Computational Music Science

Series Editors

Guerino B. Mazzola
Moreno Andreatta

Gérard Milmeister

The Rubato Composer Music Software

Component-Based Implementation of a Functorial Concept Architecture

Springer

Dr. Gérard Milmeister
Langackerstrasse 49
8057 Zürich
Switzerland
gemi@bluewin.ch

Contributors:

Prof. Dr. Guerino B. Mazzola
164 Ferguson Hall
Minneapolis MN 55455
USA
mazzola@umn.edu

Florian Thalmann
Helmern 77l
6102 Malters
Switzerland

ISBN 978-3-642-00147-5 e-ISBN 978-3-642-00148-2

DOI 10.1007/978-3-642-00148-2

Library of Congress Control Number: 2009921145

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Foreword

Gérard Milmeister's thesis, which is now published in Springer's innovative series *Computational Music Science*, is a key text for accessing the present version of the RUBATO software. It is also the beautiful trace of a conceptual and technological completion of a development that was initiated in 1992, when Oliver Zahorka and I conceived, designed and implemented the RUBATO sofware for musical performance. This first implementation on NeXT computers, written in Objective C, was driven by the idea to implement Theodor W. Adorno's theory of an analytical performance, i.e., a performance that is defined upon musical analysis of the given score. The original architecture of RUBATO was therefore modular and split into modules for analysis (rhythmical, melody, and harmonic) and modules for performance. These modules, coined rubettes, were only conceived as organic parts of an overall anatomy. However, the successive developments and also research driven investigations, such as Anja Fleischer's work[1] on rhythmical analysis or Chantal Buteau's work[2] on motivic analysis showed that there is also a legitimate interest in rubettes without being necessarily parts of a given fixed anatomy. Successive work by Stefan Göller[3] on geometric concept spaces associated with RUBATO data formats, and Stefan Müller[4] on performance and gesture rubettes proved that there is a different approach to RUBATO, which by far transcends the original "hardcoded" anatomy.

[1] Fleischer, Anja. *Die analytische Interpretation: Schritte zur Erschließung eines Forschungsfeldes am Beispiel der Metrik*. Dissertation, Berlin 2003

[2] Buteau, Chantal. *A Topological Model of Motivic Structure and Analysis of Music: Theory and Operationalization*. Dissertation, Zürich 2003

[3] Göller, Stefan. *Object Oriented Rendering of Complex Abstract Data*. Dissertation, Zürich 2004

[4] Müller, Stefan. *Pianist's Hands—Synthesis of Musical Gestures*. Dissertation, Zürich 2004

The new requirements had to face different conceptual, soft-, and hardware conditions and challenges. To begin with, the NeXT computer had finished to exist, and the platform-dependent strategies, such as the original Objective C implementation had become obsolete by the now standard Java virtual platform environment. The other point of change was that the rubettes had to become a modular construct that would be of any size and would also be of open usage without much predefined larger anatomical preconditions. It turned out that the decisive requirements were defined by component-driven programming. However, this generic setup also entailed a radical redesign of the data format of denotators, which was invented in the early collaboration with Zahorka. The redesign was however not only affected by the component-driven data architecture, but by a meanwhile dramatic urge to step from naive (zero-addressed) denotators to functorial denotators, i.e., to quit the old-fashioned concept of a point and to introduce functorial points, i.e., morphisms defined on variable address modules and with values in not necessarily representable space functors.

All these delicate requirements set up an agenda that could not be realized except by a computer scientist with excellent programming and really solid mathematical competence. Gérard Milmeister was the ideal researcher to bring such a difficult enterprise to its completion. His award-winning doctoral dissertation, which is now in your hands, is the written counterpart of his remarkable programming work, available as a GPL software on `http://www.rubato.org`. The thesis is not only a clear and concise introduction to the conceptual and mathematical architecture of the RUBATO enterprise, but offers also precise and concrete tutorials for the programming of rubettes and their networks.

The success of Milmeister's work is, last, but not least, documented by contributions from Florian Thalmann and myself, which prove that the RUBATO software may now reach out to compositional tasks that were postponed since the early developments of a geometric composition software *presto* in the late 80s. Thalmann's *BigBang* rubette is the long-awaited extension of RUBATO to gestural strategies in composition, and it is the proof that Milmeister's work is not only the completion of a long march through the hard- and software meanders and conceptual revolutions, but also is setting a pointer to creative future music software perspectives.

Minnesota,
October 2008

Prof. Dr. Guerino Mazzola

Preface to the Springer Edition

For this edition published by Springer, I am happy to be able to include as chapter 17 and chapter 18 two contributions by Guerino Mazzola and Florian Thalmann. The first is the description of a sophisticated rubette that provides an extensive gestural interface to manipulate musical structures. The second contribution is the first major application of RUBATO COMPOSER in music theory and computational composition. It resulted in a remarkable piece of music starting from the idea of "analyse créatrice" forwarded by Pierre Boulez. The whole process involves many of the features presented in this book, and, thus, is something of a "proof by construction" of the usefulness of these concepts. I therefore thank both for their energy and ingenuity in putting the RUBATO COMPOSER system to test and exercising its capabilities.

Zurich,
December 2008

Gérard Milmeister

Preface

Trained as a computer scientist, I have always been interested in music and musicology. Therefore I took the opportunity offered by PD Dr. Guerino Mazzola to work with his MusicMedia group, then a part of the MultiMedia Laboratory at the Department of Informatics of the University of Zurich, directed by Prof. Dr. Peter Stucki.

Thus, first and foremost, thanks go to Guerino Mazzola, who introduced me to mathematical musical theory, most of which I had never heard of before. He gave me the conviction of working at the forefront of music research and taught me the use of modern mathematical methods, such as category theory. He supervised my work with all the enthusiasm and competence one could wish for.

I would also like to thank the reviewers Prof. Dr. Bernd Enders of the University of Osnabrück and Prof. Dr. Renato Pajarola of the University of Zurich, who suggested improvements to this thesis.

The thesis could not have been accomplished without the backing by Prof. Dr. Pfeifer, whom I like to thank for his willingness to support it as the responsible member of the Faculty of Mathematics and Natural Sciences.

Finally, I have to thank the staff of the Department of Informatics for helping me with the tedious administrative tasks that a doctoral student and assistant has to manage.

Zurich, *Gérard Milmeister*
November 2006

Introduction

It is significant that the art and theory of music production and performance have always been connected to the newest developments of the natural and engineering sciences of the time. Indeed, a sophisticated theory of sound and music has been an important part of the earliest mathematics in ancient Greece. The theory of musical intervals set forth by Pythagoras is still a matter of discussion among psychologists of music and theorists alike. On the other hand, since the appearance of digital computers, and the development of computer science as a mathematical and engineering discipline in the late 1940s and early 1950s, music has been among the first applications to appear besides numerical computations on the newly invented machines.

At lot has happened since, and there have always been researchers in mathematics who have been trying to apply the newest trends in their disciplines to the explanation of the principles of music, with various degrees of success. Whatever the outcome of each of these developments, the outlook of music as a whole has been changed forever to the open-minded observer. Unfortunately, it is still the case that the intersection of the set of mathematical music researchers and the set of musicologists or music theorists is vanishingly small compared to the combined number of people active in the field.

To further the penetration of mathematical music theory into the realm of music production, it is vital to offer a computer-based tool to contemporary composers and music theorists that provides the most advanced ideas from mathematics applied to music.

Category theory is the field of mathematics that has crept into almost every mathematical domain and reformulated most basic tenets in a modern language. Computer science is another discipline that has benefited enormously from the exposure to category theory, as has mathematical music theory. It is certainly not exaggerated to assert that the colossal volume *The Topos of Music* by Guerino Mazzola has brought music theory to a new

level by infusing it with advanced and recent ideas from category and topos theory, whence the name.

The following work takes the fundamental ideas expounded in that book and describes the design and implementation of a software system, called RUBATO COMPOSER, that provides the tools to work creatively and analytically with those principles. The software system is both an application development framework, targeted at programmers proficient in mathematics or music theory, or both, and a graphical user interface intended for the composer or the theorist who wants to make use of components created by developers to build an application that embodies his or her musical ideas.

The first part presents the concepts and theory. There is neither place nor need for a complete exposition of the theory developed in *The Topos of Music*. Therefore only those parts that have found their way into the implementation are discussed.

The second part is a thorough account of the implementation of the RUBATO COMPOSER software system. Here the high-level organization as well as some details are covered. This chapter is also of importance to the developer who wants to extend and build on the RUBATO framework.

The third part is about the practical aspects of using RUBATO COMPOSER. A tutorial describes a typical use through a step-by-step tour illustrated with screenshots of the running program. Several uses of the framework by external projects are introduced that exemplify the application developer aspect.

The fourth and last part acts as an appendix. The greatest part is taken up by the RUBATO COMPOSER user's manual. This manual is intended as a stand-alone reference and also features many details that are not essential to understanding and therefore are not included in the treatment in the main text.

Overview

Contents

Part I
Concepts and Theory

Chapter 1
Overview of Music Theories

The title of this chapter states *Music Theories* in the plural and not the singular *Music Theory* or *Theory of Music*. Probably no single theory will ever cover the enormous richness of music in the world, although there have been promising endeavors to extract certain principles common to most. This is a boon and a bane at the same time: a bane, because all attempts at reducing music to a single set of rules have failed so far to satisfy adherents of universally valid systems; a boon, since it relieves music lovers from the potential danger of the dreaded reduction of music to a mere system of rules, devoid of all mysticism cherished by many. Even putting aside all aspects of the subjective, this state of affairs promises new vistas for future musical activity, since the reservoir of new and modified theories for generating new music seems inexhaustible.

The role of theory in the analysis as well as the production of music has however not been the same at all times in the history of music. To better understand how theories and what theories fit into contemporary music, it is necessary to shed some light on the last few hundred years during which music theories have been perceived as such by musicians.

It is necessary to restrict the discussion to theories about Western music. Of course theories of music from other cultural environments such as India exist and its literature is very broad indeed, but including it would by far break the tight limits set by this text.

Certainly the most famous theory linking music and natural science known from Antiquity is the description of music intervals using of chord ratios by Pythagoras. However from the post-Antiquity treatise by Boethius through the Middle Ages until 19th century, music theories were rather enshrouded in the philosophical (or theological) frameworks of the day. Those parts relevant to the analysis and composition of music did not much more than describe the state of the art as it was a decade before the publication of the treatises.

Thus the *Gradus ad Parnassum* by Johann Joseph Fux published in 1725 [25] laid out the rules of counterpoint, even if those rules did not accurately describe the practice of most advanced contemporary composers, such as Johann Sebastian Bach. This work, well used even into the present, also illustrates features common to similar theoretical treatises: they delivered a set of recipes justified by empirical or psychological phenomena, thus providing weak theories, even in comparison to the studies pursued by Greek mathematicians.

The schema behind the history of these theories can be summarized in the phrase: *Theory follows Composition*. There is hardly any theoretical proposal that would provide the composer with new ways and guidelines to drive his work. Therefore, it is mostly the case that "avant-garde" composers, such as Beethoven in his late string quartets, produce the material that musicologists try to put into theory *after the event*.

The beginning of the 20th century saw new developments of music theorists trying to give the various aspects of music, such as harmonic and formal structure, a more exact and scientific character. This is certainly connected to the rise of abstract mathematics in the 19th century, which relies on new rigorous types of notation, as well as the use of diagrammatic methods. Such theories include the *Funktionstheorie* by Hugo Riemann and the structural theories by Heinrich Schenker.

During the second half of the 20th century, finally, modern mathematics found its way into theoretical considerations of music [5]. Combinatorics is seen as playing a major role throughout the history of musical composition. Dodecaphonic composition and its generalization to serial techniques provide the telling example of this trend. Driven by the popularity of structuralist views, generative theories as they were first proposed by Noam Chomsky for the formalization of language have been applied to musical form as well [38].

This has been a breakthrough in that, for the first time, theories have been devised not least with a view to provide new means for composition. Composers start complementing their musical works by theoretical studies explicating the methods that led to their construction, for example *Musiques formelles* (1962) by the Greek composer Iannis Xenakis [74]. Several other eminent composers, such as Arnold Schoenberg, Ferruccio Busoni or Paul Hindemith, have published theoretical works that give insight into their works. It has eventually become possible to state that: *Composition follows Theory*.

In the last decades, the role of mathematics in music has been on a steady increase. The invention of digital computers in the late 1940s was soon followed by the creation of music with the help of computers (for example the famous Illiac Suite from 1957 by Hiller and Isaacson [31]) and along with it the development of mathematical theories quite specific to musical applications. It was Milton Babbitt, Hillers teacher of composition at

Princeton University, who promoted the use of mathematics, its terminology and, thus, scientific precision in studying theories of music. A music theory should be objective and be stated as a body of definitions, axioms, and theorems, just as in the case of other mathematical investigations. An analogy can be made to the relation of physics to engineering: mathematical music theories play the role of physics, and musical composition the role of engineering. It is obvious that without the theoretical work provided by physics, engineering would be impossible, or, even worse, engineering with a disregard of physics would result in bad work. Cautiously applying the analogy, we would say that contemporary composition without regard to mathematics results in bad music. . . .

Two types of composition by electronic means have been predominant. One type is exemplified by the computer programs *Project 1* (1964) and *Project 2* (1966) developed by the composer Gottfried Michael Koenig. They implement composition algorithms and operate on the level of musical structure. On the mathematical side, aleatoric and combinatorial methods play an important role [36].

Another line of development involving the use of computers for musical composition concerns the production of *sound* by means of digital processing. In this case music is controlled on a very low level, effectively down to the sound wave. Many parameters, which before were constrained by acoustic instruments, become available. These parameters allow qualities of sound never heard before and thus fruitful for new principles of composition. The pioneering work has flown into the MUSIC-N family of synthesis languages, the prototype program MUSIC having been written by Max Mathews in 1957 at Bell Labs [42]. The most recent descendant in the family, widely used today on many computer platforms, is Csound [10].

The import from mathematics to musical theories comes from all domains of abstract algebra, such as group theory. More recently, category theory has provided comprehensive tools for both analysis and composition. Mathematical studies have been conducted for fields such as counterpoint or the theory of the string quartet. A main event in this direction has been the publication of *The Topos of Music* by Guerino Mazzola [47], which uses extensive knowledge from category theory to provide theoretical treatments of a broad selection of branches in musical theory.

The mathematical approach to music analysis and composition also allows new means for joining the musical past to the future. In accordance with the idea of "analyse créatrice (creative analysis)" put forward by Pierre Boulez [49], Guerino Mazzola undertook the ambitious project of a new composition based on the analysis of the first movement of Beethoven's sonata op. 106 "Hammerklavier" [43]. The process involved the discovery of mathematical principles buried in the piano sonata and the extraction of significant parameters ("analysis"). The result is an analytical model. After changing the analytical coordinates of this model using mathematical

transformations, the model was remapped to produce a new musical work that fundamentally based on, but different from, Beethoven's composition ("creative"). The process itself is not entirely new, as similar principles have been employed in a vague manner throughout the history of music. But the mathematical approach, the exactness of the procedure, and the awareness of its happening are aspects of a modern phenomenon and open new vistas for future experimentation. The result of the project is the piano sonata "L'Essence du Bleu" which has also been recorded [48].

The sonata has been constructed completely by hand, using the traditional tools available to the composer and mathematician, such as pencil and graph paper. To manage this kind of composition, computer software implementations of the methods used would provide an ideal tool. The development of such software is of paramount importance for future experimentation in this direction. The RUBATO COMPOSER presented in this thesis is the result of this insight. It is the attempt to bring the mathematical and computational tools to a level that a composer is comfortable with. The mathematics it is based on is category theory, specifically the category of modules. Module theory is very important in mathematical music theory, since it allows the representation of the common objects in music, such as notes and chords, and the description of the usual transformations in the theory of counterpoint and composition, such as transposition, inversion and retrograde, among others. The theory of modules contains the complete theory of linear algebra, and thus Euclidean spaces, which ensures the availability of general geometric manipulations at the hands of the composer. Categorical considerations led to the development of a very general data model, called denotators, which is used throughout RUBATO COMPOSER for the transport of concepts and information.

Chapter 2
The Representation of Music

Theories, whether in mathematics or in music, presuppose a world of objects that we can be theorize upon. In mathematics the notion of objects is quite clear, from obvious number domains to high dimensional manifolds. Musical objects, if we may call them thus, are much more elusive. Identifying *the* musical objects would be nothing less than determining what music ultimately *is*. So far, no one really knows what happens in the brain when listening to music, what symbolic structures are generated and and how they are processed. The psychology of music does make some inroads, but without effective results useful to the theorizing we have in mind. We are left with the only possibility of grasping musical objects, namely proposing more or less formal schemes for the representation of music, preferably in a mathematical context.

Such a representation should allow the manipulation required by the various methods of musical composition as well as analysis. Certainly, no one format will be able to provide all of this, and, in effect, the formats developed during the centuries of written music targeted overlapping, but different, uses.

2.1 Types of Representation

To bring a little order into the various types of representation, it is helpful to borrow from the ontology and semiotics of music. The cube shown in figure 2.1 is an abstract model of locating music within our knowledge system. It considers multiple aspects, which may or may not enter in a specific type of representation. The levels of *reality* discriminate between the layers where music takes place. The *physical* and *mental* levels will be of primary concern in this context. The *communication* coordinate of the cube refines these layers and relates the musical work to its creator and the recipient.

This coordinate is usually not made explicit in the representations that we discuss below. The third coordinate *semiosis* deals with the semiotics of music. It provides the theory of how expressions (*significants*) are related to the meaning (*significate*) of a musical work. The act of relating significants to significates is called *signification*. The process of signification is the attempt of extracting the *content* from a representation. An in-depth discussion of musical ontology can be found in [47].

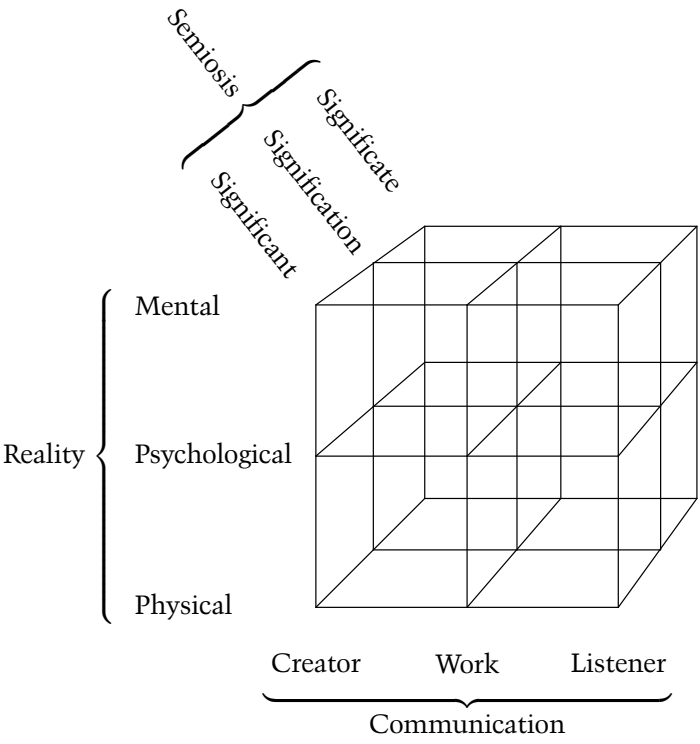

Fig. 2.1: Topographic cube of musical ontology, after [47].

The semiotic status of the various types of music representation is very difficult to assess, since we have to deal with two types of information. On the one hand, there is *explicit information* (or extension). It is the kind of information that can be controlled by rules, many of which have even been cast into formal clothes, for example counterpoint rules according to Fux [25] or harmonic theories such as Schoenberg's [63]. On the other hand, there is *implicit knowledge* (or intension). This knowledge cannot be formalized. It depends on environment, personal preferences, historical background, etc.

[2]. The consequence of this is that, whatever representation we use, the act of signification, and therefore the relation of expressions to meanings, varies from individual to individual, thus making impossible the "ultimate", completely unambiguous representation.

It has already been hinted that the levels of *physical* and *mental* realities play an important role in the representation of music. A physical representation aims to reproduce music in terms of sound, often as low-level as air pressure changes as in the physical theory of sound. The theories of sound production as in the acoustics of instruments also belongs to this class. Common formats of representation include analog tape recordings or digital audio file formats such as MP3. It is the quality of *sound* rather than of *music* that is transported in this kind of representation. It has been the means of old to preserve musical performances and to archive oral, non-written music common in ethnomusicologist studies. More recently, audio has been supplemented by video, thus capturing an important part of performance, namely the gestures involved in making music.

However important the interest in the other types of representation is, we are going to concentrate on symbolic representations, which try to capture the *mental* aspects.

2.2 Symbolic Representation of Music

The printed score is the most famous of all representations. Developed from a gestural language, the neumatic notation used in manuscripts of early church music (see figure 2.2 for an example) [19], it was extended and refined over ten centuries to the point of integrating a wealth of information for the musical performer. In its original form it reached its peak during the first decade of the 20th century, when composers such as Gustav Mahler determined every aspect of the performance of their works using extremely fine-grained instructions affecting even single notes. Some structure is indicated and beginning in the Romantic era, content is hinted at by often poetic directions (for example, *très doux et pur* in figure 2.3 showing the beginning of Scriabin's 10th Sonata).

Other printed representations targeted at specific instruments and types of music have been devised, such as the various tablatures for lute and keyboard music, very common from the 15th to the 18th centuries [23].

Along with common notation, many composers from the second half of 20th century devised notations of their own to communicate the intent of their works or even specific to a single work [17]. In many cases these graphic "scores" have been considered as works of art in themselves.

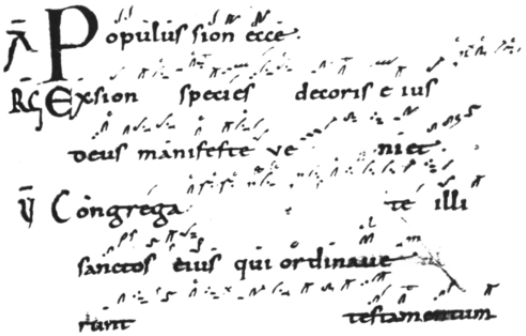

Fig. 2.2: Neumatic signs describing the melodic motion above the lines of text (St. Gall, 10th century).

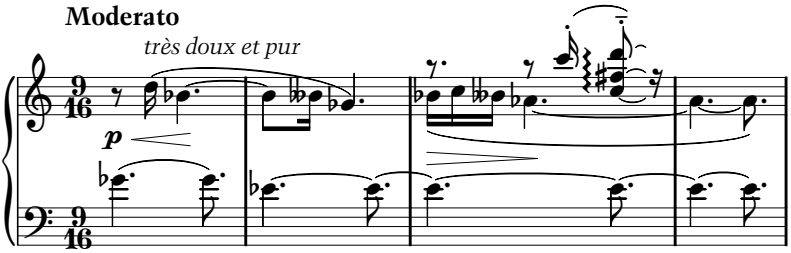

Fig. 2.3: The first four bars of Alexander Scriabin's Sonata No. 10, Op. 70, in traditional notation.

2.2.1 Electronic Scores

Early in the development of computers, handling of music has already been a hot topic. Thus it comes as no surprise that many efforts have been made to bring musical scores into electronic form. One aim has been, of course, to follow the way of digital book production and make the production of musical scores entirely digital. A whole industry is devoted to the development of so-called *music notation software*, and with it, the invention of electronic score representations.

Two examples of quite different approaches shall illustrate this idea. The popularity of XML has had its effect on music representation too, and we now have a large number of XML based markup languages for music, such as SMDL (Standard Music Description Language) [65], NIFF XML (Notation Interchange File Format), and many more.

One particular scheme is MusicXML [59], which has become the lingua franca for the interchange of electronic scores between many notation pro-

grams, e.g., Finale and Sibelius. As is often the case with XML, even a small musical fragment generates a rather large instance of MusicXML. Therefore only part of the music from figure 2.3 as coded in MusicXML is shown in figure 2.4.

```
<?xml version="1.0"?>
<score-partwise>
  <part-list>
    <score-part id="right">
      <part-name>Right</part-name>
    </score-part>
    <score-part id="left">
      <part-name>Left</part-name>
    </score-part>
  </part-list>
  <part id="right">
    <measure number="1">
      <attributes>
        <divisions>960</divisions>
        <time>
          <beats>9</beats>
          <beat-type>16</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <rest/>
        <duration>480</duration>
        <voice>1</voice>
        <type>eighth</type>
      </note>
      <note>
        <pitch>
          <step>D</step>
          <octave>5</octave>
        </pitch>
        <duration>240</duration>
        <voice>1</voice>
        <type>16th</type>
      </note>
      <note>
        <pitch>
          <step>A</step>
          <alter>1</alter>
          <octave>4</octave>
        </pitch>
        <accidental>sharp</accidental>
        <duration>1440</duration>
        <voice>1</voice>
        <type>quarter</type>
        <dot/>
      </note>
    </measure>
  </part>
  <part id="left">
    <measure number="1">
      <attributes>
        <divisions>960</divisions>
        <time>
          <beats>9</beats>
          <beat-type>16</beat-type>
        </time>
        <clef>
          <sign>F</sign>
          <line>4</line>
        </clef>
      </attributes>
      <note>
        <tie type="start"/>
        <pitch>
          <step>F</step>
          <alter>1</alter>
          <octave>4</octave>
        </pitch>
        <accidental>sharp</accidental>
        <notations>
          <tied type="start"/>
        </notations>
        <duration>1440</duration>
        <voice>1</voice>
        <type>quarter</type>
        <dot/>
      </note>
      <note>
        <tie type="stop"/>
        <pitch>
          <step>F</step>
          <alter>1</alter>
          <octave>4</octave>
        </pitch>
        <notations>
          <tied type="stop"/>
        </notations>
        <duration>720</duration>
        <voice>1</voice>
        <type>eighth</type>
        <dot/>
      </note>
    </measure>
  </part>
</score-partwise>
```

Fig. 2.4: MusicXML representation of the fragment figure 2.3.

These XML based formats are mainly intended for interchange and are not meant to allow authoring on the source itself. In contrast, there is a school of typography that favors textual representation of typesetting instructions that are compiled into high quality renderings. The main representative is Donald Knuth's TEX, which allows for score typesetting using the add-on package MusixTEX. Modern descendants are ABC [70] and the powerful score typesetting software Lilypond [28]. The score fragment in figure 2.3 is effectively the result of compiling the code in figure 2.5. Observe how Lilypond tries to find out a good layout without the author providing precise instructions. The implementation of such layout software requires advanced research and techniques in computer science [27] and the construction of a fully functional and complete software application is a major undertaking.

```
\score {
    \new PianoStaff <<
        \time 9/16
        \new Staff {
            \clef treble
            {
                r8^\markup{\bigger\bold{Moderato}}\p\<
                d''16^\markup{\italic{tres doux et pur}}\(
                bes'4.\! ~ bes'8[ beses'16] ges'4.\)
                <<
                    {
                        r8. r8 c'''16-.(
                        <d''' fis'' c''>8\arpeggio)-.--\laissezVibrer
                        r16
                    }
                    \\
                    {
                        bes'16\>\([ c'' beses'] as'4.\! ~ as'4. ~ as'8.\)
                    }
                >>
            }
        }
        \new Staff {
            \clef bass
            {
                ges'4.( ges'8.)
                es'4. ~ es'8. ~
                es'4. ~ es'8. ~
                es'4. ~ es'8.
            }
        }
    >>
}
```

Fig. 2.5: The Lilypond source code used to generate figure 2.3.

All of these types of representation, different as they may be, serve the main purpose of the visual layout of traditional printed scores. Machine-based manipulation and feature extraction for analysis would be hard, if not impossible. This is to be expected, since they adopt all of the idiosyncrasies of human music notation accrued over the centuries.

Some attempt to address the strictly musical aspect and allow the export of a MIDI version of their data (Lilypond for example), but for a more deeply musical understanding, other representations are needed.

2.2.2 MIDI

The Musical Instrument Digital Interface (MIDI) was first proposed in 1981 as a means for driving electronic instruments, such as synthesizers. The standard [55] comprises three parts:

1. a specification for a serial cable and plug that establishes the physical connection of electronic devices;
2. a protocol that defines the possible *events* that are sent over the cable in real-time and are to be interpreted by the receiving instrument (where each instrument is assigned a channel number);
3. a file format that integrates such events, provides them with the times of their occurrence, and is enriched with so-called meta events for changing tempi and selecting programs (a means for choosing among several sound types provided by a device).

The most important type of event is *Note on/off*. The *Note on* event signifies the pressing of a key. This event requires as parameters the channel (which is the address of the receiving device), the pitch (where the number 60 is defined to be c^4), and the velocity, which essentially corresponds to the loudness. The *Note off* event analogously signifies the release of a key.

Figure 2.6 lists the MIDI events generated from a simple rendering of the fragment in figure 2.3.

MIDI does show its age though, in particular in its hardware limitations. Its affinity to keyboard instruments makes it cumbersome to control musical events that are not based on semitone scales, the workaround being to make use of meta events such as pitch bending. Nevertheless, MIDI can be put to good use when it comes to create fragments of musical performance, since the file format is probably the most common of all among music software and therefore is a sure value for interchanging data. We will come back later to this use when discussing the RUBATO COMPOSER software.

Time	Event	Channel	Pitch	Velocity
0	Tempo	Meta Event	Meta Event	MSPQ = 1666666
0	Program	1	program = 0	
0	Note on	1	66	100
240	Note on	1	74	100
360	Note off	1	74	127
360	Note on	1	70	100
1080	Note off	1	66	127
1080	Note on	1	63	100
1320	Note off	1	70	127
1320	Note on	1	69	100
1440	Note off	1	69	127
1440	Note on	1	66	100
2160	Note off	1	66	127
2160	Note on	1	70	100
2280	Note off	1	70	127
2280	Note on	1	72	100
2400	Note off	1	72	127
2400	Note on	1	69	100
2520	Note off	1	69	127
2520	Note on	1	68	100
2760	Note on	1	84	100
2880	Note off	1	84	127
2880	Note on	1	72	100
2880	Note on	1	86	100
2880	Note on	1	78	100
3120	Note off	1	72	127
3120	Note off	1	86	127
3120	Note off	1	78	127
4320	Note off	1	68	127
4320	Note off	1	63	127

Fig. 2.6: A sequence of MIDI-events. In this example the unit of time is defined to be 1/480 of a MIDI quarter note. The *Tempo* event specifies the duration of a MIDI quarter note as MSPQ, which means milliseconds per quarter note.

2.2.3 Musical Representation Languages

Musical notation and instrument control languages are ultimately surface representation and not conducive to analytical research. They require a musical mind to make sense of the analytical signification looming behind the music they describe. On the other end of the spectrum are the languages explicitly designed for the purpose of unfolding analytical structure. These languages are invented in the context of the development of software for musical composition and analysis.

In the world of computer-based manipulation of symbolic data, the programming language LISP assumes a prominent place. Predominantly used