21

Stemma Theory

O matre pulchra filia pulchrior.
Horace (658 B.C.)

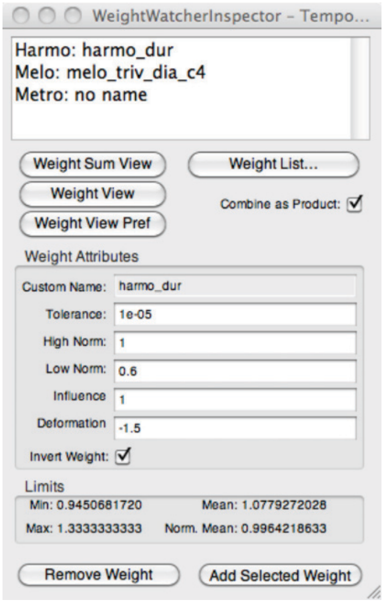

Fig. 21.1. The WeightWatcher window of the PerformanceRubette enables the input of a weighted combination of different weights for the chosen operator.

The Rubato methodology was opened towards a variation of performance rules and their rationales, and although only the analytical rationale has so far been implemented, there is no obstruction against enrichments with emotional or gestural expressivity. This is due to the modularity of the Rubato architecture. Performance rules are by no means encoded in the analytical components. This is a categorical improvement on the KTH approach, which takes, for example, the harmonic charge rule as a compact, undivisible rule that cannot separate the analytical part (in this case, it is the calculation of harmonic charge) from the shaping commands. This is reflected in the structure of shaping operators. They do certain things with whatever weight comes as input. For example, the tempo operator would take any weight and apply it to the given articulation field via the linear operator $Q_w(E, D)$ as described in chapter 17. The weight can come from rhythmical, motivic, or harmonic analysis, or even be a weighted mixture thereof. This is taken care of in the PerformanceRubette's *WeightWatcher unit* (figure 21.1).

G. Mazzola, *Musical Performance*, Computational Music Science,
DOI 10.1007/978-3-642-11838-8_21, © Springer-Verlag Berlin Heidelberg 2011

Even more importantly, the Rubato methodology is also open toward the multistep performance approach. In fact, the KTH approach is a one-step procedure: We apply a bunch of operators (rules) to a given input (the primavista performance) and then obtain the final performance. There is no internal mechanism to concatenate a number of performances. The changes defined by those rules are "stateless," so to speak, they have little if any (some of the rules might act on a previously shaped performance, but most don't) consciousness of the previous shaping results. This is, however, far from the human approach of rehearsing.

But let us first discuss the philosophy of rehearsal! Rehearsal is not only a technical process. It is above all a logical evolution and an unfolding of understanding in time, be it in the individual process of working out a performance or be it in the historical evolution of the understanding of a composition and its performances.

That is to say that rehearsal is not a sign of imperfection, but the silver bullet to the X-ray of the score's innervation as imagined by Adorno.

The logic is that performance is multilayered, since the rationales and operators regard not all parameters at once, and since certain deformations are more basic than others, e.g. the physical operators should come after the field operators. And it is hierarchical in the sense that certain deformations regard larger portions of the score, whereas others regard more local parts, like global tempi and local agogics or the general emotional "mood" of a piece that preceeds the local changes. The entire logic is also a type of syllogism: If this is done to the score, then do that next.

This insight can be formalized by an evolutionary architecture. We model performance as a process of genealogical development, similar to the biological situation. In this sense, earlier stages of evolution are not just less perfect, but also necessary in the logic of steps to more complete organisms (which for humans is sometimes not so clear...). We call the genealogical formalism of performance evolution the stemma. So we have to think about generating stemmata as an evolutionary process in performance.

This being so, the temporal stemmatic consecution is also one of evolutionary logic. The historical dimension of rehearsal is—besides the trivial technical aspect—the trace of an evolutionary process.

Rehearsal is a deployment of logical understanding in time, much as composition is the gestural deployment of the compositional formula.

The big problem here is the communication of such a historical unfolding in a single performance! How can one communicate the stemmatic construction from its resulting output? Should one—in the limit—play the entire stemma? The question is similar to the question of how to perform a Schenkerian analysis in order to make evident the Schenkerian layers in their logical implications. But our situation is more complicated since Schenkerian analysis is a simultaneity of layers, whereas the stemmatic structure is a historical process whose result is not necessarily faithful to its genesis. It is that fundamental problem in jazz: to play the tradition even in the most advanced free context. And to play it not with apologies about the imperfection of an earlier stage, but with the full appreciation of stemmatically prior levels of a logical tree.

So for the time being, we understand the necessity of a stemmatic evolution in the making of a performance, but we do not fully understand how such a fact would be performable.

21.1 Tempo Hierarchies in the *presto*® Software

To illustrate the necessity of stemmatic techniques in performance, we want to discuss the implementation of tempo hierarchies in the composition software *presto*®. Tempo hierarchies are stemmata in the simple domain of tempo. They are now a model for the general stemmatic approach.

Tempo is a classical performance topic, but when modern performance researchers, such as Todd, Friberg, or Peter Desain and Henkjan Honing [26] attempted to control this phenomenon on a precise conceptual a computational level, some strange things happened. First, the very concept was put into question. Desain and Honing in [26]: "There is no abstract tempo curve in the music nor is there a mental tempo curve in the head of a performer or listener." This was perfectly in the line of the early electronic musicicians, who negated tempo altogether. Herbert Eimert and Hans Ulrich Gumpert write [27]: "Electronic music neither knows tempo nor metronome marks, but it documents its connections to the phenomenon of time by the most precise time indications which exist in music." The negation of tempo came from the confusion of symbolic and physical reality in music. But it also came from the fact that traditional performance science in music was all but adequate to performance on an artistic level. We have found in a recent book [21] a reference to the classical work "Pianoforte Schule" by Carl Czerny [17] as exclusive reference to precise tempo shaping. Let us look at that example (figure 21.2).

It shows a short exercise by Czerny, where he compares tempi according to four shapings: 1. in tempo, 2. in tempo/ un poco ritenuto/ smorzando, 3. in tempo/ poco accelerando/ rallentando, 4. in tempo, molto ritardando/ perdendo. My coworker Zahorka has reconstructed these situations with a quantitative tempo curve that can be defined in the module AgoLogic of the composition software *presto*®. The results are all unsatisfactory; hear the first three items of example ♫ 14. The shaped score sounds dull and is not what one would expect from an expressive performance.

This lack of a valid tempo theory does however not mean that there is no tempo, it simply means that one tempo curve is not sufficient. With AgoLogic, one can define very interesting tempo curves, for example *accelerando* curves for drum rolls. It works, but it fails when it comes to artistc tempo shaping. One of the evident situations where a tempo curve is insufficient is Chopin's rubato. This means that one typically plays the left hand with a certain tempo (not necessarily constant), and the right hand plays in a varied tempo, however

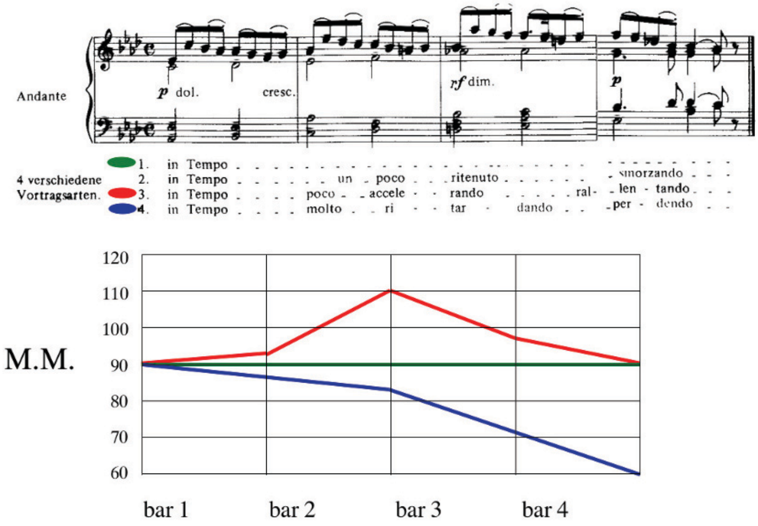

Fig. 21.2. Czerny's experiment with four varied tempi.

synchronous with the left hand tempo in all important onsets, such as bar lines. We have tried this idea in AgoLogic, see figure 21.3.

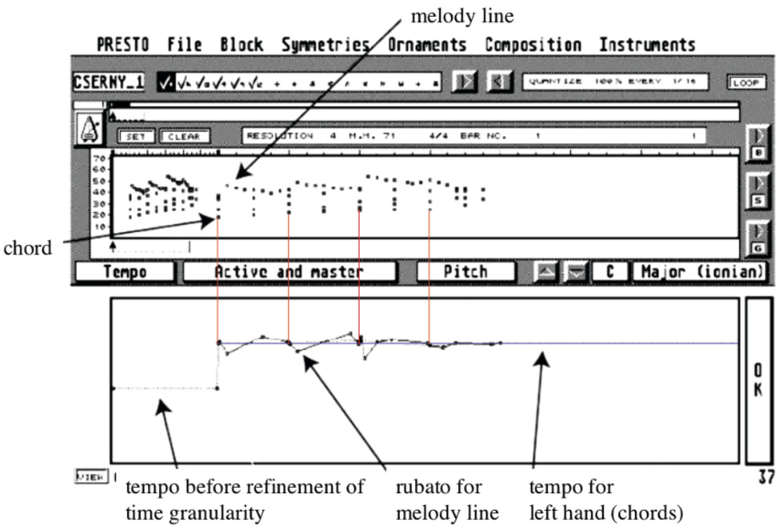

Fig. 21.3. The Chopin rubato, as represented in *presto*®'s AgoLogic module. The constant tempo of the left hand is locally varied for the right hand, synchrony being set for all bar lines.

The left hand is given a fixed constant tempo T_L. AgoLogic enables local variation of this tempo. To this end, we define a splitting of the entire time into four one-measure units. Within each such measure unit, the right hand notes are subjected to locally varied tempi. This is achieved by graphically defining one new curve per measure. The user can draw polygonal tempo curves $T_m(E)$ in each measure m; however, the software will always correct the user's curve to get an overall duration equal to that of the left hand's tempo, i.e., if D is the symbolic duration of a measure, we must have

$$\frac{D}{T_L} = \int_{E_0}^{E_0+D} \frac{dE}{T_L} = \int_{E_0}^{E_0+D} \frac{dE}{T_m}.$$

The software applies an algorithm to approximate the precise value by varying the polygonal shape drawn by the user until a limit of unprecision is reached. The result of this user-defined graphical shaping of local tempi turns out to be quite satisfactory. See ♫ 14 for an audio file (with the three versions 1,3,4 from Czerny and last our Chopin rubato version) recorded from a Yamaha MIDI grand.

The general idea behind this experiment and AgoLogic's method is to introduce hierarchies of tempi (figure 21.4). Which means that we are given "mother tempi," which have "daughter tempi." The daughters are those local variations of the mother tempi, subjected to synchrony on determined boundary points. AgoLogic enables arbitrary deep and arbitrary broad hierarchical ramifications: A mother tempo curve might have as many daughters as required, and the genealogical tree from mother to daughter to granddaughter, etc. might be as long as desired.

Figure 21.4 shows a mamma curve, which has two daughters: curva1 and curva2, and each of them has one daughter: curvetta3 and curvetta4. Each daughter has a curve whose duration is the same as that of the mother within the daughter's time interval. Since every curve in such a hierarchy of tempi is associated with its own set of notes, the entire piece has no overall tempo curve. But the hierarchy expresses a genealogy of tempo logic (this is why we coined this module AgoLogic). The daughter's tempo is a derivative of the mother's tempo, and the daughter's proper shaping is due to the logic applied to this daughter's agogical performance rationales. An impressive application of this module has been realized for a small portion of Frédéric Chopin's *Impromptu* op.29 (figure 21.5).

This short part in Chopin's *Impromptu* is a delicate performance task since it has a complex hierarchy of time layers. The basic mother tempo is the one defined by the three half notes. The mother then has six daughters, each half-note duration having two daughters, one for the left hand, one for the trill in the right hand. Each left hand's daughter has its own daughter defined by the arpeggio. This tempo hierarchy enables very different tempo configurations, from the beginner's rendition to the virtuoso, such as Barenboim or Pollini style—see ♫ 19 for an audio file with these four version: primavista, beginner, Barenboim, and Pollini.

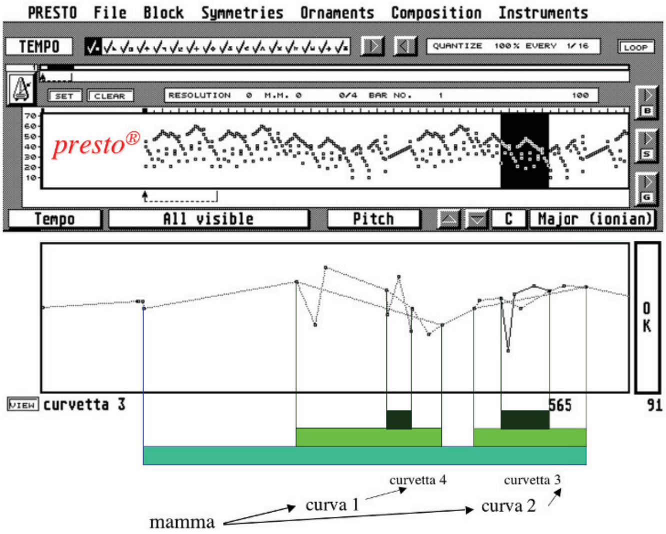

Fig. 21.4. The AgoLogic module in *presto*® models tempo hierarchies by a matrilinear paradigm of mother-daughter inheritance. A mother can have several daughters, and the genealogical tree can be of arbitrary depth.

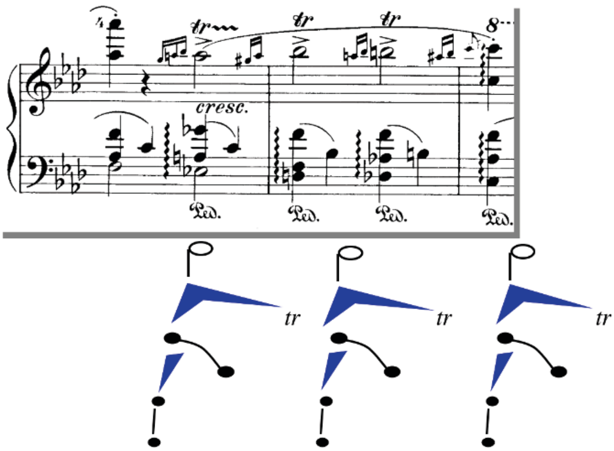

Fig. 21.5. This short part in Chopin's *Impromptu* op.29 is a delicate performance task since it has a complex hierarchy of time layers.

21.2 The General Stemma Concept

With tempo hierarchies in mind, we can now formally describe the stemmatic unfolding of a performance. It is a tree whose nodes are local performance scores (LPS) as defined earlier in chapter 20. Each non-top node is attached to its mother LPS and ramifies to its daughter LPSs. The top LPS, the *primary*

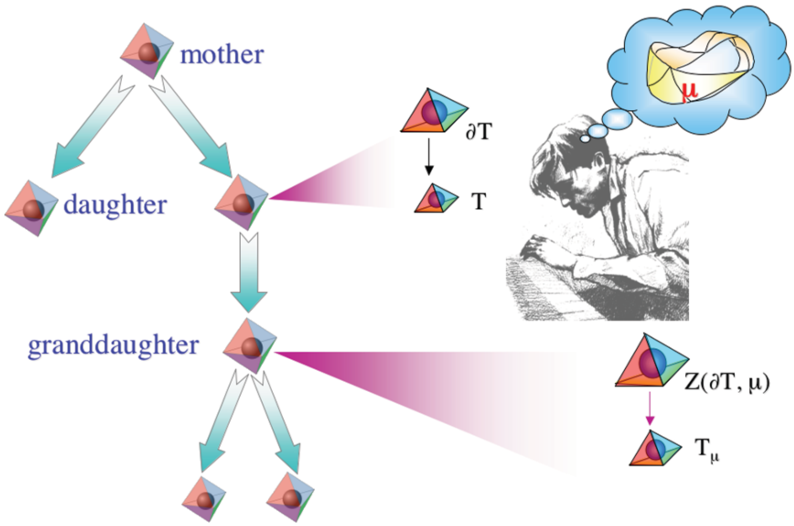

Fig. 21.6. The general stemma concept shows a tree of local performance scores. Their unfolding corresponds to the successive refinement of performance according to selected rationales. In our image, this would be Glenn Gould's harmonic thought that refines an articulation field.

mother, is attached to the score, which is not an LPS. Figure 21.6 illustrates a stemma scheme. It starts with the primary mother and ramifies to her two daughters, one of them having her own daughter (granddaughter), who in turn has two daughters. Therefore, there are three leaves of this tree, and it is those that will be played as a performance output. The others are only part of the internal processing. The passage from a mother to her daughter, as shown in this figure for the right daughter and the granddaughter, is caused by quasi-sexual propagation in that the granddaughter is generated from her mother using the granddaughter's operator, which plays the role of a father.

In our example is shown an articulation hierarchy with a parallel hierarchy $\partial T \to T$, which is reshaped by the analytical weight stemming from a harmonic analysis symbolized by μ, the harmonic Möbius strip of the triadic covering of a scale (see [84, section 13.4.2] for this analytical structure) and yields the granddaughter's articulation hierarchy $Z(\partial, \mu) \to T_\mu$. Since the stemma is a tree, there are no closed paths and no ambiguous states for multiple mothers, for example. This seems reasonable, but it prevents us from formally thinking of feedback cycles. This is an important requirement for realistic modeling of performance genealogy. The only thing we can do at present is to take the output, analyze it 'offline' and redefine parts of the existing stemma to get a better final performance. This is similar to the analysis-by-synthesis approach of the KTH school. But it differs insofar as it is not new rules that must be reprogrammed, because the very flexibility of the system enables us to choose

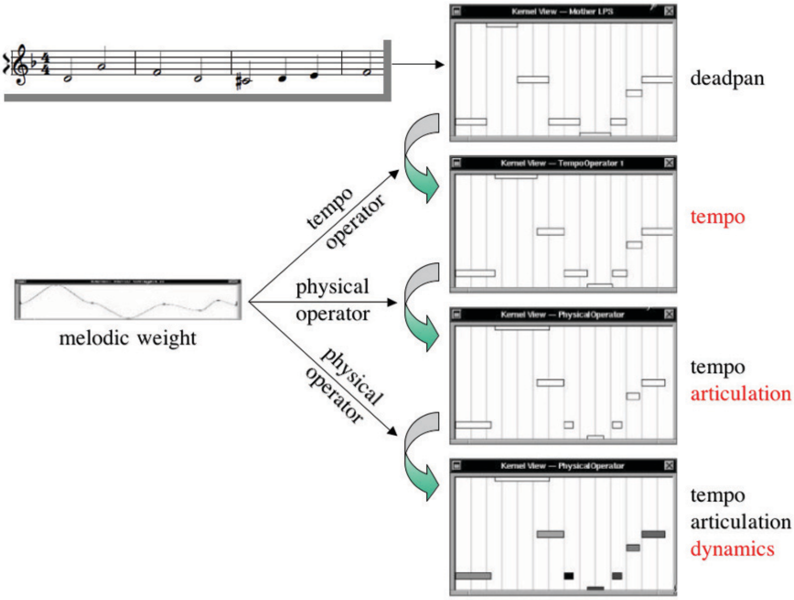

Fig. 21.7. The main melody of Bach's *Kunst der Fuge* is performed in a stemma starting with the deadpan primary mother, then generating its daughter by a tempo operator using the inversed melodic weight, then by a physical operator using the non-inversed melodic weight and acting on duration (for articulation), then by a physical operator using the inversed melodic weight and acting on loudness for dynamical shaping. The graphical representation of the score's notes in Rubato's windows here is in rectangles with horizontal position equal to onset, vertical position is pitch, width is duration, and gray level is loudess.

different parameters or weights or operators when stepping to an improved performance.

The important point in this reshaping process is that we can redefine any of the nodes' parameters at any time. So the performance history can be rewritten in any of its nodes. And, if necessary, one can also destroy a sub-tree starting from any node and reconstruct the tree below that node.

Figure 21.7 is a very simple example of a stemma, which we did on the first NEXTSTEP implementation, showing the NEXTSTEP setup for historical reasons. We take the input as being the main melody of Bach's *Kunst der Fuge*. It is performed in a stemma starting with the deadpan (no shaping, just the mechanical rendering) primary mother, then generating its daughter by a tempo operator using the inversed melodic weight, then by a physical operator using the non-inversed melodic weight and acting on duration (for articulation), then by a physical operator using the inversed melodic weight and acting on loudness for dynamical shaping. The corresponding sound example is ♫ 20.

21.3 The New Performance Rubette: Realtime, Openness, and Gesturality

The increasing potency of current computer systems facilitates a new way of representing and manipulating structural as well as analytical data. This has been exploited for music composition in recent developments for the *Rubato Composer* software. The module *BigBang Rubette*, for instance, provides a gestural interface with which results of affine and other transformations of music are visualized and sonified during the process of manipulation. This gives composers the impression of transformations being a gradual and gestural process, and it makes them accessible even for non-mathematicians. Furthermore, it reduces the ubiquitous distance between production and product.

These paradigms bear a great potential for the further development of tools for performance analysis and modeling, as well. The *MetroRubette* (chapters 16.1, 22.1), currently reimplemented by the author, uses gestural concepts for the input of its parameters, such as *local meter length* and *profile*. The resulting weights are immediately calculated whenever the parameter values are changed. These weights, output in denotator format, can then be used as an input for other rubettes for example, for those who use them to shape a performance of the piece the metrical analysis was made for.

As described in chapter 20, the architecture of Rubato Composer enables the design of smaller computational units than previous software products used. One traditional rubette is now split up into several, all dealing with a well-chosen, specific, and atomic task. This is mostly facilitated by the universal and flexible data format of denotators. Because of their generality, rubettes can be designed such that they can process inputs of any imaginable form. In this way, a rubette created for processing musical denotators in a general manner might as well be used for image processing, as soon as appropriate forms are defined. As described in [135], this type of open design, paired with the generality of the denotator format, ensures maximal flexibility and modularity. The more universal their elements are defined, the more modular the potential networks will be. Such rubettes can be seen as analogous to the musical agents described in section 4.11. They are "performers" themselves that receive a score (neutral niveau), process it in their own way, and pass on a new score that reflects their operations. To reach the highest possible expression, it is important to keep the ears as widely open as possible. Only true musical hearing will define a collaborative space of maximal extent [91].

The next step in the development of Rubato Composer will be to enhance communication between rubettes using concepts from gesture theory. For this, we will need to redesign the Rubato Composer communication system, so that it not only handles offline messages triggered by the *run network* button but also streams of messages, the rate of which would be defined by the outputting rubettes themselves. This way, if a parent rubette changes its input values or its properties, all dependent rubettes will receive the newly calculated output immediately and in turn recalculate their results. This improvement will lead to entirely gestural networks of rubettes, which all react to one another in an immediate way. For instance, a rubette that calculates a performance from specific weights will recalculate the performance and ideally play it back immediately, in a similar way as in the BigBang rubette.

The two requirements formulated in this section—the openness and generality, as well as the gesturality of the communicated data in Rubato Composer—will lead to a comprehensive and flexible system for music composition, analysis, and performance. Just as openness and gesturality form the foundation of human conceptualization and experience.

An example for the application of these paradigms would be a new implementation for the calculation and manipulation of performance fields, which were introduced in chapters 8-10. A future software project aims at a tool for visualization of such performance fields. It is inspired by the *EspressoRubette* (chapter 10) and works in a similar way but is enhanced by several realtime and gestural characteristics. It will visualize the quality of a musical performance during the performance itself. If the input reference composition is an unshaped original score, the performer will see how her interpretation deviates from the original in tempo, pitch, dynamics, and articulation, each of which will be represented by an individual color field, which could be projected on a big screen. This setup can be used as a practical reference for music education, for instance, where students and teachers will be able to see immediately and precisely how their performance is shaped. The visualized performance field could then be modified with gestural interaction and played back for investigative purposes. This way, high-quality performances could be produced in a combined procedure of playing and gestural shaping and then be either saved or played back. Again, this module could then communicate in realtime with other rubettes, which could produce detailed analyses of the resulting performance or use it for music composition.