A Joyful Ode to Automatic Orchestration

FRANÇOIS PACHET, Sony Computer Science Laboratories - Paris

Most works in automatic music generation have addressed so far *specific* tasks. Such a reductionist approach has been extremely successful and some of these tasks have been solved once and for all. However, few works have addressed the issue of generating automatically fully fledged music material, of human-level quality. In this article, we report on a specific experiment in holistic music generation: the reorchestration of Beethoven's *Ode to Joy*, the European anthem, in seven styles. These reorchestrations were produced with algorithms developed in the *Flow Machines* project and within a short time frame. We stress the benefits of having had such a challenging and unifying goal, and the interesting problems and challenges it raised along the way.

CCS Concepts: • **Information systems → Multimedia content creation;**

Additional Key Words and Phrases: Machine learning, music, generation

ACM Reference Format:
François Pachet. 2016. A joyful ode to automatic orchestration. ACM Trans. Intell. Syst. Technol. 8, 2, Article 18 (October 2016), 13 pages.
DOI: http://dx.doi.org/10.1145/2897738

18

1. INTRODUCTION

"Computer programs that compose music have been around almost as long as the computers that run them" [Smoliar 1991]. Indeed, automatic music generation has been a recurring theme of computer music research since the 1950s. The Illiac suite [Hiller and Issacson 1959] showed that simple Markov models could capture surprisingly well local features of music melodies. The program designed and implemented by Kemal Ebcioglu to generate Bach chorales using a hand-made rule system [Ebcioglu 1990] produced impressive results that remained a reference for years. David Cope's work on music style modeling [Cope 2005] showed that computers could produce models of musical style to assist productively a composer (David Cope himself). Since then, the domain of automatic music composition has grown substantially (see Fernández and Vico [2013] for a comprehensive survey).

The general trend has been to apply *reductionist* approaches: many problems considered initially as minor ones have been isolated from their context and addressed using increasingly sophisticated technologies. With the progress of artificial intelligence notably, music generation has progressively turned from a subdomain of computer music addressing ill-defined problems to a fully fledged area of machine learning. As an example, we can consider the problem of identifying the underlying tonalities of a chord sequence. This problem is of little musical interest in itself, but it makes sense in the context of improvisation, as it enables the musician to find interesting scales to use for

This research is conducted within the Flow Machines project funded by the ERC under the European Union's 7th Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 291156. Parts of the work presented here was funded by the Lrn2Cre8 project (FET grant agreement n. 610859).
Author's addresses: F. Pachet, SONY CSL, 6, rue Amyot, Paris, France; email: pachetcsl@gmail.com.

DOI: http://dx.doi.org/10.1145/2897738

building up a musical discourse. This problem was addressed with an ad hoc, *island-growing* approach [Ulrich 1977] and validated by a single eight-bar example. What was a small subtask deserving a few paragraphs in 1977 is now the subject of specific studies that focus on the problem as such, with an increased level of sophistication (see, e.g., Granroth-Wilding and Steedman [2014]). Obviously, by becoming context independent, these works also tend to forget their raison d'être (in that case, guide a music generator to produce better improvisations).

Such a scheme has been at stake in virtually all domains of music generation. As a result, several subtasks have been formulated as well-defined, general AI problems, and some of them have been actually solved. For instance, the problem of sampling sequences from a Markov model under various types of local and global constraints has been repeatedly mentioned as a stumbling block in statistical music generation, since the first results in musical Markov chains [Hiller and Issacson 1959]. After many attempts using generate-and-test methods, the problem has recently been solved once and for all, with a polynomial complexity solution, involving a mix of constraint programming, automata theory, and belief propagation [Papadopoulos et al. 2015].

However, *holistic* approaches to music generation are still rare. The problem of generating music material of human-level quality by addressing *all* the dimensions at once, and not only specific ones, has made arguably little progress. This article describes such a recent attempt in music generation using models and techniques developed in machine learning, statistical inference, and computer music, and through a specific experiment in automatic reorchestration.

2. A COMMISSION FROM THE COMMISSION

Most of the research described here is conducted in the Flow Machines project,[1] which aims at designing and studying systems able to generate music and text in the style of an author, under various user constraints [Pachet 2015]. The experiment also includes contributions from the Lrn2Cre8 project,[2] which aims at producing computational models of musical creativity.

At about the midterm of the Flow Machines project, we were asked by the European Commission if our style imitation software could be used to reorchestrate the European anthem, to be played at the European Parliament.[3] Such a request could not be turned down, so we agreed. We had a couple of weeks.

Of course, we did not have ready-to-use software for this task. We had bits and pieces: some powerful but incomplete algorithms, mixing in novel ways of statistical inference with combinatorial optimization, ideas and intuitions, and a lot of data (lead sheets and audio recordings). We could generate nice but imperfect scores for specific cases. Reductionism and the pressure to publish make it virtually impossible to publish an article with *several* contributions, so a natural tendency is to cut the problem into slices: melody, harmony, timbre, groove, and so forth and to solve each subproblem as it comes, hoping that some day someone would put all the pieces together. That day had come.

So we started working. The constraint was to orchestrate Beethoven's *Ode to Joy*, in several styles. We came up with a selection of seven styles, varied yet popular, all of which raised different modeling challenges for music generation. The styles were as follows:

[1] www.flow-machines.com.
[2] www.lrn2cre8.eu.
[3] For a ceremony to celebrate the 5,000th ERC grantee.

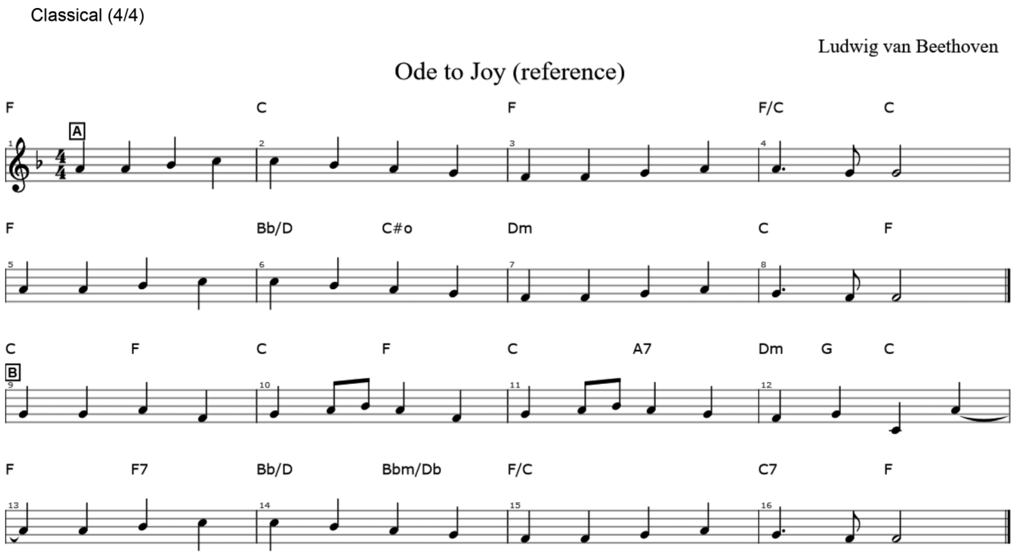

Fig. 1. The lead sheet of *Ode to Joy* from LSDB. This lead sheet was produced by ear as Beethoven did not compose with lead sheets.

(1) Bach chorales. We wanted essentially to avoid the criticism "Your styles are all fine, but what about Bach?" This style of music is a typical exercise, both for music students and for computer music researchers. We had to address it somehow.
(2) Bossa nova. Brazilian music has long been a subject of study of the group [Pachet and Cabral 2013], and the ongoing Brazyle project[4] provided us with many recordings of Brazilian guitarists.
(3) Jazz. We had been working on jazz modeling for a long time, and we wanted in particular to imitate the style of the band *Take 6*, for which we happened to have a set of 10 complete transcriptions.
(4) Lounge. We had an interest for chill-out lounge music from the 1990s as an ideal target style to study multi-instrumental electronic music. We also had a set of high-quality transcriptions of complete lounge pieces.
(5) *Penny Lane* by the Beatles. Relating research with a Beatles title has become a sort of trademark of computer music research, and was an obvious choice.
(6) A commission officer asked for the style of Ennio Morricone. We picked up the piece *Chi Mai*, discarding other tunes that do not have the same international coverage.
(7) *Prayer in C*. This song was released by pop duo Lilly Wood and the Prick in 2010. German producer Robin Schulz released a remix of the song as a single, which became a worldwide hit. We wanted to also target more recent styles and picked up this song, easily identifiable by its guitar riff and enjoyed by younger generations all over the world.

The goal was to generate music so that attendants would recognize easily both the lead soprano (*Ode to Joy*) and the orchestration style. The only information we could feed our software as input was the lead sheet of *Ode to Joy*, taken from the LSDB database [Pachet et al. 2013] (see Figure 1), and containing the melody and chord progression, as well as representative samples of the orchestration style. The resulting orchestrations were produced in due time, played successfully during the ceremony,

[4]www.brazyle.net.

and are available to readers,[5] as is a more didactical video.[6] The next sections describe the main technical challenges and problems encountered along the way.

3. A MAX ENTROPY MODEL FOR BACH CHORALES

Bach chorales are known to be interesting for the inner movements of voices more than, say, for their exotic harmonic modulations and digressions (as opposed to jazz, see later). Modeling multivoice music is a big challenge in music representation. Departing from the approach of Ebcioglu [1990], we built a model of Bach chorales from an automatic analysis of existing chorales that could be constrained with the soprano to *Ode to Joy*. Obviously, multivoice music cannot be modeled as Markov chains, so more elaborate models of music had to be employed. Recent attempts at using deep learning models [Boulanger-Lewandowski et al. 2012] are promising but not yet able to fit our needs, so we had to look at other models.

Max entropy models are widely used in physics and biology, in particular to infer probability distributions that satisfy observed pairwise correlations [Lezon et al. 2006]. Such models (see, for instance, Ekeberg et al. [2013]) are variations of the so-called *Potts* model [Potts 1952] developed in statistical mechanics. The primary advantage of these models, in our context, is precisely that they can capture faithfully the statistics of binary correlations between events, either horizontally or vertically, and possibly for long range, as opposed to Markov models.

In practice, we represent the sequence to be generated by a probability distribution on a number of variables (for instance, representing the pitch of the notes): $P(X_1, \ldots, X_N)$.

Markov models assume that the probability of events depend only on a small number (the order) of preceding events. Consequently, this probability can be written as a product of probabilities for each event as follows (here for order 1):

$$P(X_1, \ldots, X_N) = P(X_1) \times \prod_{i=2}^{N} (P(X_i|X_{i-1}).$$

By contrast, the probability distribution of a sequence in the considered max entropy models is represented as a Boltzmann-Gibbs distribution of the form

$$P(X_1, \ldots, X_N) = \frac{1}{Z} e^{-H(X_1, \ldots, X_N)},$$

where H, the *Hamiltonian*, represents the energy of the system and is itself represented as a sum of potentials, each corresponding to interactions between neighboring notes, and Z is a normalization factor.

Following earlier works on the application of these models to monophonic music [Sakellariou et al. 2015], Gaétan Hadjeres[7] extended the basic model developed for melodic patterns to handle specifically four-voice homo-rhythmic music (i.e., so that each chord is composed of notes with the same duration).

The model for monophonic sequences was extended to handle four-voice music and trained on a set of Bach chorales, which abound on the web. The model could generate pretty impressive random chorales, which not only replicated the statistics of binary correlations but also were able to create interesting new harmonic progressions, which all sounded Bach-like, at least from the viewpoint of inner voice movements.

These models can be controlled in a number of ways, and in particular via so-called *local fields*. Local fields are the probabilistic equivalent of unary constraints and can be

[5] www.flow-machines.com/OdeToJoy.
[6] AAAI 2016 best video award, http://aivideocompetition.org/aaai-video-competition-winner-2016/.
[7] He was the ideal person, being both a student at the conservatory and a PhD student in mathematics.

Fig. 2. *Ode to Joy* in the style of Bach chorales (complete score).

imposed at arbitrary positions of the sequence in order to bias locally the probability distribution of events (here notes at a specific position). Imposing *Ode to Joy* at the soprano consisted of imposing a local field for each variable of the soprano to be the corresponding note in Beethoven's melody (i.e., forcing the probability of that note to 1 and the others to 0) and propagating the changes in the model. Interestingly, imposing the harmony was not necessary in that case: coherent inner movements and harmonic motion were naturally generated from the model without human intervention. The generated scores, obtained by sampling the model and without adding any musical knowledge, are surprisingly good (see Figure 2).

More work was done after the *Ode to Joy* project to assess precisely to which extent these models actually produce novel chords and chord progressions, making them ideal candidates for computational models of creativity [Hadjeres et al. 2016].

4. JAZZ WITH A FIORITURE-BASED MARKOV MODEL

The jazz orchestration also used a statistical model, but a different approach: jazz arguably requires more digressive conduct than Bach chorales, and such a variety is not well modeled (yet) by statistics *only*. We needed a model able to generate interesting variations, especially harmony-wise, like a jazz pianist typically does when he or she performs a well-known (and therefore slightly boring) tune: digressions, substitutions, and *side-slips* [Pachet 2012] are the rule.

To achieve this goal, we used a corpus of 10 musical scores by a cappella band *Take 6* of six-voice harmonizations of standard gospel tunes. We then proceeded in two phases. In a first phase, we built a chord sequence using dynamic programming, so that each chord fitted optimally with the corresponding chord label of the lead sheet, and so that chord transitions are smooth bass-wise (i.e., no big jumps). This chord

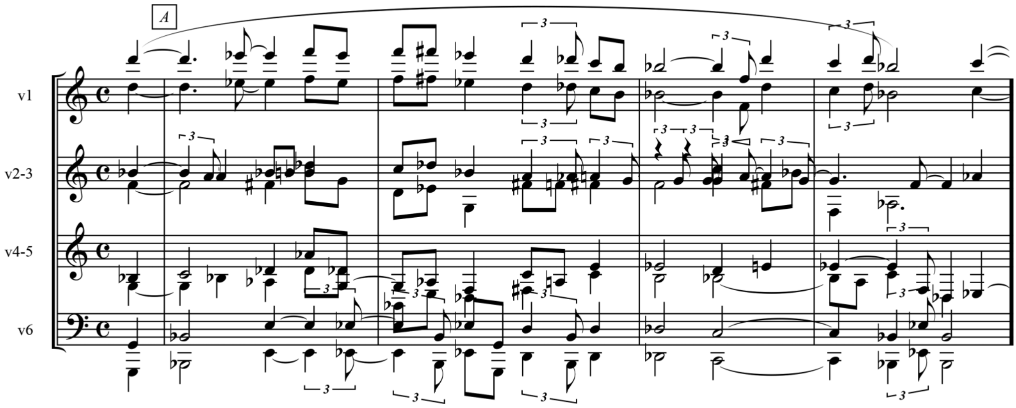

Fig. 3. *Ode to Joy* in the style of Take 6, piano version (extract).

sequence provided a conformant harmonization of the lead sheet but did not include any interesting deviation from the lead sheet, as a jazz musician would typically do. To introduce such deviations and generate creative harmonization, we let the system perform *controlled random walks* during long notes, called *fioritures* [Pachet and Roy 2014]. These random walks are performed in a Markov model of chord transitions estimated from the corpus. Technically, we used a loose representation of chords (a viewpoint consisting of only bass and soprano), so that the resulting model was not too sparse.

This approach generated convincing six-voice harmonization in the style of Take 6 (see Figure 3). Interestingly, although this model does not have the ability to invent new chords (as opposed to max entropy models), this approach turned out to be more satisfactory for the jazz style. One reason is that the underlying Markov model of chords was, in that case, sufficiently rich to produce interesting sequences. As a consequence, its inability to invent new chords is hard to detect by ear without listening to many hours of generated material. Note that the rhythms of the fioritures (see Figure 3) are generated by sampling randomly compatible rhythms (i.e., rhythms with the same metrical position and duration as the original note) from melodies taken from the LSDB database [Pachet et al. 2013].

5. BOSSA NOVA AND GROOVE PRESERVATION

The bossa nova orchestration could not be MIDI based. Bossa nova is so deeply rooted in the sound of acoustic guitar that we had to use an audio model to produce convincing bossa nova guitar accompaniments. Bossa nova can be characterized by two features: sophisticated harmonies and groovy rhythm patterns [Pachet and Cabral 2013]. We used a concatenative synthesis approach [Maestre et al. 2009], which enabled us to reuse real, high-quality, leadsheet-synchronized recordings of Brazilian guitarists, collected a few months before during the Brazyle project.[8]

However, segments cut from human recordings raise problems because onsets are never played exactly *on the beat*. They are often either slightly before (anticipations) or after (delays), for instance, to create specific grooves. A smart gluing mechanism (see Figure 5) ensures that these onset deviations are preserved and restituted during generation, without producing glitches or perturbations of the overall groove [Ramona

[8]www.brazyle.net.

Fig. 4. The lead sheet of *Ode to Joy*, jazzified.

et al. 2015]. We came up with a convincing guitar bossa nova track that fits with the harmony of the song, as given by its lead sheet.

However, Beethoven's chord progression is too simple to be musically interesting when played in bossa nova style. So we designed a chord grid *variation generator* to produce a more interesting, harmonically sophisticated version of the target lead sheet (see Figure 4). This generator is based on dynamic programming.[9] We then used this jazzified lead sheet as a target for the concatenative synthesis engine. The resulting generation was perfect to the ears of the guitar players of the team.

However, it still lacked something. Drums, bass, possibly violins, and someone to sing or play the tune were needed to evoke songs like *Girl from Ipanema*. Now that we had a concrete and convincing guitar accompaniment generator, the need for complementing it with other instruments was blatant! We identified the missing ingredients: a simple bass, a typical rim shot back up, and violins that come into the game at section B. We implemented basic generators in ad hoc ways, guided by the specific goal of having to evoke a bossa nova for a lay audience. Previous works have addressed the problem of generating accompaniments for popular music styles, for example, with an HMM [Simon et al. 2008] or other machine-learning techniques [Chuan and Chew 2007]. However, these works are difficult to reuse as ingredients of larger systems. Most importantly, they are based on general assumptions about accompaniments, which did not fit our needs. The violin part, for instance, was better modeled as a monophonic line played across chord changes so as to minimize intervals and maximize the use of thirds and fifths of the underlying chords.[10] In any case, this approach worked perfectly for this particular context. The result sounds great to our ears.

[9]Similarly to the chord sequence analysis problem mentioned in the introduction, this variation generation problem could be isolated from its context and treated as such, probably with more sophisticated techniques and results.
[10]Another example of a problem that could be worth extracting from this specific context and addressing with better tools...

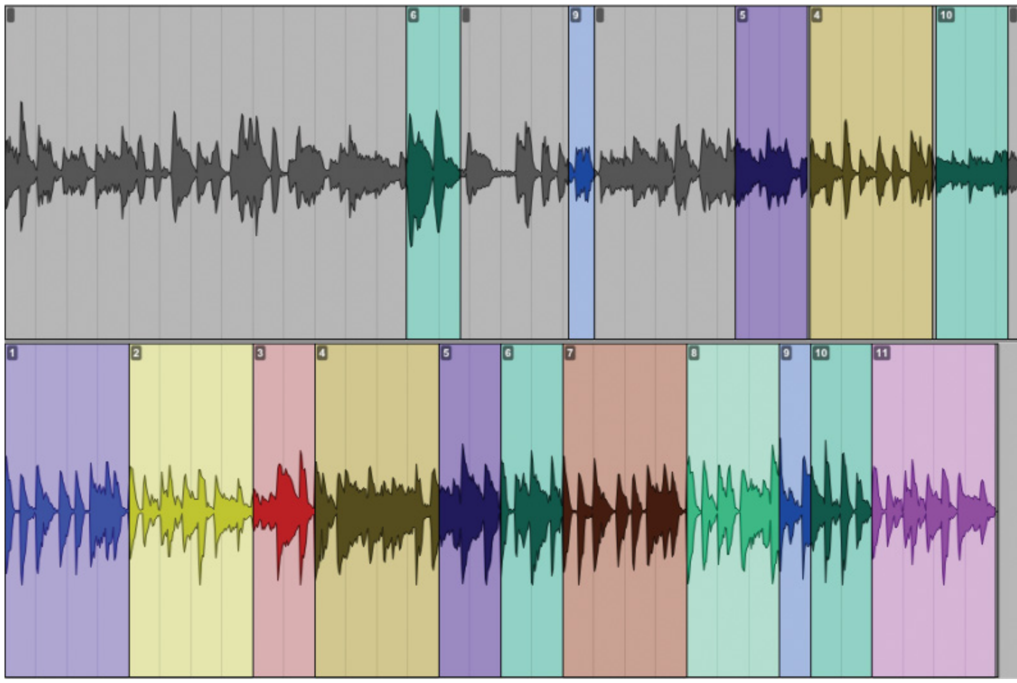

Fig. 5. Groove-preserving concatenative synthesis to generate a convincing bossa nova accompaniment of *Ode to Joy* (extract). Segments of the source audio file (top) are reused, modified, transposed, and stretched to generate the target (bottom).

6. SONG-SPECIFIC STYLES

Three of our seven styles were not styles per se but individual, popular songs. For the *Penny Lane* (the Beatles), *Chi Mai* (Ennio Morricone), and *Prayer in C* (Lilly Wood and the Prick) styles, we first asked a pop artist and arranger (Benoit Carré) to produce multitrack recordings of the original songs. We then fed these multitracks to *FlowAudio*, a constraint-based audio generator developed by Marco Marchini, based on the *meter* constraint [Roy and Pachet 2013]. FlowAudio models audio sequences as a finite-domain constraint satisfaction problem, to which various Markov constraints are applied to enforce metrical and synchronization relations.

However, it did not work at all! Not because the system had bugs (it also had bugs), but because it was not designed to handle the myriad details that turned out to be crucial to generate a convincing piece. For example, we did not deal at all with the structure: the fact that a song has several parts, and the transition from one part to another one usually corresponds to different orchestrations.

So we had to design and implement urgently these features in the system. The basic mechanisms underlying FlowAudio were kept, but we added provisions and GUI, for example, for setting all the required parameters for each part, in a sort of extreme programming [Beck 2000] atmosphere. Eventually, convincing multitrack generations were produced for each of the three song-specific styles (see Figure 6). However, the combinatorial nature of the problem was not treated satisfactorily (generations were obtained by simplifying the problem and cutting it into several phases, thereby limiting the number of solutions). Again, this exercise helped us identify a new class of scheduling problems, involving precisely vertical constraints defined metrically between various time series [Roy et al. 2016].

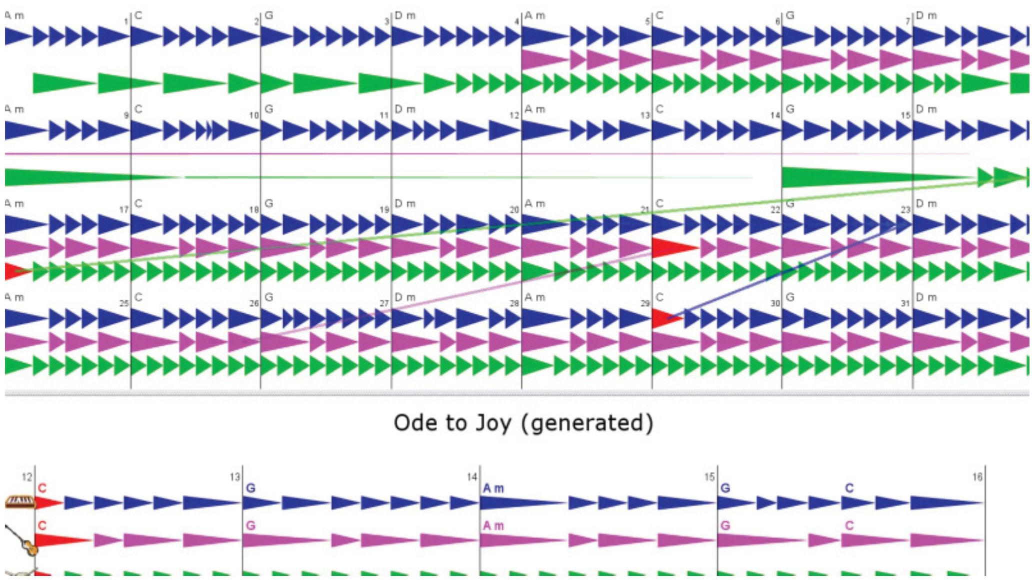

Fig. 6. *Ode to Joy* in the style of the song *Prayer in C* (Lilly Wood and the Prick). Each track (piano, bass, drums) is segmented into chunks (musical events) using an onset detector. Chunks are then reorganized using Markov constraints to fit with the chord progression and structure of the lead sheet, scheduled and mixed to produce the final output.

7. PRODUCTION

Generating the basic music material was not enough. Our goal called for a good *production*. The following section describes the main production operations conducted to deliver audio files that could be rendered in a real-world setting. During production, the musical content of the generated material was not modified by hand; only the renderings and mix were done manually. These indications are given here as examples and possible *use cases* for a yet-to-be-designed automatic production engine.

7.1. Bach

Preliminary renderings of the four-voice Bach chorale showed that the melody was not sufficiently prominent to be recognizable. Simply raising the level of the soprano did not produce convincing results. Using human voices was discarded, since singing voice technologies are not able to produce perfect voices yet, and we wanted to avoid attention being drawn to deciding whether the voice sounded natural or not. Eventually we decided to use two different sounds for the soprano and for the other voices.

The choice of an organ sound for the other voices came quickly, to remain in the style of Bach as much as possible. The sound had to be such that the three voices were intelligible, that is:

(1) Sound should not have too many resonances in any particular spectral zone.
(2) Sound should have short attack and release times.
(3) Sound should be bright enough so that at least some frequencies get into the ear's most sensitive spectral zone (1–4 kHz).

We finally picked up an oboe sound for the melody. However, we could not find a good enough sound in our sound libraries, so we eventually recorded a human musician

(Anne Régnier). The sound was slightly reverberated, to evoke an acoustic space similar to the organs (i.e., not direct).

7.2. Bossa

For the bossa nova, the goal was to evoke sounds of the 1960s, easily recognizable by a large audience thanks to hits such as *Desafinado* or *Girl from Ipanema*. We chose a piano sound, to evoke Tom Jobim's recordings (such as Jobim [1967]), in which he plays the melody with a sense of restraint and minimalism, similar to the scarce piano interventions of Count Basie.

The piano sound was chosen to also sound like the 1960s: medium with soft attacks. The sound was doubled to the upper octave, such a treatment being possible by the low note density of the piece. We added compression and large room reverb, a standard practice in pop.

Finally, and although guitar accompaniment was the main focus of this generation, we realized we needed some percussion to complete the picture. We added a simple drum track consisting of a repeated two-bar bossa nova pattern. The track was equalized to emphasize its percussive side: high mediums and basses were added. We used a mix of virtual instruments (bass drum) and real ones (rim shot, brush loop, and hi hat). We also added a triangle, as an additional hint to the 1960s. We applied compression, large room reverb, and damped frequencies above 10kHz.

7.3. Jazz

After trying many renderings of the generated score for Take 6, we realized we had two issues:

(1) Intelligibility. To be intelligible, each voice had to have short attack and release times, hence the choice of a piano sound. The sound itself was chosen to reach a compromise between realism, brightness (so that the most sensitive spectral zones of the ear are used), and roundness (i.e., lower frequencies to smoothen the sound).
(2) Expressiveness. The generated score sounded too mechanical when rendered with a polyphonic sound. Expressiveness was added manually, with subtle-onset deviations and dynamics. Onset modification was synchronous on all parts to preserve intelligibility. Note that the automatic generation of expressive performance is one of the targets of the Lrn2Cre8 project, and this is a task that could be automated.

Finally, the bass voice was doubled to balance the overall spectral profile.

7.4. Penny Lane

The main concern for the *Penny Lane* production was to recreate a sound from the mid-1960s, early Beatles period. Taking inspiration from Milner [2009] and Kehew and Ryan [2006, pp. 132–137], we ensured that the mix (1) did not contain too many very high or very low frequencies, to reproduce the limitations of the recording equipments of that period, and (2) applied an amount of compression comparable to the one used during this period (a crest factor of about 12db and no brickwall digital limiting, as described in Deruty and Pachet [2015]). One can hear in the result that such a concern hampered the intelligibility (by today's standards) of the rhythm track, which is at times intentionally muddy.

We chose an English horn sound for the lead soprano because of its rather medium frequencies (as opposed to oboe, for instance). Basic equalization and compression made it sound like a trumpet, as a reference to the original *Penny Lane* mix. A high-pitched recorder was used to add some missing high frequencies to the horn. Both were synchronized to sound as a unique instrument.

7.5. Prayer in C

As described earlier, the original song was re-recorded to get three individual tracks (guitar, bass, drums). This task was challenging because the sounds in this music genre are most of the time obtained through complex combinations of various synthesizers. For instance, the hi hat contains a *TR808* sound, probably mixed with other, unidentified sound sources.

For the soprano, we recorded an acoustic guitar, which distinguishes itself nicely from the basic riff while retaining a clear guitar-like quality (a challenge). Guitar was produced with reverberation and delay. Equalization was applied as an insert to brighten the sound.

An important aspect of the mix was to use *side-chain compression*, which highlights the kick drum while making the whole mix more compact (a method heavily used by bands such as Daft Punk). Finally, the mix had to be bright, large (pan-wise), and dynamically limited to sound like the original.

7.6. Mastering

Mastering had to address two issues: (1) handle *disparate* pieces, from Bach to lounge music, and (2) enable a rendering fit for a *noisy* environment. For (1), we equalized each piece to make them sound closer to each other than initially. For (2), we controlled the dynamics with a limiter to ensure a constant level. Since people were supposed to talk during the playback, we emphasized medium high frequencies (the frequencies of speech). As a summary: the master could not sound beautiful, so it had at least to be audible!

8. CONCLUSIONS

Regardless of the various outcomes, this experiment was above all an experiment in holistic research. The main lesson learned is that focusing on one dimension of music only (be it the melody, chords, harmony, rhythm, timbre, etc.) is not enough to generate anything convincing to a wide audience. The beauty (and difficulty) of music is precisely that it requires a simultaneous consideration of all its dimensions at once to make sense.

By having to put together several ingredients of music generation, we identified new problems we had not thought of, some of which are mentioned in this article. Another interesting problem we found is: "how to remove notes in a melody so that you can still recognize it?" This process is at play with the melody used for the lounge style: almost all notes of the original theme were removed, but it is still somehow recognizable. This was done by hand in this project, but the question remains: how to design an algorithm that strips out notes and keeps only the essential ones so that the melody can still be recognized?

A more general problem that became apparent is that if the goal is to produce music to be played to a standard audience, expressiveness was eventually more important than content: a bad score can sound great if rendered the right way. A great score can be destroyed by a mechanical interpretation. Automatic generation of expressive performance has so far mainly focused on classical music, but this domain is progressing fast (see, e.g. Widmer [2016]).

A yet unaddressed and fascinating issue is the interplay between expressiveness, timbre, and symbolic content, which obviously plays a central role in human composition. In particular, pop music is not composed, *then* rendered, *then* produced: everything is done concurrently, but we do not yet have good models of this process.

While reductionism produces regularly beautiful scientific results, it is not sufficient to achieve the dream of automatic music generation. Integration is not just about putting modules together; it is also a way to look at the problem from a listener-oriented viewpoint, and we claim such a position is key to identify interesting, grounded, meaningful new problems. We hope that this project can motivate other projects of this kind in the future.

ACKNOWLEDGMENTS

This article is an account of work performed by the whole SONY CSL music team: Benoit Carré, Emmanuel Deruty, Fiammetta Ghedini, Gaétan Hadjeres, Marco Marchini, Dani Martín, Timothée Neullas, Alexandre Papadopoulos, Mathieu Ramona, Pierre Roy, and Jason Sakellariou, with the contribution of Anne Régnier (oboe). We also thank Giordano Cabral and Geber Ramalho for their active participation in the Brazyle project.

REFERENCES

K. Beck. 2000. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman Publishing Co., Boston, MA.

N. Boulanger-Lewandowski, P. Vincent, and Y. Bengio. 2012. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*, 1159–1166.

C. H. Chuan and E. Chew. 2007. A hybrid system for automatic generation of style-specific accompaniment. In *4th International Joint Workshop on Computational Creativity*. Association for Computing Machinery, London.

D. Cope. 2005. *Computer Models of Musical Creativity*. MIT Press.

E. Deruty and F. Pachet. 2015. The mir perspective on the evolution of dynamics in mainstream music. In *16th International Society for Music Information Retrieval Conference (ISMIR'15)*. 727–727.

K. Ebcioglu. 1990. Special issue: Logic programming applications an expert system for harmonizing chorales in the style of J.S. Bach. *Journal of Logic Programming* 8, 1 (1990), 145–185.

M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, and E. Aurell. 2013. Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E* 87, 1 (Jan. 2013), 012707.

J. D. Fernández and F. J. Vico. 2013. AI methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research* 48 (2013), 513–582.

M. Granroth-Wilding and M. Steedman. 2014. A robust parser-interpreter for jazz chord sequences. *Journal of New Music Research* 43, 4 (2014), 355–374.

G. Hadjeres, J. Sakellariou, and F. Pachet. 2016. Style Imitation and Chord Invention in Polyphonic Music with Exponential Families. http://arxiv.org/abs/1609.05152.

L. Hiller and L. Issacson. 1959. *Experimental Music: Composition with an Electronic Computer*. McGraw-Hill.

T. Jobim. 1967. Wave. A&M Records. (1967).

B. Kehew and K. Ryan. 2006. *Recording the Beatles*. Curvebender Publishing.

T. R. Lezon, J. R. Banavar, M. Cieplak, A. Maritan, and N. V. Fedoroff. 2006. Using the principle of entropy maximization to infer genetic interaction networks from gene expression patterns. *PNAS* 103, 50 (2006), 19033–19038.

E. Maestre, R. Ramírez, S. Kersten, and X. Serra. 2009. Expressive concatenative synthesis by reusing samples from real performance recordings. *Computer Music Journal* 33, 4 (Dec. 2009), 23–42. DOI:http://dx.doi.org/10.1162/comj.2009.33.4.23

G. Milner. 2009. *Perfecting Sound Forever: An Aural History of Recorded Music*. Faber & Faber.

F. Pachet. 2012. Musical virtuosity and creativity. In *Computers and Creativity*, J. McCormack and M. D'Inverno (Eds.). Springer.

F. Pachet. 2015. *Flow-Machines: CP Techniques to Model Style in Music and Text.* Association for Constraint Programming. (September 2015).

F. Pachet and G. Cabral. 2013. Why is Brazilian Guitar Interesting? ISMIR tutorial. (2013). https://www.youtube.com/watch?v=Zhv7HaTcDkE.

F. Pachet and P. Roy. 2014. Non-conformant harmonization: The real book in the style of take 6. In *5th International Conference on Computational Creativity (ICCC'14)*. 100–107.

F. Pachet, J. Suzda, and D. Martín. 2013. A comprehensive online database of machine-readable lead sheets for jazz standards. In *14th International Society for Music Information Retrieval Conference (ISMIR'13)*. 275–280.

A. Papadopoulos, F. Pachet, P. Roy, and J. Sakellariou. 2015. Exact sampling for regular and Markov constraints with belief propagation. In *Proceedings of the 21st Principles and Practice of Constraint Programming Conference (CP'15)*.

R. B. Potts. 1952. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society* 48, 1 (1952), 106–109.

M. Ramona, G. Cabral, and F. Pachet. 2015. Capturing a musician's groove: Generation of realistic accompaniments from single song recordings. In *24th International Joint Conference on Artificial Intelligence (IJCAI'15)*. 4140–4142.

P. Roy and F. Pachet. 2013. Enforcing meter in finite-length Markov sequences. In *27th Conference on Artificial Intelligence (AAAI'13)*.

P. Roy, G. Perez, A. Papadopoulos, J.-C. Régin, and F. Pachet. 2016. Enforcing structure on temporal sequences: The Allen constraint. *Proc. of Principles and Practice of Constraint Programming - 22nd International Conference (CP'16)*. 786–801.

J. Sakellariou, F. Tria, V. Loreto, and F. Pachet. 2015. Maximum entropy model for melodic patterns. In *ICML Workshop on Constructive Machine Learning*.

I. Simon, D. Morris, and S. Basu. 2008. MySong: Automatic accompaniment generation for vocal melodies. In *Conference on Human Factors in Computing Systems (CHI'08)*. Association for Computing Machinery. http://research.microsoft.com/apps/pubs/default.aspx?id=64277.

S. W. Smoliar. 1991. Algorithms for musical composition: A question of granularity. *Computer* 24, 7 (July 1991), 54–56. DOI:http://dx.doi.org/10.1109/2.84836

J. W. Ulrich. 1977. The analysis and synthesis of jazz by computer. In *Proceedings of International Joint Conference on Artificial Intelligence*. 865–872.

G. Widmer. 2016. Getting closer to the essence of music: The con espressione manifesto. *ACM Transactions on Intelligent Systems and Technology, Special Issue on Intelligent Music Systems and Applications* (2016).

Received November 2015; revised January 2016; accepted February 2016