



HAL
open science

Musique Lab 2: From computer-aided composition to music education

Jean Bresson, Fabrice Guédy, Gérard Assayag

► **To cite this version:**

Jean Bresson, Fabrice Guédy, Gérard Assayag. Musique Lab 2: From computer-aided composition to music education. *Journal of Music, Technology and Education*, 2013, 5 (3), pp.273-291. 10.1386/jmte.5.3.273_1 . hal-01159540

HAL Id: hal-01159540

<https://hal.science/hal-01159540>

Submitted on 28 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Published in: Journal of Music, Technology & Education. 5(3), 2013. Intellect, Ltd.

Musique Lab 2: From computer-aided composition to music education¹

Jean Bresson, Fabrice Guédy, Gérard Assayag

Abstract

We present Musique Lab 2, an application for music education based and inspired from computer-aided composition technology. A general contextualization and overview of the software is given as well as several specific examples illustrating possible uses and musical domains covered by the application.

1. Introduction: Music – Computer – Education

The use of new technologies for music education reflects contemporary trends and practices in music composition. However, most of the recent projects bridging computer music technology and education tend to seek technological advances in the fields of information access or multimedia integration by focusing on problems such as interactive/collaborative distant networking, interactive listening (augmented with multimedia information), or instrumental practice teaching based on sound or motion capture. No much work has been carried out so far to consider musical concerns underlying the use of computer and calculus in composition. Honing (1993) applied and extended the "micro-worlds" concepts for composition and musical data processing.² Since then however, very few research and projects have considered programmable environments for music education.

Musique Lab is a project carried out at IRCAM in partnership with the French Ministry of Education with the objective of developing pedagogical tools leveraging recent computer music technology and adapted to the contemporary music creation practice. A first software suite, released in 2002, allowed for a quick and intuitive immersion in the world of computer music and helped familiarize students with elementary musical and acoustic concepts such as pitch, scales, rhythm, dynamics, and the like. A few years later, a new phase was engaged with *Musique Lab 2*, and several new guiding directives were specified in collaboration with secondary school and conservatory music teachers. These directives included for instance the possibility to represent and manipulate the different parameters of sound and music, high-level musical structures, tonal knowledge, or the emphasis of relationships between composition and acoustics, or between music and science in general.

Among the different efforts initiated for this purpose (Guédy 2006), the application presented in this article has been conceived and based on the OpenMusic computer-aided composition environment (Agon 1998; Assayag et al. 1999; Bresson et al. 2009). In contrast with sound-oriented computer music research and independently from real-time performance issues, computer-aided composition as meant in this context mostly focuses on symbolic

¹ This article is partly adapted and translated from "Musique Lab Maquette : approche interactive des processus compositionnels pour la pédagogie musicale" (J. Bresson, F. Guédy, G. Assayag), *Revue STICEF*, Vol. 13, 2006.

² Micro-worlds are basic programmable environments with simple syntax allowing exploration and experimentation within a specific domain and using few of its main constitutive elements. The *Logo* system is a famous and notable example of a micro-world.

musical information and the simultaneous consideration of musical and computer processes through the use of programming languages (Assayag 1998). Following these principles, and after several generations of computer-aided composition systems, the OpenMusic environment has been created as a complete visual programming framework in which musicians can build experimental situations in a wide range of musical domains through formalization and design of compositional processes. At the meeting point of computer sciences and music, the works carried out in this environment have been dealing with a broad diversity of problems related to programmability, music codification, constraints, dialectic between musical objects and processes, between signal and symbolic representations, and many more. In addition, OpenMusic introduced a new formal approach for the specification and manipulation of temporal structures with a hybrid object/programming interface called "maquette", which allows unfolding programs and data structures in a common graphical and temporal context. Throughout many years of use by composers (see Agon et al. 2006; Bresson et al. 2008), but also by musicologists and researchers, OpenMusic was enriched with a considerable base of knowledge and tools, which keeps in a continual extension.

This original approach of music devised and analyzed by computing, as well as the corresponding compositional practice and experience, are integrated as an educational framework in *Musique Lab 2*, where the main features of the environment have been turned into innovative and interactive pedagogical tools.

After an overview of the *Musique Lab 2* software, we will develop some perspectives on its use and application areas by emphasizing the relations linking this pedagogical project to fundamental concepts in computer-aided composition and contemporary music creation. Concrete thematic examples will be given for this purpose, along with the description of specific musical or technical features.

2. Musique Lab 2: The application

The *Musique Lab 2* software (*ML2*) is an extra level on the functional and graphical kernel of OpenMusic, which is itself based on the Common Lisp programming language and its object-oriented layer CLOS (Common Lisp Object System) (Steele 1998). The dynamic aspects of this language, which made it particularly adapted and appreciated in the field of computer music composition (see Dannenberg et al. 1997), as well as the object-oriented features provided by CLOS – inheritance, meta-object programming, etc. – allowed making *ML2* benefit from the existing and future objects and functionalities available in the computer-aided composition environment. Most of these features have been modified or extended with specific graphic interfaces, editors, and protocols adapted to the educational purpose.

Three main conceptual levels of the musical material are provided by OpenMusic, and guided the initial conception of the application: (1) Data structures adapted to symbolic musical representations and processing; (2) methods for the inspection, manipulation, and creation of these structures; and (3) protocols for the integration of these structures in a temporal context.

The *ML2* documents are versatile pedagogical supports used and manipulated interactively by the music teacher or his/her students. The main window (see Figure 1) is divided in 3 areas, respectively called "objects", "operators", and "maquette". The formalization of this window partition corresponds to the three approaches of the musical structures mentioned above, with the level of data structures (objects and materials), the level of the calculus (transformation and creation of objects), and the level of the global (temporal) organization.

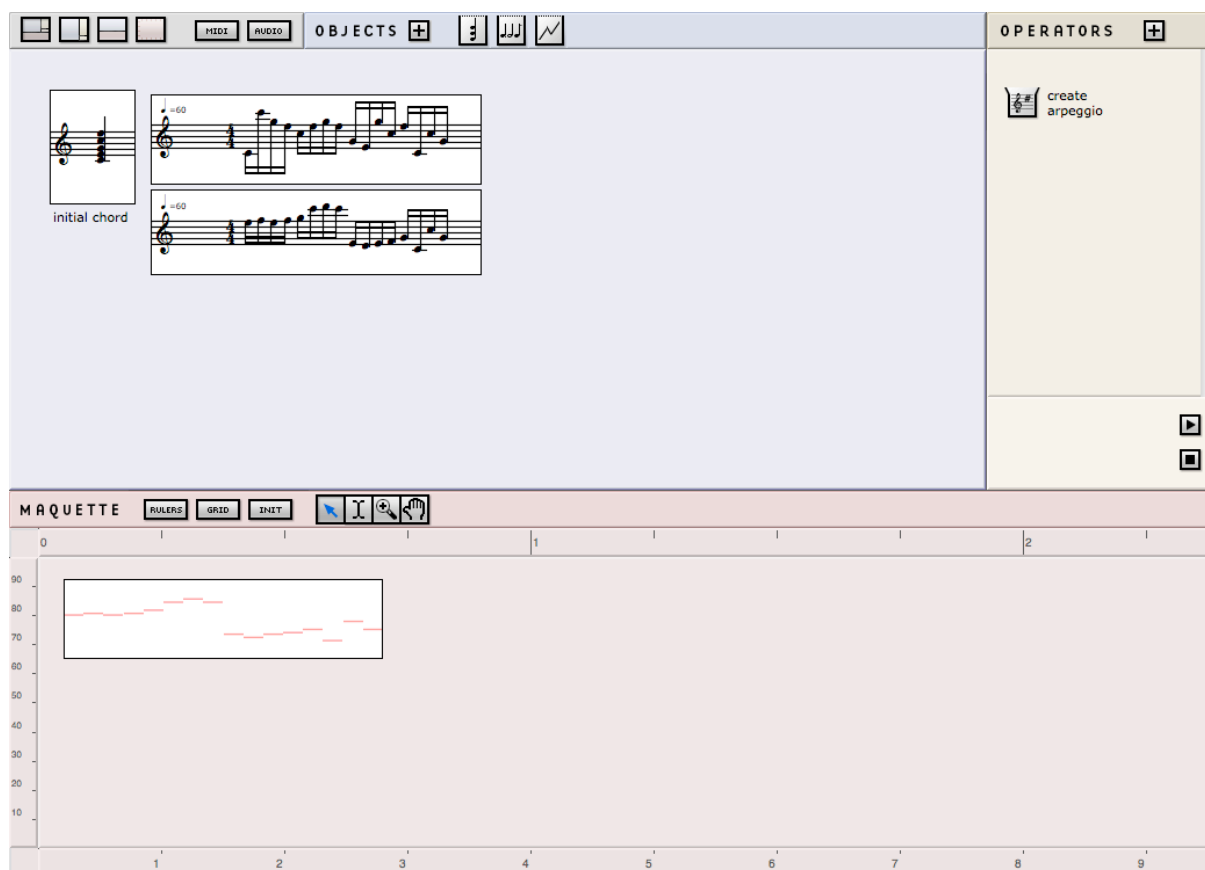


Figure 1. Example of a Musique Lab 2 document.

Most of the interaction is based on *drag-and-drop* operations: The successive steps in the "life" of musical objects (instantiation, construction, transformation, organization) are triggered by dragging icons and boxes on the document window.

2.1 Objects

The upper-left part of the window is the "objects" area. Different classes of musical objects (chords, sequences of chords or notes, rhythms, curves or "envelopes", sounds) can be instantiated in the main panel of this area. These objects are represented by a rectangular box on which is displayed a preview of their contents. They can also be edited using specialized graphical editors, as shows Figure 2.

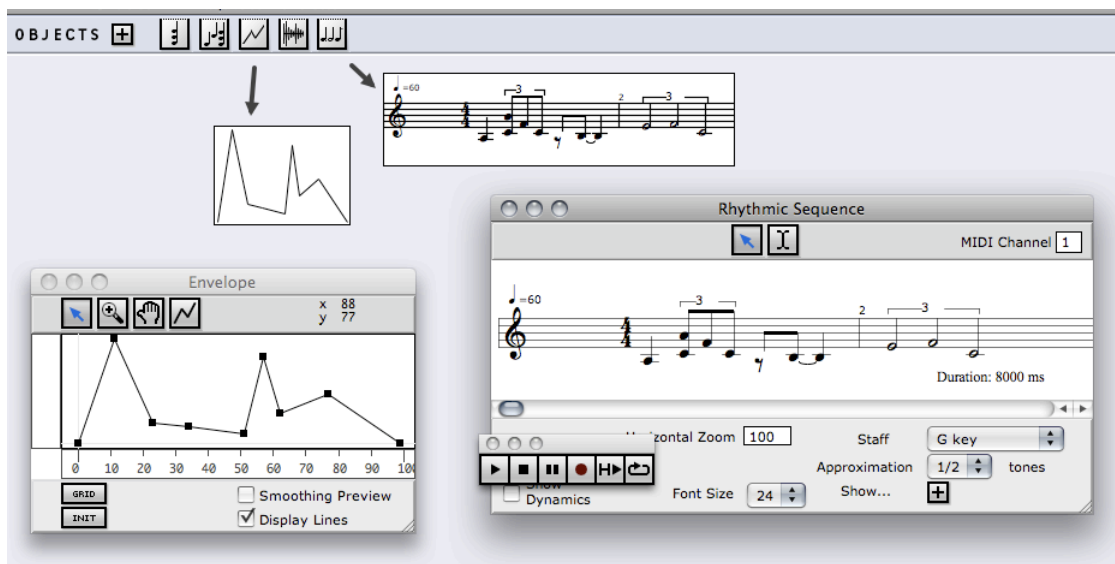


Figure 2. Instantiation of objects and graphical editors: envelope and rhythmic sequence.

2.2 Operators library

The right-hand part of the window is the "operators" area. It contains a list of operators which can be applied to the objects from the "objects" area. This list is created by selecting from a library of operators (see Figure 3).

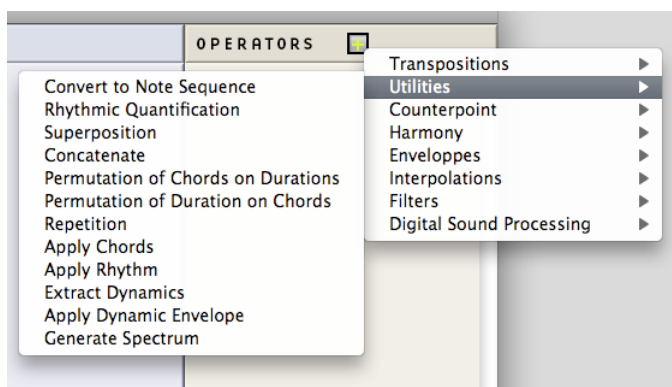
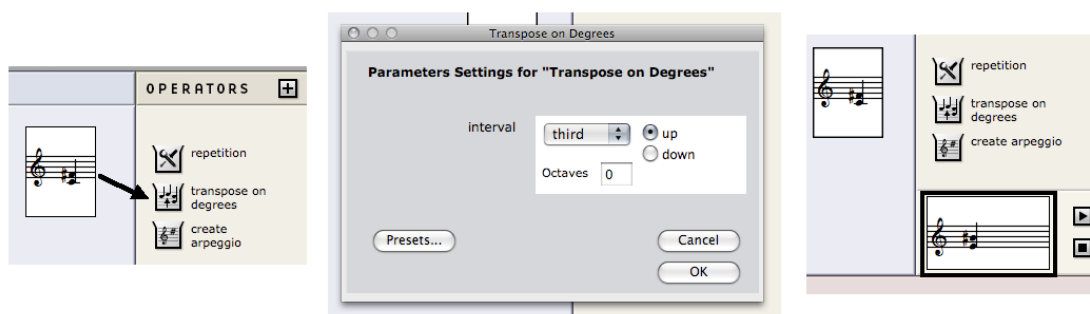


Figure 3. Selecting operators.

To apply an operator to an object, the object box must be dragged from the "objects" area onto the corresponding icon in the operators list. A dialog window appears, which allows the user to set the possible parameters of the operator. Once these parameters are set and confirmed, a new object is computed and a new box appears in the bottom part of the "operators" area (see Figure 4).



Figure

4. Applying an operator to a musical object: 1) the object (a chord) is dragged onto the operator (a transposition); 2) a dialog window allows entering the operator's parameters (interval of the transposition); 3) a new object is created.

The new box can then be dragged back into the "objects" area, where further modifications or operations can be performed: A cycle of transformation begins, allowing the progressive elaboration of complex and structured musical materials.

The operators of the library are grouped in several categories. These categories begin with general-purpose manipulations on musical structures, and then get specialized to specific musical or extra-musical domains such as tonality, counterpoint, curves manipulations, signal processing, and so forth. Examples of general-purpose utilities include for instance the concatenation, superimposition, the repetition of musical sequences, the application of a rhythmic pattern onto a melody or, conversely, of a melodic pattern onto a given rhythm, the conversions between proportional and rhythmic time notations, etc.

Figure 5 is an example of a chord interpolation process, which generates a sequence of chords going from a group of pitches to another in a pseudo-continuous way. Other examples of operators will be provided in the following sections.

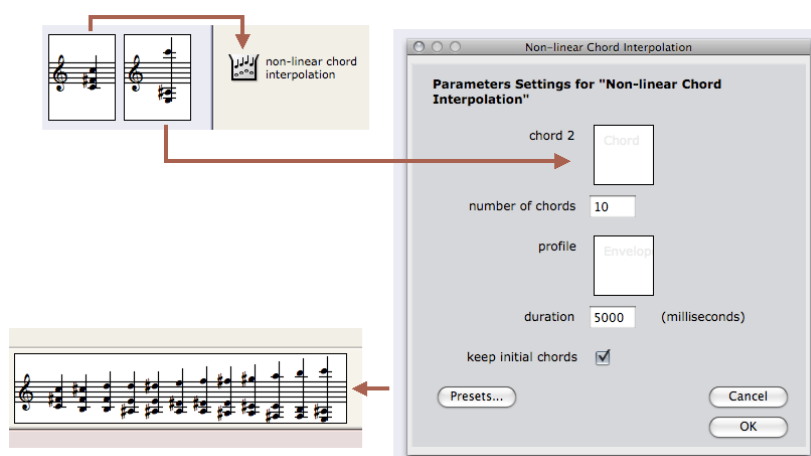


Figure 5. Chord interpolation. Note that the first chord is dragged onto the operator, while the second one and the optional interpolation profile are specified by dragging objects into corresponding "drop areas" of the parameters setting window.

The library of operators constitutes a reservoir of tools and building blocks for the elaboration of pedagogical situations using ML2. Some operators are adaptations of tools previously developed in the OpenMusic environment, for compositional or analytical purposes. Other ones were created especially for this application. The extensibility of this library is a

fundamental aspect of the project, which will make it possible to address new applications fields, musical formalisms and aesthetics. Since both environments share the same language and the same data structures, the tools developed in OpenMusic can easily be adapted and used as operators in ML2. A simple protocol exists allowing to create and add new operators starting from functions written in Lisp or in OpenMusic (Bresson 2007).

2.3 Maquette and "potential objects"

Finally, the bottom part of the window is called "maquette", in reference to the OpenMusic interface from which it is inspired. The horizontal axis in this area represents time, so that it can be used as a montage environment where the objects from the upper area are put together and sequenced in time. Therefore, the *maquette* permits the global organization of the structures created in the previous phases: The separation of this concept from that of the actual creation of the musical materials reflects a common compositional approach – see for example "in-time" vs. "out-of-time" in (Xenakis 1992) – and allows focusing individually on these different aspects.

Musical objects are dragged into the *maquette* from the "objects" area and are then represented by another kind of rectangular box using a "piano-roll" display (like in standard MIDI sequencers – see Figure 1). Specific graphical attributes can be assigned and displayed on the different boxes, such as a name, a background colour or a picture.

The horizontal position of the boxes in the "maquette" area represents the object's onset in the global time referential, and their horizontal extent determine durations (the objects can be stretched by changing the box size). In addition to these temporal properties, the boxes' graphical attributes can also hold a functional meaning: If set accordingly, their vertical position and extent, as well as the corresponding modifications by user actions, can modify the musical objects they represent. In a first stage, these values can control the amplitude level of the object (e.g., the higher the position of the box on the vertical axis, the higher the level of the object inside), or its MIDI output channel (in order to change the timbre dynamically). Each time the box is moved or resized, then, the musical object can be modified, depending on a specified correspondence between the object's attributes and the *maquette's* vertical ruler units.

Vertical size and position can also be used to specify more advanced transformations on the objects. An object dragged into the "maquette" area keeps in memory some information about the operator from which it was created (if it was created by an operator), and about the corresponding initial object and parameters. Hence, this object has all the information needed for computing itself again, possibly with a modification of one or more of these parameters. The graphical properties of the box (position, size) can therefore be used here again and hold a new functional dimension. To set such behaviour to a box, the user must also specify a correspondence between the range of a parameter value and the *maquette* ruler units. Each time the box is moved or resized, the object will then be recomputed and will take a new form. Figure 6 shows this process applied with a chord associated to a transposition, where the transposition interval changes with the vertical position of the box.

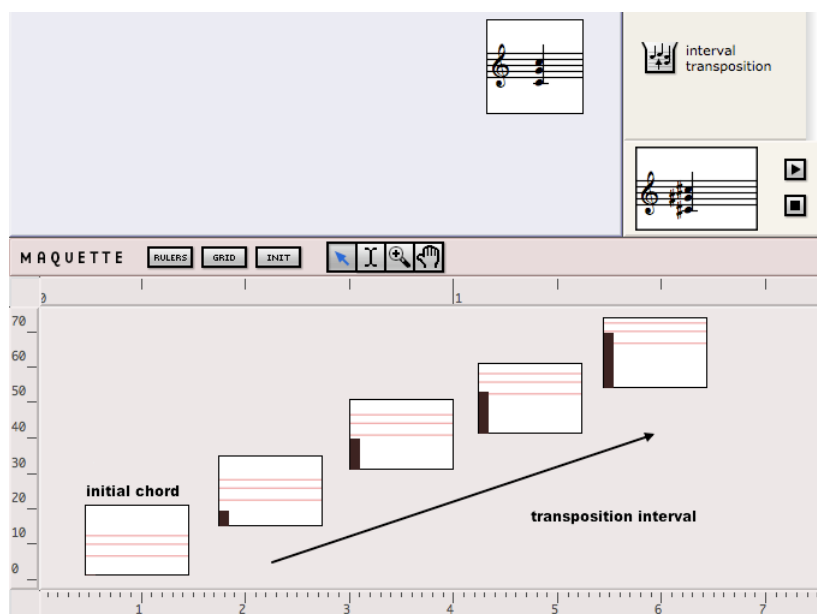


Figure 6. *Dynamic object computation in the maquette area: duplicate boxes are transposed according to their vertical position.*

Therefore, several instances of a same "potential object" (copies of a unique initial box) can have different effective values depending on their position and/or size in the maquette area. Considering the objects as potentials of musical realizations rather than fixed structures can be useful and pertinent in teaching situations and allows for rich experimentations on the influences and effects of the different musical parameters.

3. Applications

In this section we present some examples and pedagogical situations focusing on specific musical domains and making use of dedicated groups of operators.

3.1 A preliminary example: Experimenting with arpeggios

As a first example, let us apply the procedure outlined in the previous section and illustrate a pedagogical approach of the arpeggio structures. This example corresponds to the document visible in Figure 1.

The *arpeggio* operator is a function that creates an arpeggio starting from a chord, following a rhythmic specification, a pitch interval and probabilities for the occurrence of passing tones or neighbour notes (see Figure 7).

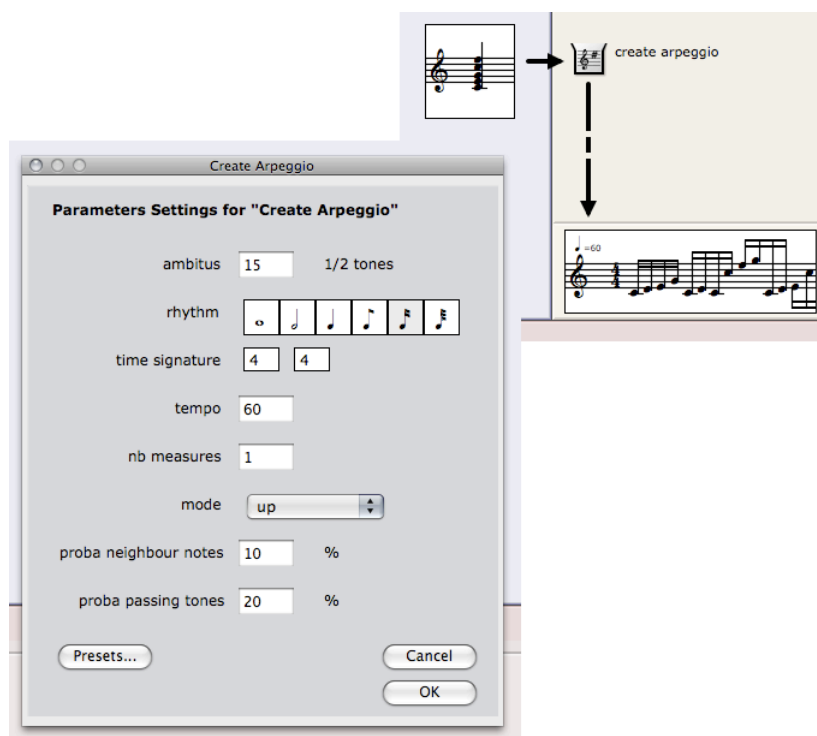


Figure 7. *The arpeggio operator.*

Starting from chords, possibly prepared beforehand by the teacher, the pupils can create arpeggios by dragging them onto the operator. Different configurations of the parameters settings permit productions of particular styles, while keeping diversity in the results thanks to probabilistic aspects. Figure 1, for instance, showed two different versions of an arpeggio created with the same chord by changing the probability of neighbour and passing tones.

The arpeggios can then be dragged into the maquette, and parameterized in order to make the parameters used for their construction depend on graphical properties. In Figure 8 the parameters of the arpeggio are set in order to make its pitch range depend on the vertical position of the box. The box is duplicated on the temporal surface, with different vertical positions, which produce the corresponding arpeggio realizations.

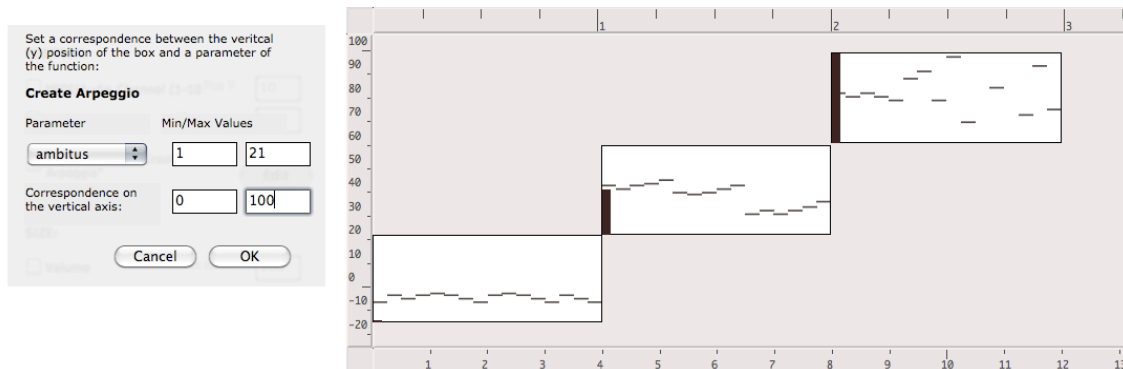


Figure 8. *Making the arpeggio parameters depend on the box graphical attributes.*

Figure 9 illustrates a complete teaching session centred on Mozart sonatas and using this operator: A series of chords is extracted from an initial sequence, and each one of them is used and processed as previously to create an arpeggio. The chords played in arpeggios are then ordered in the maquette as an accompaniment of the original theme.

The image shows a software interface for music analysis. The top part, titled "FORME SONATE", features a musical score for "THEME A" by W. A. Mozart. Below the score, there are several musical elements represented as boxes, some with chord symbols like "I", "V", "II", "VI", and "v 6 4". To the right, there is a panel labeled "OPERATORS" with two icons: "create arpeggio" and "interval transposition". The bottom part of the interface is labeled "MAQUETTE" and contains a graphical representation of the sonata form structure. It has a horizontal axis from 0 to 8 and a vertical axis from 20 to 120. The structure is shown as a series of boxes of varying heights and widths, representing different sections of the sonata form.

Figure 9. A document constructed for the study of the sonata form using a theme from W. A. Mozart and the arpeggio operator.

This example shows how a simple operator can be used to explore a wide musical domain. The sonata simulation generated can also be experimented with various configurations of the different elements (theme chords, arpeggios) and instrumentations (by assigning different instrument timbres to these different elements).

3.2 Curves and envelopes: A continuous approach to the musical parameters

The "envelope" objects are used to represent the evolution of a musical parameter or value. A dedicated group of operators allow for the generation and transformation of these objects (using sampling, smoothing and other procedures) or to use them to set some properties of other types of objects (pitch profiles, dynamics, etc.) For example, a melodic envelope can be extracted from a sequence, re-sampled with a given sample rate, eventually converted into a new note sequence. Figures 10 and 11 show examples of the use of these operators for the generation of melodic profiles.

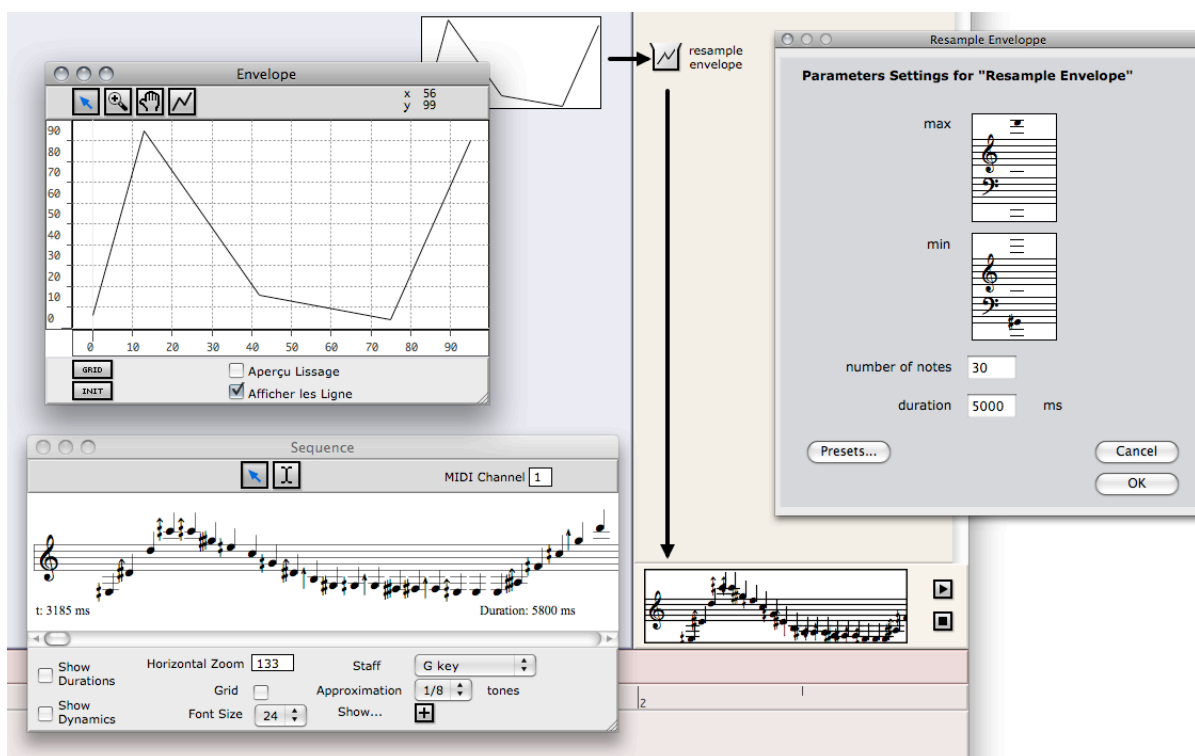


Figure 10. *Creating a sequence of notes by sampling an envelope.*

This "continuous" conception of the musical parameters is powered in the pitch field by the availability of microtonal scales for the visualization and rendering of musical sequences (see score editor in Figure 10). With the creation of a sequence of notes starting from a 2D B-Spline curve algorithm, the example of Figure 11 also illustrates the possibility to introduce mathematical concepts and parameters in the musical operations.

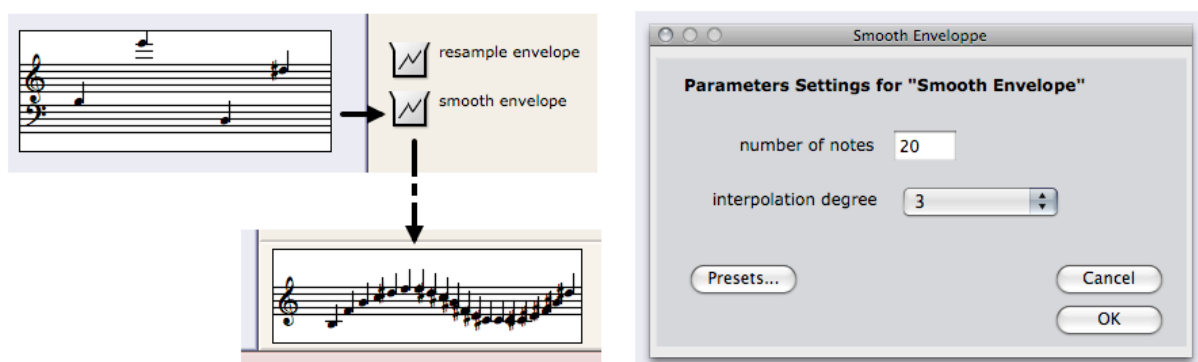


Figure 11. *Creating a sequence of notes from notes and B-spline curve computing ("smooth envelope").*

Envelopes can also be used as parameters in other operators such as transpositions or interpolations. In Figure 12 an envelope is used to create a sequence by transposing a chord following a given profile.

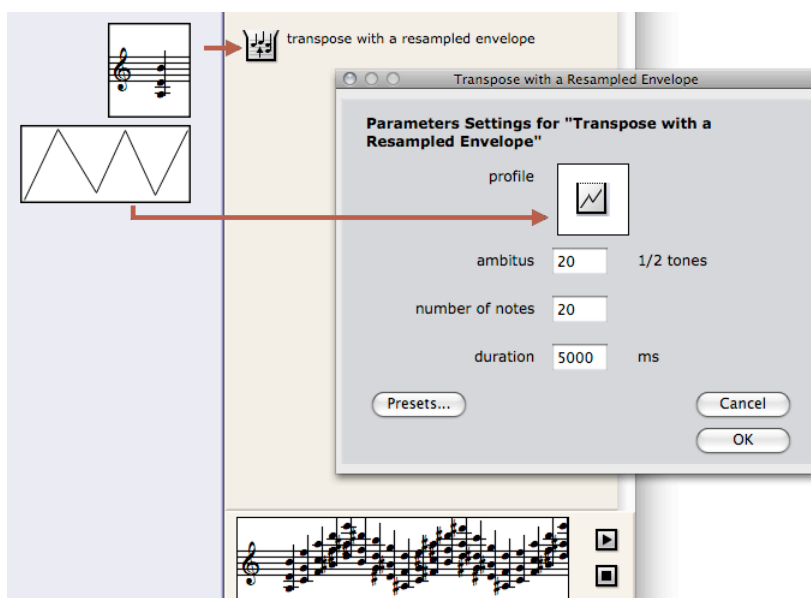


Figure 12. *Transposing using an envelope.*

Figure 13 shows a document created for a lesson on Philippe Leroux's piece *VOI(REX)*. Hand-drawn or computer-generated curves are used to create melodic forms, which are combined to reconstitute the original compositional processes and some fragments of the piece.

Figure 13. *A study of VOI(REX) (Ph. Leroux) in Musique Lab 2.*

3.3 Tonality: Computational models for the study of early/classical music

The major part of the repertoire taught in secondary schools and conservatories is situated in the tonal period of the music history. In order to undergo manipulations in the tonal context, the OpenMusic musical objects have been extended with a hierarchical model of tonality: Every sequence can have a tonality, and each chord inside this sequence can either have its own tonality (which represents a tonal modulation), or inherit the tonality from its containing sequence. The tonal information can be set for the different musical objects in the score editors, and is then considered both at the level of the graphical representation (in these same editors) and during objects processing by "tonal" operators. Indeed we have already seen in the previous sections some examples of operators requiring tonal information in order to process the musical objects, such as the transposition on degrees (Figure 4) or the arpeggio (Figures 7-9). Figure 14 is an other example with the transposition of an object from a tonality to another one.

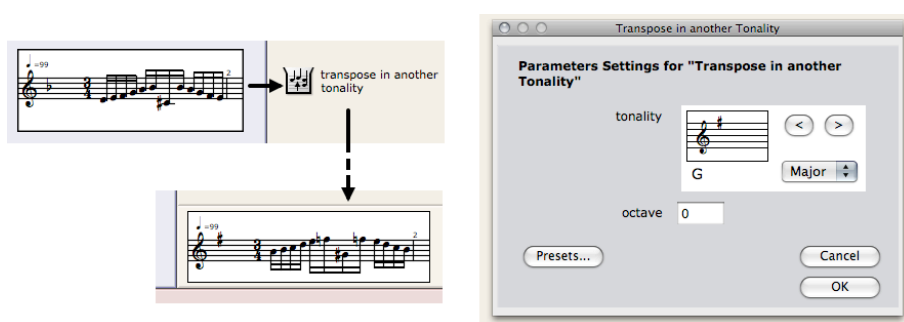


Figure 14. *Transposition of a sequence from a tonality (D minor) to another tonality (G major). This operation is complementary to the transposition on degrees illustrated in Figure 5, where the transposition was specified by an interval in a same tonality. The present example is closer to the concept of a tonal modulation.*

Other tonal operators allow to search chord inversions, or rhythmic canons. The "counterpoint" operators create contrary and/or retrograde movements, following the tonal or serial music models. Figure 15 shows a document created for a music lesson on J.-S. Bach's *Goldberg Variations*. In this example a complete variation is reproduced using the tonal transposition and the counterpoint operators.

J.S. Bach
Variations Goldberg
Fabrice Guédry - IRCAM
version de travail 24/03/09

Variation 12 : canon à la quarte

Enregistrement de référence (Glenn Gould)

Tempo = 100

Initial sequence

Basse de l'aria

sequence initiale quantifiée

MODE D'EMPLOI

- 1) transposition à la quarte (quarte inferieure)
- 2) mouvement contraire (outil "inversion")
- 3) transposition 1 degré - 1 octave
- 4) quantifier l'antécédent
- 5) mouvement contraire obtenu sur permutation d'accords
- 6) et voilà, faire glisser le résultat dans la maquettes.

permutation sur les durées de la sequence initiale

OPERATORS

- retrograde
- contrary
- transpose on degrees
- permutation of chords on durations
- rhythmic quantification

MAQUETTE

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Figure 15. A study of J.-S. Bach's Goldberg Variations.

3.4 Digital sound processing and electro-acoustic music

The latest developments in ML2 have been oriented toward the analysis and representation of sounds. A suite of sound processing operators allow the extraction of symbolic description data from sound objects, or the creation of sounds starting from symbolic data. These operators are based on the IRCAM sound processing kernel *pm2* and adapted from previous works carried out in OpenMusic for the integration of sound analysis and synthesis in the compositional environment (Bresson 2006). Figure 16 illustrates the use of the "partial tracking analysis" operator, which detects and follows the evolution of the main partials in a sound using *pm2*, and converts them into a chords sequence (Hannape and Assayag, 1998).

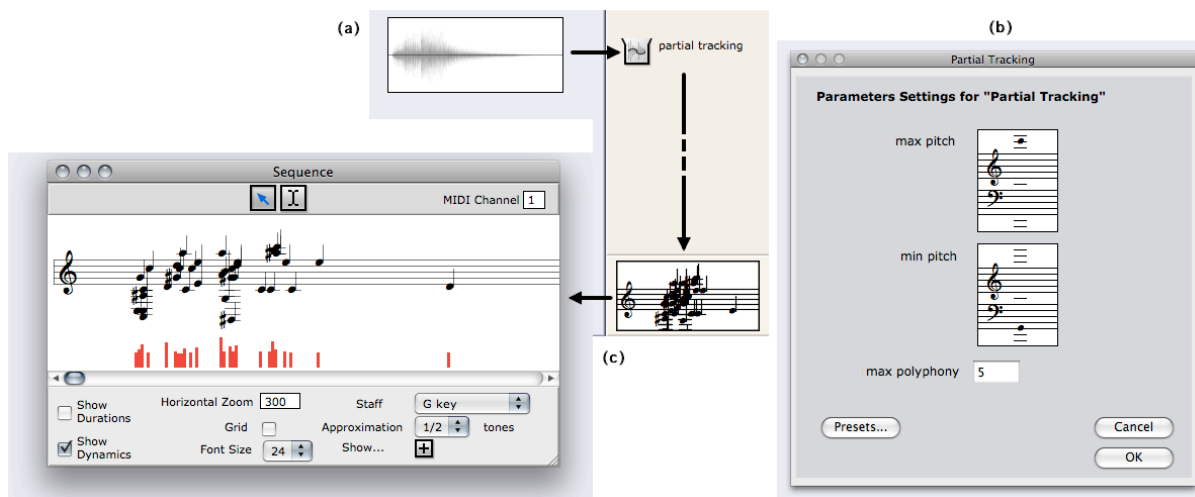


Figure 16. *Partial tracking analysis of a sound. (a) Initial sound, (b) analysis parameters, and (c) new sequence created from the sound analysis.*

Among the other sound processing operators available are also the "chord-sequence" analysis, which uses a preliminary segmentation of the original sound in order to perform a more structured sound analysis, the additive synthesis, which synthesizes a sound from partials corresponding to the tones of a musical sequence, or the fundamental frequency estimation, which returns a time/frequency envelope from the analysis of a sound. All these operators are illustrated in Figure 17.

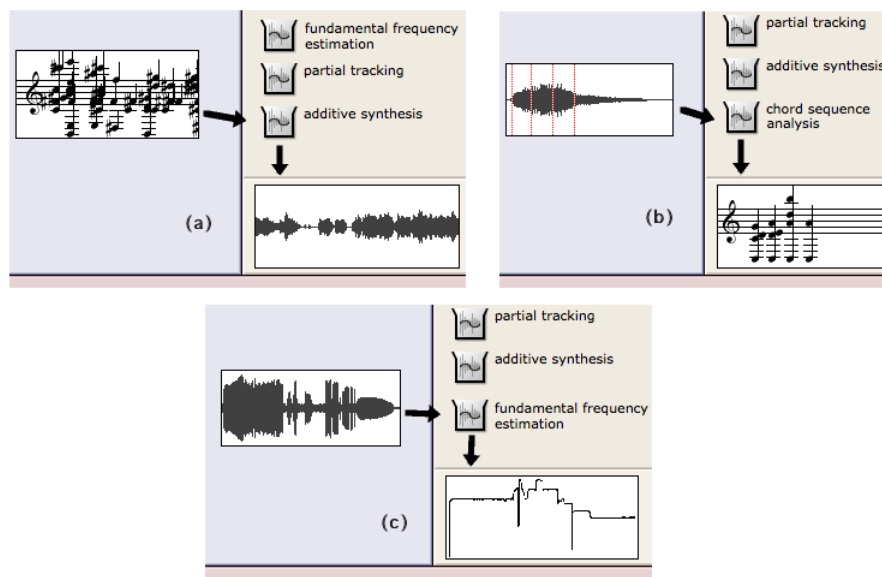


Figure 17. *Sound processing operators: (a) additive synthesis; (b) "chord-sequence" analysis; (c) fundamental frequency estimation.*

The use of sounds and sound descriptions in compositional works is a common practice in the 20th/21st century music, and the reconstitution of corresponding processes in pedagogical situations using elementary interactive procedures allows a better understanding and experience of their technical and musical meaning. Indeed, these sound analysis and synthesis tools naturally put forward fundamental notions of contemporary music such as the relations between acoustics and symbolic pitch representations, between natural recorded and synthesized sounds, or more generally between harmony and timbre. These notions, which require advanced signal and symbolic processing techniques in computer music systems, are a

few clicks away in ML2. Teachers can therefore easily demonstrate, and pupils experiment with these concepts and create their own sounds and material following the principles and inspired by particular compositional processes and works.

Studies and workshops about spectral music have been carried out at IRCAM using the signal processing tools in ML2,³ focusing on works such as Gérard Grisey's *Partiels*, Tristan Murail's *Désintégrations* or Xu Yi's *Le plein du vide* (in the music track programme of the French Baccalaureate in 2006 and 2007). Figure 18 shows a document created by a pupil during one of these workshops.

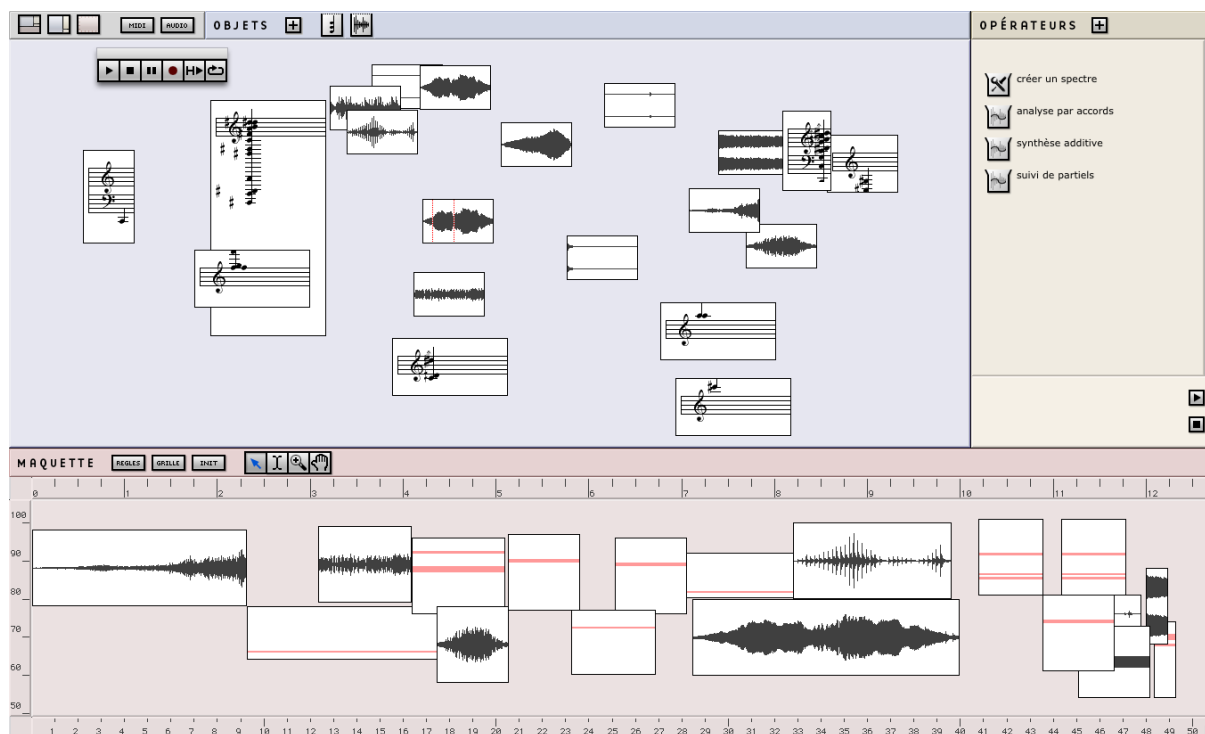


Figure 18. *Spectral music in Musique Lab 2: A pupil's work.*

3.5 Merging territories

To conclude with the different groups of operators, it is important underline the possibility and interest of associating them to one another in common documents and projects. Indeed, it quickly becomes natural in this environment to combine the different tools and operators available, which leads to original and unique pedagogical scenarios. For example, the sound analysis operators can be used in relation to the interpolation and envelope operators, in order to generate and synthesize timbre interpolations following hand-drawn trajectories. Harmonic constraints can also be added to these interpolation processes, emphasizing again the links between sound and symbolic compositional issues (see Figure 19).

³ Workshops for secondary school students, supervised by Grégoire Lorieux, IRCAM, Paris.

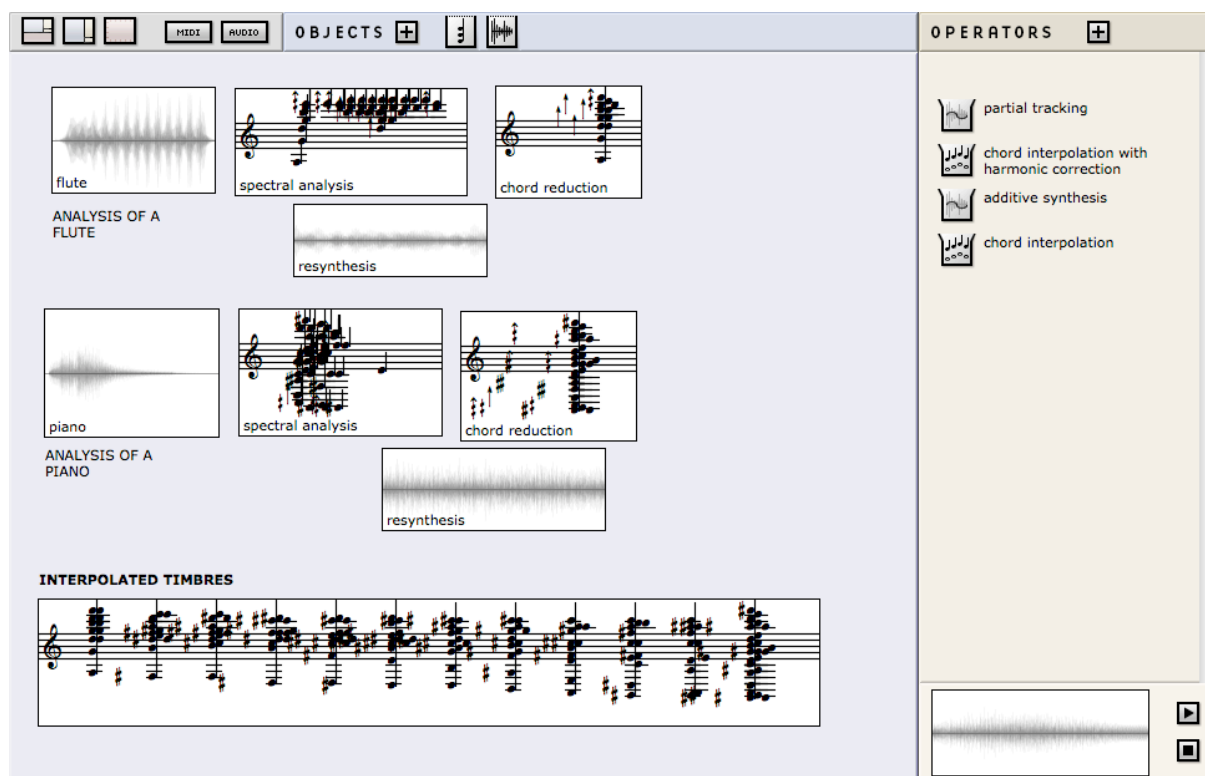


Figure 19. *Timbre interpolations.*

5. Pedagogic use and scenarios

As shown in the previous section, ML2 can address a wide range of topics related to varied approaches to music history, analysis, compositional techniques or aesthetics. From a pedagogical point of view, the experimental and interactive aspect of the application are likely to stimulate the interest and creativity of the music students by re-enacting compositional attitudes.

The first experiments carried out in pedagogical situations (workshops, music classes) emphasized several archetypal use-cases for this application. The simplest one is to use ML2 as a dynamic and multimodal presentation tool, where documents are projected in the classroom as "augmented blackboards". Easy and interactive data generation, transformation, visualization and playback features allow developing dynamic lessons with multiple possible orientations. A more interactive use, which generally takes place as a second phase, has the pupils working individually or in small groups on computer stations, and manipulating documents prepared by their teacher or built in interaction with him/her.

Figure 20 is a ML2 document illustrating similarities between Aka Pygmy traditional music and a prelude by J.-S. Bach, made for pupils from a music school.⁴ In this document is implemented the model of "rhythmic imparity" as described by Arom (2004), used as a starting point to progressively transform Pygmies' rhythmic patterns into the prelude score, or conversely, in other cases, to extract rhythmic imparities from Bach's score and reconstitute the Aka Pygmy polyrhythmic patterns. Several documents of this kind have been created, which reconstitute other pieces or fragments related to varied musical formalisms and aesthetics (see Guédy, 2006).

⁴ *Atelier des feillantines* (school of music), Paris, France.

Figure 20. *Ethnomusicology and the study of similarities between a Bach's prelude from the Well Tempered Clavier and Aka Pygmy rhythmic patterns.*

Finally, another possible use of ML2 concerns more creative projects, where pupils are encouraged to generate musical structures and sequences by their own, using pre-established or freely chosen sets of objects and operators. Figure 21 shows pupils working on a project about electronic music centred on the *Forbidden Planet* science fiction movie soundtrack (F. Wilcox, 1956, music by L. & B. Barron), and a document produced in this context.⁵ Envelopes are designed and applied to various musical parameters (pitches, velocities) in order to create sounds and sequences inspired from the movie and its original soundtrack.

In these situations where students and pupils directly work with ML2, the pedagogical documents turned out to be practical "musical desktops" making it possible to store and keep track of the state and progress of their work.

⁵ This workshop has been carried out with about a hundred students from different junior high schools in Paris (average 12-14 years old).

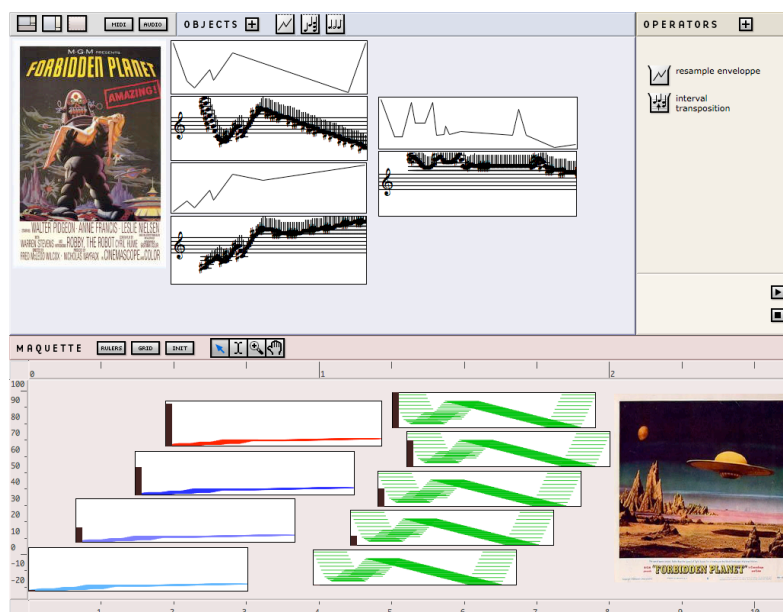


Figure 21. Pupils working on a creative project around the *Forbidden Planet* movie soundtrack (Collège Lavoisier, Paris, France).

6. Conclusion

We presented a new application for music education based on compositional concepts and the potentials of a computer-aided composition system. The musical structures representation and computation possibilities powered by this application, associated to easy and intuitive operating procedures, allow to re-explore and improve the objectives of the early micro-worlds using state-of-the-art computer-aided composition technologies.

As compared to other existing musical/pedagogical environments, the specificity of ML2 principally lays in its computational aspects, which enable connecting the symbolic representations to versatile, interactive generation and transformation of the musical material following pedagogic or creative purposes.

Following the objectives at the origins of the project, this unusual bridge between computer-aided music composition and education allows novel pedagogical assistance in reconstituting musical works starting from their constituting elements, working on specific notions abstracted from particular pieces, aesthetics or periods, or connecting the signal (sound) and symbolic worlds (a recurring topic in new music research targeted by the French Ministry of Education when commissioning the project). These representations and procedures can be used in various contexts and expertise levels, from basic illustrations to advanced organisations and formal frameworks.

In order to avoid the somehow easy way to consider musical complexity without really understanding it, an effort was made to multiply relatively simple and elementary operators to be combined with one another rather than complex ones with multiple options and parameters. No predefined scenario is suggested by the application, and each situation represents a specific progressive approach, which requires effective pedagogical support. We therefore consider ML2 a "pedagogical assistance" tool rather than a pedagogical software. For the same reason, it is important to provide users (i.e., pupils but also music teachers) with pre-established situations to explore in order to progressively initiate an autonomous interaction. Support documents have been created for this purpose and distributed complementarily to the application. These documents shall then be personalised by the teachers themselves according to particular pedagogical orientations or musical knowledge levels, shared between teachers and/or distributed to the students.

The possibility to create new operators (eventually by third-party users and developers) is also another important issue for enlarging the potentialities and scope of the application. The conception and development of these operators will probably constitute the most relevant part of the future work on this project.

Musique Lab 2 is currently available in French and English languages. It is distributed to the teachers of the French Ministry of Education via the *Educ'Net* network, and to the broader users community under IRCAM "Forum" licence.

References

- Agon, C. (1998) *OpenMusic: Un Langage Visuel pour la Composition Assistée par Ordinateur*. PhD Thesis, Université Pierre et Marie Curie, Paris, France.
- Agon, C., Assayag, G., Bresson, J. (Eds.) (2006) *The OM composer's Book 1*, Editions Delatour France / IRCAM Centre Pompidou.
- Arom, S. (2004) *African Polyphony and Polyrhythm. Musical Structure and Methodology*, Cambridge University Press.
- Assayag, G. (1998) 'Computer Assisted Composition Today', *1st Symposium on Music and Computer*, Corfu, Greece.
- Assayag, G., Rueda, C., Laurson, M., Agon, C., Delerue, O. (1999) 'Computer Assisted Composition at Ircam: PatchWork & OpenMusic', *Computer Music Journal*, 23:3, pp. 59-72.
- Bresson, J. (2007) 'Processus compositionnels et opérateurs musicaux dans ML-Maquette. Les outils de traitement du signal', *Actes des Journées d'Informatique Musicale*, Lyon, France, pp. 140-144.
- Bresson, J., Agon, C., Assayag, G. (2009) 'Visual Lisp/CLOS Programming in OpenMusic', *Higher-Order and Symbolic Computation*, 22:1, pp. 81-111.
- Bresson, J., Agon, C., Assayag, G. (Eds.) (2008) *The OM composer's Book 2*, Editions Delatour France / IRCAM Centre Pompidou.
- Bresson, J., Guédy, F., Assayag, G. (2006) 'Musique Lab Maquette : approche interactive des processus compositionnels pour la pédagogie musicale', *Revue STICEF*, Vol. 13, 2006.
- Dannenberg, R., Desain, P., Honing, H. (1997). 'Programming Language Design for Music', *In Roads, C. et al. (Eds.) Musical Signal Processing*, Swets and Zeitlinger, pp. 271-315.
- Guédy, F. (2006) 'Musique Lab 2, un environnement pour la pédagogie musicale', *L'Inouï*, 2, IRCAM - Editions Léo Scheer, pp. 149-153.
- Hanappe, P., Assayag, G. (1998) 'Intégration des représentations temps/fréquence et des représentations musicales symboliques', *In Chemiller, M., Pachet, F. (Eds.) Recherches et applications en informatique musicale*, Hermes, France, pp. 199-207.
- Honing, H. (1993) 'A microworld approach to the formalization of musical knowledge', *Computer and the Humanities*, 27, pp. 41-47.
- Steele, G. L. (1998) *Common LISP The language*, second edition. Digital Press.
- Xenakis, I. (1992) *Formalized Music: Thought and Mathematics in Composition*, Indiana University Press.