawback for assessing the interest of a solution. Representing musical intentions however is difficult, although several works have attempted to do so (Henz et al., 1996). But the most difficult aspect is to link a representation of intentions with the actual note production, in both ways. This raises the problem of controlling a solver from a high level representation of goals, which is yet another open research issue. Finally, the notion of musical “pattern” is not taken into account; which contributes to making the computed solutions sound mechanistic. A human solution (not to mention “nice”) usually contains well known patterns of notes or chords, even when they are not required by explicit rules. Combining *constraints with patterns* is something we do not know yet how to do, or in a very awkward manner.

The landscape of multimedia has of course much changed since the first works on automatic musical harmonization. In the next section, we outline the major facts of multimedia that open new doors for constraint applications, particularly in the music field.

1.2 New facts of Multimedia

Multimedia is among the promising new fields of constraint technology, for three main reasons: the size of digital catalogues, the availability of symbolic meta-data, and the new roles of end users in the production chain.

First the availability of large catalogues of multimedia information (films, video, music) to users via networks (internet, set-top-boxes, or digital broadcasting) creates a huge demand for high-level user services. Digitalization of multimedia data has yet another consequence: the availability of huge catalogues of multimedia data to users. In the case of music, there is a conceptual shift which has nothing to do with the technology of large databases. The main issue raised by this technological advance is how to access huge catalogues of music, not from a technical viewpoint, but from a user’s viewpoint. Recall the juke box, invented in the late 20s: a typical juke box would contain about 120 titles, which is the size of an average user’s discotheque. Browsing through all the titles was probably part of the pleasure, and selection could be made just like at home: by choosing one item out of a collection of items, which at least the user has seen once. Now a typical catalogue of a major company is about 50.000 items. What happens when the collection to select from is such a catalogue ? Even more terrifying, what happens if all the recorded titles become available through networks to users at home ? Estimating the total number of all recorded music is difficult, but it can be approximated to about 2 million titles (see, e.g. the size of MusicBoulevard or Amazon databases). The figure can be probably doubled to include non Western music. Every month, about 4000 new CDs are issued on the market. It is clearly impossible to apply usual techniques of music selection in this new context. What does it mean to “look for” a title when the mass of titles is so huge ?

The second fact is the possibility, with digital representation of audio-video signals, to incorporate so-called “content information” together with the data (see e.g. the upcoming and Mpeg-7 standard for audio-video content description (MPEG7, 1998; Chiariglione, 1997). These standards make it possible to use symbolic techniques such as constraints, to specify and solve problems related to audio-video manipulation and access on a large scale. Although the details of these new standard are not decided yet, they are certain to impact dramatically the nature of the software industry, by opening up a new field of applications. Why would one want to transmit such meta-data ? The interests are obvious in the context of document indexing. If musical data is accompanied with corresponding adequate descriptions, digital catalogues can then be accessed using sophisticated query systems. Meta-data opens also doors for imagining new listening systems in which the user may access data in a drastically different way. Instead of being a passive, neutral support, music becomes an active, self-documented knowledge base.

Finally, the new standard efforts convey a radically different vision of the production chain, from multimedia production to end user access: the idea that home becomes a full-fledged *reconstruction factory*, where the multimedia is not “consumed” directly, but reconstructed. The idea of structured audio has initially been devised to allow better compression of high quality audio. Standardization efforts like the Mpeg-4 project embody this idea, and try to make it practical on a large scale (see, e.g. the Machine listening Group of the Media lab, Sheirer et al., 1998). The idea is simple: instead of transmitting a ready-to-listen sound, only a description of how to make the sound is transmitted. The actual sound is reconstructed at home, or at the listener’s location, provided of course he/she has the right software to process this reconstruction properly. Structured audio actually extends this basic idea to include fully-fledged *scene descriptions*, that is, not only descriptions of individual sounds, but description of groups of sounds playing together to make up a piece of music. The actual technical details of scene description also include all what is needed to reconstruct a sound or piece of music rightfully, e.g. effects, adaptation to the local sound reproduction system, and so forth. In our context, the notion of scene description opens up new doors for meaningful controls. Indeed, since the music is delivered as a “kit”, lots of possibilities can be imagined to influence the way the kit is actually built, according to user preferences. Of course, these variations around how the kit should be assembled have to be “coherent”, which are precisely where constraint come into play.

In this context, constraints are an enabling technology for building high-level services with clear added-value. These new areas create new instances of classical problems; e.g. scheduling (digital audio broadcasting), but also whole new classes of problems. We will now outline current works at Sony CSL-Paris which use constraints for addressing these issues.

2. Research at Sony Csl-Paris: Active Listening

Active Listening refers to the idea that listeners can be given some degree of control on the music they listen to, to produce different musical perceptions on a piece of music, by opposition to traditional listening, in which the musical media is played passively by a neutral device. The objective of this research is both to increase the musical comfort of listeners, and, when possible, to provide listeners with smoother paths to new music (music they do not know, or do not like). These control parameters create implicitly control spaces in which musical pieces can be listened to in various ways. We illsutrate this idea with two concrete projects going on at our lab.

3. The MusicSpace Project

3.1 Motivations

The first parameter which comes to mind when thinking about user control on music is the spatialization of sound sources. We conduct a project for investigating the technical and conceptual issues related to meaningful user-control of music spatialization, called *MusicSpace*.

In MusicSpace, the user can listen to pieces of music using an interface in which each instrument in the piece is represented by a graphical object (see Figure 3). Moving these objects around modifies the mixing of sound sources in the global sound. Moreover, an object representing the listener himself - avatar - is also represented in the interface, so that all the mixing parameters (volume, panoramic position, etc.) are computed according to the avatar's position. The basic system provides the possibility of 1) moving around the avatar, to induce a mixing as if the listener was moving around the actual musical set-up, and 2) moving around the instruments themselves, thereby inducing a different mixing as if the listener was a sort of sound producer.

Experiments of this basic system were conducted on average listeners and music composers. It clearly appeared that although the physical actions of moving avatar or instrument icons around in a window are very similar, the possibility of moving around listener's avatars is quite different conceptually than the possibility of moving around instruments. Indeed, moving the avatar corresponds to the action of moving oneself around a musical setting. Moving instruments correspond to a more technical view on the music - the sound engineer's view. This second possibility appeared to some users as heretic, since it practically gives users the possibility of totally changing the overall mixing of the musical piece !

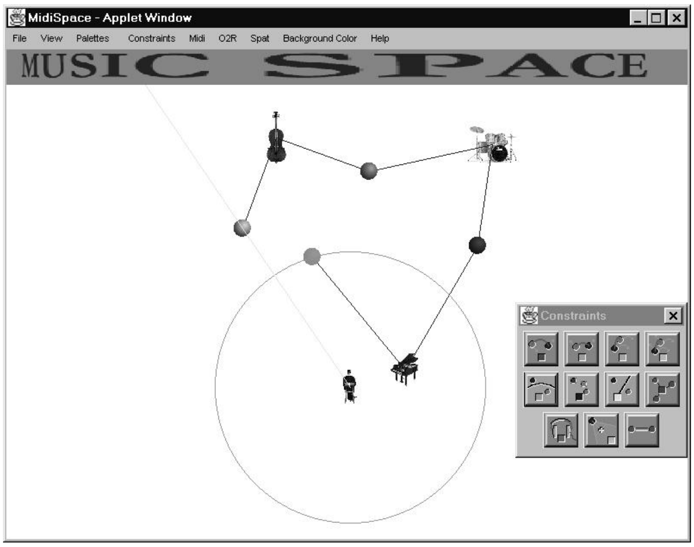

Figure 3. The interface of MusicSpace. Instruments are related by constraints. The avatar as well as instruments can be moved around by the user. The constraints embody an "automatic" sound engineer.

The second phase of our project consisted in introducing a way of somehow constraining user actions, to avoid situations where the mixing produced is totally unrelated to the original spirit of the music (Pachet & Delerue, 1998). We proceeded by introducing a particular technique, called constraint perturbation, which precisely allows instruments to be linked together by relations that are always enforced: the system uses these constraints to propagate changes, so that the set-up always remain consistent. For instance, a "related" constraint may be set between the drum and the bass, so that one of them is moved closer to the listener's avatar, the other one is moved accordingly (with the same distance ratio). On the contrary, a "balance" constraint may be set between two sound sources that should always be mutually in opposition: for instance, when the chorusing instrument is brought closer, the accompaniment is moved away. These constraints can finally be composed together to create rich environments in which users may change the instrument positions, but the constraint system ensures that the overall mixing always remain consistent with the engineer or composer constraints.

3.2 Constraints for mixing

Most of the constraints on mixing involve a collection of sound sources and the listener. We describe here the most useful ones.

- Constant Energy Level

The simplest constraint is the constraint stating that the energy level between several sound sources (*i = 1, .., n*) should be kept constant. According to our model of sound mixing, this constraint can be stated between variables *pi, i = 1, .., n* as follows:

$$\prod_{i=1}^{n} \|p_i - l\| = Cte$$

Intuitively, it means that when one source is moved toward the listener, the other sources should be "pushed away", and vice-versa. The constant value on the right-hand side of the constraint is determined by the current values of p_i and l when the constraints are defined. In practice, the total energy level may be approximated by a linear expression, yielding:

$$\sum_{i=1}^{n} \|p_i - l\| = Cte$$

Note that this constraint is non linear. Moreover, the constraint is not functional, except in the case of two sources only.

- Constant Angular Offset

This constraint is the angular equivalent of the preceding one. It expresses that the spatial organization between sound sources should be preserved, i.e. that the angle between two objects and the listener should remain constant. It can be stated between variables p_1 and p_2 as follows:

$$(p_1, \hat{l}, p_2) = Cte$$

The constraint is generalized to a collection of objects between variables $p_1, \ldots, p_i \ldots, p_n$:

$(p_1, \hat{l}, p_2) = Cte_{1,2}$; $(p_1, \hat{l}, p_i) = Cte_{1,i}$; $(p_1, \hat{l}, p_n) = Cte_{1,n}$

- Constant Distance Ratio

The constraint states that two or more objects should remain in a constant distance ratio to the listener:

$$\|p_1 - l\| = \alpha_{1,2} \|p_2 - l\|$$

This constraint can be generalized to *n* objects and the listener:

$$\forall i, j \le n : \|p_i - l\| = \alpha_{i,j} \|p_j - l\|$$

- Radial Limits of Sound Sources

This constraint allows to impose radial limits in the possible regions of sound sources. These limits are defined by circles whose center is the listener's avatar (see Figure 3) .

$\|p_i - l\| \ge \alpha_{\inf}$ (lower limit) ; $\|p_i - l\| \le \alpha_{\sup}$ (upper limit)

3.3 Issues

There are two ways to interpret MusicSpace. One is to see it as an embodiment - simplistic but operational - of a sound engineer: the user may move sounds using high level actions; the system “corrects” these actions by moving other sound sources according to his knowledge of sound mixing. This knowledge is explicitly represented as constraints.

The other viewpoint is to see mixing constraints as an *ontology of mixing actions,* which allows to mix in terms of properties of set-ups, rather than in terms of atomic actions on knobs and faders. This ontology allows to specify properties of configurations, which are guarantied to be always enforced, rather than specify explicit configurations. In this respect, constraints represent a semantics of sound source configuration, and the resulting - constrained - exploration space allows to explore various configurations without violating the spirit of the original mixing.

MusicSpace is also to be seen as an example of exploitation of “reconstructed music”. As outlined in Section 1.2, future standards will most probably deliver music by chunks, possibly transmitting sound sources separately, together with specifications on how to reconstruct the music whole from the parts. Constraints are one way of specifying this reconstruction, which nevertheless leaves room for new semantic-preserving user control.

MusicSpace is today an operational system, which can be used to control a variety of listening and mixing devices. The algorithms used for MusicSpace are in some sense *ad’ hoc*: they allow to design the constraints needed for mixing (Pachet & Delerue, 1998): the solver handles non functional constraints with inequalities, and cycles, but with a checking mode only. However, some constraints cannot be represented with the current state of the algorithm, for deep reasons. One example is the “equalization constraint”. This constraint states that the overall frequency spectrum of the total sound should remain within certain limits, defined by a set of values. Clearly, this constraint is a global constraint that holds on all the sound sources. Moreover, the constraint requires a full exploration of the search space, because the solutions depend heavily on the other constraints of the store. Combining satisfaction (exhaustive search) and reactive perturbation is something we do not know how to do today.

4. Music Catalogue Access

The issue of music delivery concerns the transportation of music in a digital format to users. Music delivery has recently benefited from technological progress in network transmission, compression of audio, and protection of digital data (Memon & Wong, 1998). These advances allow now or in the near future to deliver quickly and safely music to users in a digital format through networks, either internet or digital audio broadcasting. Moreover, as seen in Section 1.2, digitalization of data makes it possible today to transport information on content, and not only data itself. Together, these techniques give the users, at home, access to huge catalogues of annotated multimedia data, music in particular. These techniques aim at solving the *distribution* problem, i.e. how to transport data quickly and safely to users. Paradoxically, these technological advances also raise a new problem for the user: how to choose among such huge catalogues ?

4.1 Motivation and Ideas

From the user viewpoint, accessing a large quantity of music indeed is problematic: it cannot be reduced to a simple database problem, because, by definition, users do not know precisely what they look for. The problem of choosing items is general in western societies, in which there is an ever increasing number of products available. For entertainment and specially music the choosing problem is specific, because the underlying goals - personal enjoyment and excitement - do not fall in the usual categories of rational decision making. Although understanding a user’s goals in listening to music is very complex in full generality, we can summarize the problem to two basic and contradictory ingredients: desire of repetition, and desire of surprise.

The desire of *repetition* is well known in music theory and cognition. Experimental psychology shows the importance of repetitions in music. At the melodic or rhythmic levels of music “repetition breeds content”. For instance, sequences of repeating notes create expectations of the same note to occur. At a higher level, tonal music, for instance, is based on structures that create strong expectations or the next musical events to come (for instance, a dominant seventh chord creates an expectation of a resolution). Music theorists have tried to capture this phenomenon by proposing various theories of musical perception based on expectation mechanisms, particularly for modeling the perception of melodies (Narmour, 1992). At the more global level of music selection, this desire of repetition tends to have people wanting to listen music that they know already (and like) or music that is similar to music they already know. For instance, a Beatles fan will most probably be interested in listening the latest Beatles bootleg containing hitherto unreleased versions of his favorite hits.

On the other hand, the desire for *surprise* is a key to understanding music, at all levels of perception. The very theories that emphasize the role of expectation in music also show that listeners do not favor expectations that are always fulfilled, and enjoy surprises and untypical musical progressions (see e.g. Smith and Melara, 1990). At a larger level, listeners want from time to time to discover new music, new titles, new bands, or new musical genres. This desire is not necessarily made explicit, but is nevertheless as important as the desire for repetition.

Of course, these two desires are contradictory, and the issue in music selection is precisely to find the right compromise between these two forces: provide users with items they already know, *and* provide them with items they do not know, but will probably like.

From the viewpoint of record companies, one goal of music delivery is to achieve a better exploitation of the catalogue. Indeed, record companies have problems with the exploitation of their catalogue using standard distribution schemes. For technical reasons, only a small part of the catalogue is actually “active”, i.e. proposed to users, in the form of easily available products. More importantly, the analysis of music sales shows clearly decreases in the sales of albums, and short-term policies based on selling lots of copies of a limited number of items (hits) seem to be no longer profitable. Additionally, the sales of general-purpose “samplers” (e.g. “Best of Love Songs”) are no longer profitable, either because users have already the hits in their own discotheque, or because they do not want to buy samplers in which they like only a fraction of the titles. Exploiting more fully the catalogues has become a necessity for record companies. Instead of proposing a small number of hits to a large audience, a natural solution is to increase diversity, by proposing more customized albums to users.

4.2 On-the-fly Music Program Generation

The RecitalComposer Project (Pachet et al., 1999) is based on a the idea of proposing users fully-fledged music programs, i.e. sequences of music titles, instead of sets of individual titles, as in standard approaches. There are several motivations for producing music programs, rather than unordered collections of titles. One is simply based on the recognition that music titles are rarely listened to in isolation: CDs, radio programs, concerts are all made up of temporal sequences of pieces, in a certain order. This order is most of the time significant, i.e. different orders do not produce the same impressions on listeners. In a way, the whole craft of music program selection is precisely to build coherent sequences, rather than simply select individual titles. The second motivation is that properties of sequences play an important role in the perception of music: for instance, several music titles in a similar style convey a particular atmosphere, and create expectations for the next coming titles. As a consequence, an individual title may not be particularly enjoyed by a listener *in abstracto*, but may be the *right piece at the right time* within a sequence.

Rather than focusing on properties of individual titles, we can exploit properties of sequences. The proposal is therefore the following. First we build a database of titles, with content information for each title. Then we specify music programs by giving the properties or patterns we want the program to have. These properties are represented as constraints, in the sense of constraint satisfaction techniques. Finally, a constraint solver computes the solutions of the corresponding combinatorial pattern generation problem.

As an example, we will take a music program for which we specify the desired properties. Here is a “liner-note” like description of a typical music program. The properties of the sequence may be grouped in three categories: 1) user preferences, 2) global properties on the coherence of sequences, and 3) constraints on the exploitation of the catalogue. The following example describes a music program called “Driving a Car”, ideally suited for listening to music in a car:

User preferences

Note that these constraints specify global properties of the sequence, and do not specify the position of items in the sequence:

- No slow/very slow tempos (Cardinality Constraint)
- At least 30% female-type voice
- At least 30% purely instrumental pieces
- At least 40% brass
- At most 20% “Country Pop” style
- One song by “Harry Connick Jr”.

Constraints on the coherence of the sequence

- Styles of titles are close to their neighbors (successor and predecessor). This is to ensure some sort of continuity in the sequence, style-wise.
- Authors are all different.

Constraints on the exploitation of the catalogue

- Contains twelve different pieces. This is to fit on a typical CD or minidisk format.
- Contains at least 5 titles from the label “Epic/Sony Music”. This is a typical bias to exploit the catalogue in a particular region.

4.3 Database of Music Titles

The database required for building music programs contains content information needed for specifying the constraints. More precisely, each item is described by a set of attributes, which take their value in a predefined taxonomy. The attributes are of two sorts: technical attributes and content attributes.

Technical attributes include the name of the title (e.g. “Learn to love you”), the name of the author (e.g. “Connick Harry Jr.”), the duration (e.g. “279 sec”), and the recording label (e.g. “Epic/Sony Music”). Content attribute are typical meta-data: they describe musical properties of individual titles. The attributes are the following: *style* (e.g. “Jazz Crooner”), *type of voice* (e.g. “muffled”), *music set-up* (e.g. “instrumental”), *type of instruments* (e.g. “brass”), *tempo* (e.g. “slow-fast”), and other optional attributes such as the *type of melody* (e.g. “consonant”), or the main *theme* of the lyrics (e.g. “love”).

In the current state of our project, the database is created by hand, by music experts (including the third author). However, it should be noted that 1) some attributes could be extracted automatically from the signal, such as the tempo, see e.g. (Scheirer, 1998) and 2) all the attributes are simple, in the sense that they do not require sophisticated musical analysis to be filled.

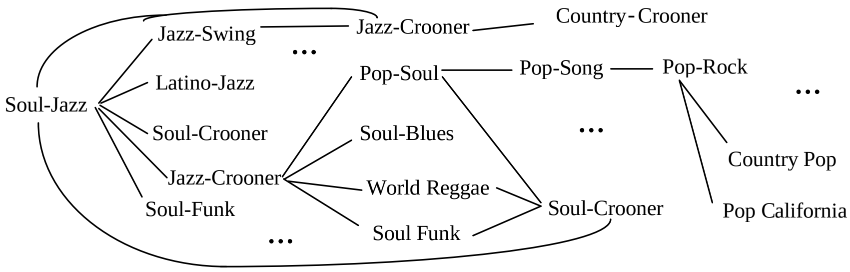

Figure 4. A part of a taxonomy of musical styles. Links indicate a similarity relation between styles. “Jazz-Crooner” is represented as similar with “Soul-Blues”.

An important aspect of the database is that the values of content attributes are linked to each other by similarity relations. These similarity relations are used for specifying constraints on the continuity of the sequence. For instance, the preceding example contains a constraint on the continuity of styles. More generally, the taxonomies on attributes values allow to establish links of partial similarity between items, according to a specific dimension of musical content.

Some of these relations are simple ordering relations. For instance tempos can take their value in the ordered list (fast, fast-slow, slow-fast, slow). Other attributes such as *style*, take their value in full-fledged taxonomies. The taxonomy of styles is particularly worth mentioning, because it embodies a global knowledge on music which is a clear added value for the system. We designed a taxonomy of styles representing explicitly relations of similarity between styles. Our taxonomy is a non-directed graph in which vertices are styles and edges express similarity. It currently includes 120 different styles, covering most of western music. A part of the graph is represented in Figure 4.

4.4 Services and Interface Issues

Computing music programs from a database and a set of constraints is shown to be a complex combinatorial problem. Constraint satisfaction techniques may be used to solve it, as explained in (Pachet & al., 1999). The resulting technique can be used to build a number of services related to music delivery with large-scale music catalogues. We list here examples of currently built applications: automatic CD assembly, a Path Builder and a Baroque recital composer. Other applications are envisaged for set-top-boxes services and digital audio broadcasting which we do not detail here for reasons of space.

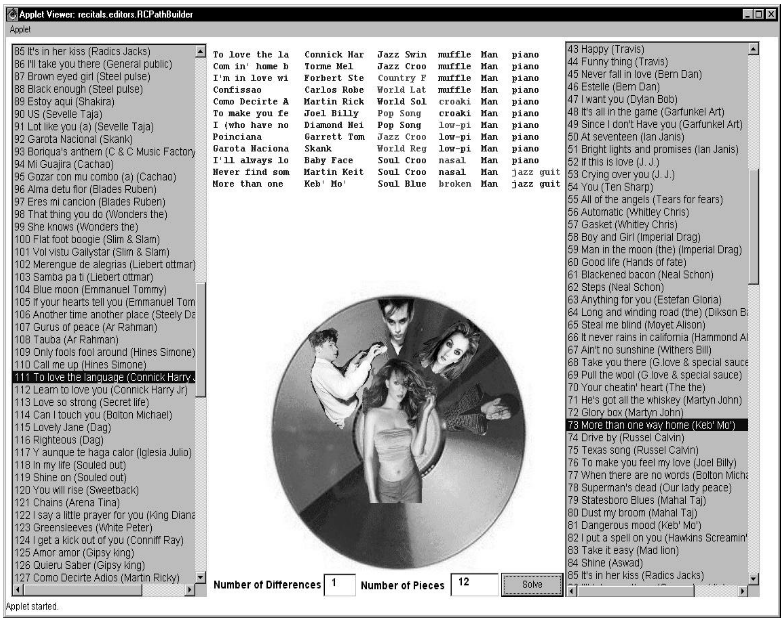

Figure 5. The PathBuilder program. The user chooses a starting and ending title, as well as a degree of tightness between successive titles.

- Sampler Builder

The simplest application of this technology is a system targeted at music professionals for building music programs (so-called samplers) from a given database. In the application, the user can specify the constraints using an interface, and launch the system on a database. This system is aimed at professionals who want to express explicitly all the properties of the desired programs, and thus have full control on all the constraints.

- Progressive programs

In this scheme, the user only specifies the stylistic structure of the program: the genres of the beginning, middle and end. This may be used for instance for creating long programs for parties, in which you know in advance the structure (e.g. begin with Pop, then Rock, then Slows, etc.).

- Path across different styles

Services dedicated to average end users should allow them to express only their preferences, possibly using automatic profiling systems, and contain predefined, fixed constraints for the coherence properties and catalogue exploitation, according to predetermined ambiences or configurations. A typical configuration is a path between two titles. In this scheme, the user can specify a starting title and an ending title. The system contains hidden constraints on continuity of genres, and tempos are fixed. For instance, find a continuous path between Céline Dion's "All by myself", and Michael Jackson's "Beat it" (see Figure 5).

- Specific music domains

The approach can be used to produce music programs in specific styles, by adding domain specific constraints. A prototype application dedicated to Baroque music implemented in our lab allows to build various "recitals" in the domain of Baroque harpsichord music. Baroque music is a good example of a specific domain, because recitals of Baroque music (XVIIth century) follow rules identified by musicologists, while allowing a great deal of freedom to performers. A typical rule concerning the structure of recitals is the "continuity of tempos" between consecutive pieces. More specific rules are also used, such as rules on the tonality: at this period of musical history, recitals where allowed to modulate - i.e. change tonality - only once. Other constraints concern the structure of the recital (introductory part with necessary piece types), as well as necessary alternation of piece types.

The system allows the user to create and listen to different music programs, while ensuring the consistency of these programs, according to the rules of the structure of recitals. The database contains titles with content description adapted to the domain. For instance, attributes such that "type" (e.g. "Gigue", "Chaconne", etc.), "tonality" and "density" are added to the database for describing relevant aspects of titles. The constraint system contains the constraints corresponding to the rules described above. The resulting system allows to produce a great number of different recitals, which all have the desired properties of "good" recitals, in the style of the composer's time.

This kind of service lies between two extreme bounds: fixed order and randomness. On the one hand, a CD played in a standard fashion contains a fixed music program. On the other hand, a common feature of CD players (or Juke boxes) is the "random" selection button, which chooses at random between different CDs and between the titles of the CDs. Constraint techniques provide an intermediary degree of control between these two extremes, where the user can still express some preferences, but the system computes a program which yields properties of coherence.

4.5 Issues

RecitalComposer is an enabling technology for building high-level music delivery services exploiting large-scale music catalogues. The system is based on the idea of creating explicit sequences of items, specified by their global properties, rather than computing sets of items satisfying queries. One of its main advantages over standard approaches in music selection is that it produces ready-to-use ordered sequences of items. It creates coherent music programs from user specifications, where the coherence is specified in terms of meta-data on music titles and as such can be seen as another example of "semantic" control, where the semantics is the structure of music programs. Compared to the juke box of the 20s, it allows to access much larger music catalogues with simple controls (user preferences) which make sense, and do not require an *a priori* knowledge of the underlying music catalogue.

5. Conclusion

There are lots of other applications of constraints in the field of multimedia that we do not mention here for reason of space: constraint-based drum machines, in which constraints are used to specify temporal relations between percussive instruments, constraint-based control of sound synthesis, constraint-based classification systems for annotation of multimedia document (ongoing DIVAN Esprit project), a etc. These applications and projects show that the new landscape of digital multimedia opens new doors for interactive multimedia environments. We have argued that such environments require some sort of semantic preserving systems. We have illustrated this idea with two projects currently developed at Sony CSL, in the areas of sound spatialization, and content-based music selection. In both cases, the technology of constraints is proposed for representing "seeds of semantics" that endow exploration spaces with meaningful user controls.

Constraints and Multimedia, F. Pachet, *Practical Applications of Constraint Logic Programming*, London, March 1999, pp. 3 -13

References

Chiariglione, L. "MPEG and Multimedia Communications", IEEE Trans. CSVT, Vol.7, No.1, Feb.1997.

Ebcioglu, K. An Expert System for Harmonizing Chorales in the Style of J.-S. Bach. Understanding Music with AI: Perspectives on Music Cognition. AAAI Press: 294-333, 1992.

Memon, N, Wong, P. W. “Protecting Digital Media Content”, Communications of the ACM, July 1998, pp. 34-43, 1998.

MPEG7 Requirements Group, “MPEG-7 Requirements Document”, Doc. ISO/MPEG N2461, MPEG Atlantic City Meeting, October 1998.

Narmour, E. The analysis and cognition of melodic complexity. University of Chicago Press, 1992.

Pachet, F. A survey of constraint-based musical harmonization systems, Constraints, special issue on constraints for artistic applications, to appear, 1999.

Pachet, F. Delerue, O. “A Constraint-based Temporal Music Spatializer”, ACM Multimedia Conference, Brighton, 1998.

Pachet, F. Roy, P. Cazaly, D. A Combinatorial Approach to Content-based Music Selection. Submitted to IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, 1999.

Pachet, F. Roy, P. Formulating Constraint Satisfaction Problems on Part-Whole Relations: The Case of Automatic Musical Harmonization, Ecai workshop on Constraints and Artistic Applications, Brighton, 1998.

Sheirer, E. Väänänen, R. Huopaniemi, J. “AudioBiffs: The MPEG-4 Standard for Effects Processing”, Proceedings of the first Digital Audio Effects Workshop, Barcelona, pp. 159-167, November 1998.

Tsang, C.P. Aitken, M. Harmonizing music as a discipline of constraint logic programming. ICMC ' 91, Montréal, 1991.

Van Hentenryck, P. Saraswat, V. Strategic directions in constraint programming ACM Computing Surveys, Vol. 28, No. 4 (Dec. 1996), Pages 701-726.

M. Henz, Lauer,S. Zimermann, D. CompoZe- Intention-Based Music Composition Through Constraint Programming. 8th IEEE International Conference on Tools with AI, Toulouse (France), 1996