Modeling Music using XML — Some Basic Considerations —

Markus Lepper
Technische Universität Berlin, Fakultät IV,
Institut für Softwaretechnik und Theoretische Informatik, ÜBB,
Sekr. FR 5–13, Franklinstr. 28/29, D–10587 Berlin,
E-mail: lepper@cs.tu-berlin.de

***Abstract*—**
The paper tries to give basic outline of a holistic and analytic approach to capture all „thinkable“ kinds of music in a generic and aspect-oriented XML-based architecture.

I. Idealized Aims for a (Not-So-)Far Future

It may hardly be called an exaggeration to say that the ongoing development of international standards of data formats and encodings based on XML realizes a **change of paradigms**. This is not because of the quality of the XML-level-one definition[1], — indeed there are certain ugly design flaws due to its historic genesis — but due to the potential of *genericity* and *compositionality*, which has just started to display its power.

The reason for the wide-spread applicability of XML is the very trivial fact of rapidly increasing transmission bandwidth and processing power. This allows a very old paradigm from the first days of ancient UNIX design to become accepted nowadays: That *text* is the adequate basis to represent complex matters. But in contrast to ancient UNIX, these texts are *structured* in a specified way, and the information content is coded implying a certain *abstraction* from the physical appearance of the text object.

While it was practice in the field of information representation during the last decades to talk about bits and bytes on the encoding level, now we can (and have to) speak on a *semantic* level. This implies the opportunity to realize some requirements which are especially important in (or even characteristic for) the field of *æsthetic* production:

Genericity of data formats is a central issue, since the expressiveness, syntax and vocabulary of project-specific data representation should in no ways be limited *a priori*.

[1] With „level-one“ we mean the core definition of the XML-document formats, not regarding schema definition languages, as given in [1].

Furthermore *compositionality* allows the combination and mutual reference of widely diversive data, — an important contribution to creative liberty when using digital machines for æsthetic production.

So (in the long run) we want e.g. to ...

- Feed the output of some off-the-shelf composition algorithm into a score rendering program of another vendor and into a sequencer or performer of a third vendor,
- attach analytic results or didactical information to a musical score or to audio data,
- generate and store the relations between audio data and notation data in different formats,
- synchronize audible sound data, visual information and control parameters for some on-the-stage installation to describe *synæsthetic* pieces of art,
- feed the output of a notation program into some self-defined *ad-hoc* transformation or analysis software, and *vice versa*.
- generate electronic interpretations from notated music by algorithmic transformation, while keeping the different stages of data consistent in a mostly automated way.
- fill eg. a CALS-based table model or an SVG-based diagram with fragments of music notation when authoring a text in the field of musicology.

All these tasks can of course already be done nowadays, but we have to pay the price of tedious programming of data converters, which are always specialized and hardly reusable.

What definitely can *not* be done today is an *automated* generation of those trans-coders and analyzers, which requires a standardized definition of *semantics* (together with some encoding definition).

The conflict between the (necessarily) fixed semantic of a desirable standardization on one hand, and the need of combinability, extensibility and integrability of *ad hoc* inventions of own structure definitions, is a „creative conflict“, requiring solutions which push these „idealistic“ postulations down to the level of mathematically concrete architecture definitions.

The following text wants to make an according proposal.

II. Analysis

A. Coding and Semantics

W.r.t. data format definitions for supporting inter-operability, the central problem is *not* a matter of encoding. Let us call

„encoding“ each transformation from a semantic model S into some grammar of „physical“ data items P. It is easy to see that in fact *all* encodings, which are complete and *injective* functions $c_n : S \rightarrowtail P_n$ are totally equivalent w.r.t. information contents: Since we always can construct $c_n^{-1} : P_n \nrightarrow S$, we can for each two encodings c_m, c_n give a trans-coding function $c_m \circ c_n^{-1} : P_n \nrightarrow P_m$.

So coding *per se* is **not an issue**!

Of course two different encodings will probably differ w.r.t. the performance of certain applications which operate on the data. This is an important aspect which must be respected, but should stay in the second line, as long as the basics are not yet defined[2].

So the central problem is not encoding, but *modeling*. What is it, what we want to encode, — what is this semantic model S which has to be mapped to P_n ?

In case of technical applications this question is answered by the operational semantics of e.g. some given protocol: We easily can encode SQL terms, SIP interactions, TCP requests etc. in XML by choosing any arbitrary *ad-hoc* transformation, because their semantics are specified by construction, e.g. using mathematical models of state machines, grammars etc.

But in case of *cultural phenomena* the situations is totally different: The first step for constructing any semantic model must be an analysis of the existing practice and of the naïve, informal pre-knowledge. The results of this analysis maybe can form a basis for constructing the necessary semantics.

In other words: We have to answer precisely the question „What *is* music“ before we can start to define any encoding[3].

For this purpose we will now present a minimized meta-meta-model[4] of cultural interaction, apply it to musical practice and try to find out the unavoidable consequences for adequate models of music.

B. INTERLUDIUM: Basic Meta-Meta-Model

Our analysis is based on a very (very, very) simplified KANTian theory of cognition.

Roughly spoken, it says …

- that there *does exists* one single „objective reality“ (the Ding-an-sich), totally independent from all human perception,
- but that this „objective world“ is not perceivable by humans at all. All we can perceive is completely bound to our Erfahrung (experience), which is always a common product of that Ding-an-sich and our Wahrnehmungsvermögen (Mode of Perception).

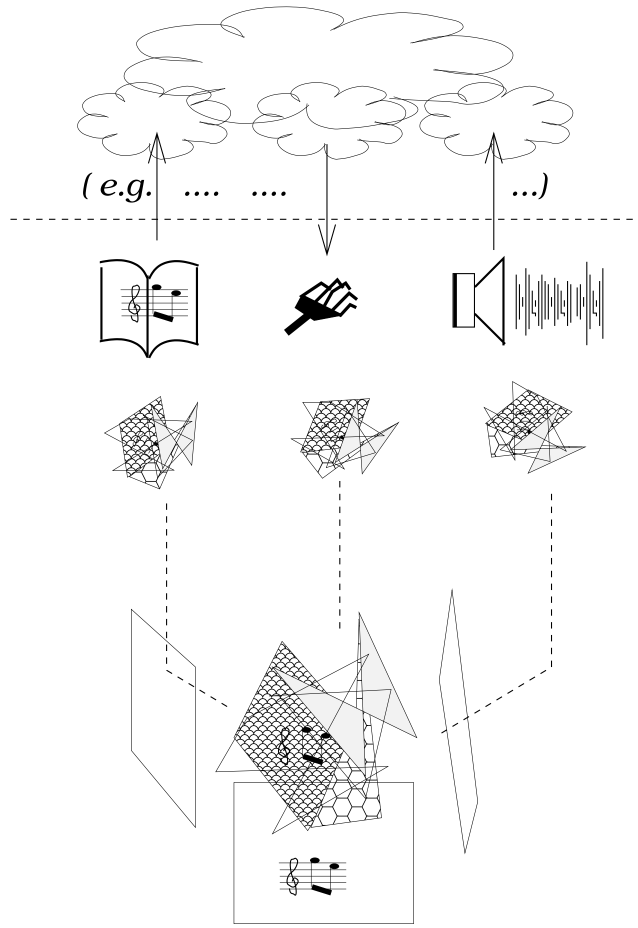


Fig. 1. Interchanging Different Aspect Models Related to „the same“ Ding

This common product is always and necessarily „undecryptable“. It can be compared to some encrypted text, where the private key has infinitely many digits: Mode of Perception and Ding together produce the experience, and all three are totally separated and incomparable aspects of the world.

So the only world which ever can appear to a human mind is a Mental Model. This world is called Wirk-lichkeit, because only internal to this mental model the notions of „cause and consequence“ (= Wirk—ung) are valid, and only internal to this world any notion of time does make sense. This world is specific for (and internal to) any single human mind, and it is impossible to interchange any information between two of these universes: No one can ever transfer to another mind directly the concrete feelings connected with the personal reception of „salty“, „red“ or „BEETHOVEN’s Fifth Symphony“[5 6].

This KANTian meta-model solves the old problem of the „praestabilierte Harmonie“ by LEIBNIZ *et.al*, i.e. it explains, why mathematics can describe the Wirklichkeit[7], — it unifies the

[2]Esp. because the decisions of to-day should take into account not only existing technology, but even more the future development of the next two decades, which seems to be the life-cycle of standards.

[3]The naïve objection, that there do exist many different encodings (with their underlying, but mostly not explicitly defined, semantics) which are well-proven in practical application, is disproved by the every-day experience of composers (as well in traditional style using notation programs, as in avant-garde-style using sound processing languages) that very often the means offered by some standard or product have to be *abused* to achieve the desired effects, e.g. squeezing some MIDI pitch encoding to micro-intervals etc.

[4]Which perhaps better should be called „infra-model“.

[5]We call this tranßendentale Geschiedenheit.

[6]Even if neuro-psychology would be able to record the electrical patterns connected to such receptions, any transfer to another material brain will certainly cause something rather similar, but (thanks God) substantially total different, — just because the brains, to which this patterns are applied, *are* different.

[7]The English word „reality“ and the corresponding German *Realität* refer

correct arguments of the idealism hypothesis with those of the realism hypothesis, — it states that all further attempts in theory of science and of cognition do operate on a „model-theoretic“ layer of reality, not on something „physical“, — and it helps a lot to clarify the use of language when talking about pieces of art.

C. Application to Music

Applying this meta-meta-model to music we therefore can suppose that there is one single, self-identic, independently of all human reception autarkicly existing Ding called „BEETHOVEN’s Fifth Symphony“.

But this Ding is never perceivable, — instead there exist *models*, and every human mind carries one personal model of „BEETHOVEN’sFifth Symphony“, constructed by the accumulation of his/her life-long experiences with all those situations, which by convention are classified as being related to this Ding.

As told above, no direct exchange of these internal models is possible between humans. But of course we are able to *act*: From the viewpoint of the acting mind, actions are operations on the mental model, which are physically (and by means which are themselves unperceivable) related to the Ding. So the modifications we impress into the Ding can of course be perceived by other humans, influencing their mental models.

In case of music these actions can be: — playing a melody on the piano, — writing notes onto a board, — talking about structures and harmonics, — replaying a tape recording, — conducting an orchestra.

All these operations do modify a tiny-tiny part of the large Ding-an-sich, which is the universe. But since we do not operate on the Ding directly, indeed these actions imply the pre-going construction of specialized sub-models of the mental model „BEETHOVEN’s Fifth Symphony“, which model certain *aspects* of the totality, which then can be imprinted onto the universe to allow inter-human communication.

If we call these sub-models *physical aspect-models*, we can say that ...

Theorem 1 *our mental model of any piece of art is exclusively* constituted *by our (social) experiences when interchanging* physical aspect-models*.*

Figure 1 illustrates this psycho-social interactions and the different projections centered around the transcendental Ding.

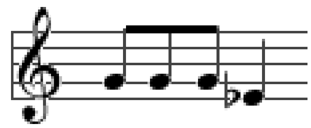
D. Establishing the Relation from Aspect Models to the Ding

One important consequence of this fact is ...

Theorem 2 *the relation from physical actions and the corresponding aspect-model to the Ding must always* explicitely *stated in an* axiomatic *way.*

to the Latin *res*, which means German *Ding*, so that *Realität* can be considered synonymous to Ding-an-Sich, the autarkicly existing, but un-perceivable world. The terminus *Wirklichkeit* but refers to „Ursache und Wirkung“ (cause and consequence) and means those mental models. The English language does not support this distinction.

Imagine an orchestra during warming-up before a concert or rehearsal, when you hear some violinist playing something which sounds like ...



When asking the player, what it *is*, what he or she has played, you can get very different answers:

1) „I played the first notes of my part, the part of the second violin — I did the same thing I will have to do in a few minutes, right at the begin of the performance of the piece.“
2) „I played the well-known beginning of the piece as you could read it in any piano-score, and I could have played it on any instrument which produces well-tempered pitches, but accidentally I have a violin in my hands.“
3) „I played the well-known *motif* from the piece, which happens to appear everywhere in the orchestral score.“
4) „I played the beginning of the recapitulation section, because there the conductor wants a different *timbre*, which I just practiced.“

In cases 2 and 3 the player *refers explicitly* to an abstract model of analysis, which contains only a certain combination of rhythmic values and pitch classes, — in the other cases he or she constructs an index into the larger context of a certain audible interpretation. The same physical action can be *meant* as a model for totally different things of totally diversive contexts.

Another example:

When looking at some written or printed musical text, in most cases parameters like the kind of paper and the chemical composition of the used ink is *not* part of this model: If I want to play a piano piece I only look at the position of the small black circles w.r.t. the horizontal lines.

But some methods in historical musicology deal explicitly with ink, paper, chemistry when looking at a piece of written music. Furthermore it does make a difference w.r.t. law and prosecution, if the copy I am playing from has been legally purchased or illegally photo-copied, — leaving the compositorial structure of the sonata totally untouched.

E. A Possible Classification of Aspect-Models for Music

One first attempt to classify these models of music may identify three areas of formats, cf. figure 2, which can be called...

- Action
- Notation
- Concept

One example for a pure model of **action** flavor is the MIDI encoding, which originally only intends to reflect the activities of a keyboard player, described by the physical coordinates of key presses and releases.

Some older examples are the TABULATURA-style notations, specifying directly the mechanical activities which shall produce the meant musical Ding of music.

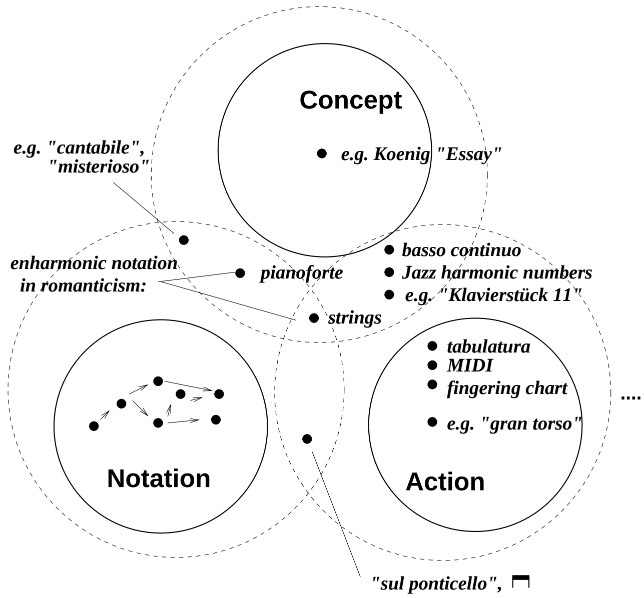


Fig. 2. A possible classification for Aspect-Models of Music

An impressive recent example is HELMUT LACHENMANN's „*Gran Torso*" for string quartet, where the activities the player physically perform make up the material which is organized by compositoric decisions.

The **notation** area is populated with the different, most diversive procedures for fixing music as scripture. Please note that already only the western European tradition during four centuries did produce a whole *family* of notation systems. Since these include even *contradicting* definitions of „encoding" (e.g. w.r.t. the scope of validity of additional accidentials), any attempt to construct a unifying and comprehensive modeling including all these notations must necessarily fail.

The notation of **concepts** became relevant to practice primarily in the last half century with the evolution of the *Darmstadt*-style avant-garde („Neue Musik"): E.g. in G.M.KOENIGS electronical piece „*Essay*" the formulae and data which *generate* the music are notated as lists of integers, and all compositoric/constructive decisions are presented openly to the reader or interpreting musician. Similar with STOCKHAUSEN's „*Studie Eins*", where an important part of the score is made up by the detailed description of the production process.

Any encoding of music cannot be called exhaustive, unless it is capable of representing even these pieces and notations.

Of course these areas indeed overlap, — something like CANTABILE is firstly a semantical directive for interpretation („like singing"), therefore something conceptual, but is contained in the notation, and requesting some physical way of producing notes (legato or portamento, medium range of volume, etc.), thus indicating actions.

Some other termini like A TEMPO ALLEGRO historically began as indication of semantics („happy and gay"), and ended as mere technical advice how fast to tune your metronome.

The BASSO CONTINUO notation and the jazz-style harmonic number system is part of the concept world. But having practiced enough to play a Generalbaß *prima vista*, it becomes a kind of action indication, — the hand nearly automatically is formed to the grip corresponding to the numbers, and you only have to hit it down.

„*Klavierstück Elf*" by STOCKHAUSEN is just a large poster on which some fragments of notated music are dispersed, and has to be played following the accidental jumps of the eyes of the pianist. This also can be considered as a combination of action notation and conceptual contents.

Beginning with C.PH.E. BACH, the *enharmonic* notation becomes more and more relevant as an expression of **concept**: especially when a pianoforte is the instrument, where a naïve and verbatim realization by pitch modification is impossible, enharmonic notation is *purely* conceptual (cf. BEETHOVEN, op. 110, the a" repetition preceding the ARIOSO, SCHUBERT, Sonata in a-minor, etc.)

When instruments with flexible pitch are involved (cf. MOZART, KV 550/I, return section to the recapitulation section, WAGNER, „Schicksalfrage-Motiv", featuring {c, d-sharp, g} as a c-minor pseudo-function), enharmonics *may* additionally carry action semantics, because intonation may be influenced by the notation[8].

III. CONSTRUCTION

A. The Hypothesis of Events

While these „philosophical" considerations may seem to be somehow far-fetched, indeed the opposite is true: Ninety-nine percent of music representation and interpretation will work fine without them, but as soon as advanced compositoric or analytic techniques are applied, there will be a point where all *closed* notation systems will necessarily not suffice.

Indeed does the transcendental nature of music have a sever impact on every-days work with musical objects, even in the most trivial contexts, — but mostly in a non-obvious way.

Let us now continue our analysis and concretize it towards the construction of a meta-model:

Let S be a transcendental Ding, a piece of music, and $M_1 \dots M_m$ be some models related to S by axiomatic proposition.

A reasonable hypothesis, which indeed is the basis for nearly all existing model languages and their encodings, is …

Theorem 3 *The transcendental Ding S and each model M_n can be thought of being composed as a collection of* Events, $E(S)$ *and* $E(M_n)$.

The axiomatically proposed reference between S and M_n includes the axiomatic proposition of one certain mapping relation $E(S) \leftrightarrow E(M_n)$.

But since the transcendental Ding itself is not accessible to our experience, this does not help at all. Luckily we can conclude that …

[8] But this is a derived, second-order phenomenon, since even in absence of enharmonics the intonation of a string quartet is *always* an act of interpretation: In the most simple C-Major context you always have e.g. at least *three* physical pitch classes, all notated with „e" (the third of the tonic and the fifth of the dominant to the parallel of the subdominant, and the fifth of the triple-dominant), between which each interpretation has to decide.

Theorem 4 *The axiomatic proposal of two distinct models* M_1 *and* M_2 *being related to the same Ding, must imply the definition of a mapping relation between the corresponding sets of events* $E(M_1) \leftrightarrow E(M_2)$*, consistent with both* $E(M_1) \leftrightarrow E(S)$ *and* $E(M_2) \leftrightarrow E(S)$*,*

Please note that we have in neither case any *a priori* information about the structure of these mappings, if they are functions or even injective functions, if they are total or surjective etc. This indeed may cause severe practical problems, as will be shown in some examples later (cf. section III-D).

The next theorem is also an abstraction of every-day's practice and underlies most existing model languages:

Theorem 5 *Every event* $e \in E(M_n)$ *of some model* M_n *can be* totally *characterized by a family of (maybe partial) functions* $P_p : E(M_n) \nrightarrow V_p$*, where the ranges* V_p *of the functions* P_p *may be different for different p.*

The values $P_p(e)$ *are usually called the „parameters of the event e".*

Theorem 6 *Usually there are some ranges of parameters* V_p*, on which a* total order *is defined.*

And that's all. Sorry.

Indeed further analysis shows, that it is impossible to sensefully extract or define any further property of events and parameter functions, without massively narrowing the domain of representable musical concepts and data.

Practical Application: Consider e.g. a musical piece given (a) as full orchestral score, (b) as a set of orchestral parts, (c) as a set of different MIDI-interpretations, (d) as a set of different CSound-interpretations, and (d) a wave file containing the recording of a production. Each of these formats has totally different ways of indicating starting point, duration and pitch of the single events, and functions for converting from one format to the other may be rather complicated (e.g. requiring further parameters for an interpretative act like in the Rubato project [5], or are simply context dependent, since global tempo indications have to be considered!) or even not existing at all (e.g. between different interpretations).

In our approach it does not make any sense to declare one of those formats to be the „real" music, and the others as „only derived". Instead all formats are first-order residents with equal rights, each format reflecting just different aspects of the abstract events an-sich .

B. Foundation

Indeed we can use the results of the transcendental analysis almost immediately to give a sound foundation to the mathematical formulation of all possible models of music.

This is done by a kind of co-algebraic approach, which states, that the „objects" are not accessible *per se*, but that different *observations* can be made on a certain object, by applying to it one out of a family of „enquiring functions".

So we *can* model transcendental objects as such, as long as we do make observations only on their counterparts on model-level. We define ...

Definition 1 *Whenever enquiring for some information from a model of a piece of music S, there exists a fixed, known and finite set of „Abstract Events" A. This set is exclusively characterized by the fact that it is the one and only common domain for all parameter functions* P_p*.*

It is always true that $E(S) \subset A$*.*

This seems to be not much, but indeed it is a very practical and powerful approach: It implies that e.g. when modeling a violin glissando or a crescendo or a parameter interpolation in an electronic context, it must always be sufficient to model only a *finite* set of data points. For an uncomplicated mathematical foundation of transformation and language semantics this is a necessary requirement.

In practice this means that all modeling of a given piece of music is done by incrementally adding parameter functions, the domain of which is always the same, namely a finite set (e.g. of natural numbers) representing *abstract events*. The *only* purpose of this set of abstract events is to establish a relation between the *domains* of two given parameter functions. Abstract events are never accessible by our perception, and we cannot state anything about them, beyond their usage as elements of the overall common domain[9]!

Practical Application: Please note that also all „auxiliary" score items like

- tempo indications and interpretation directives,
- key signatures and measure indications,
- titles, texts and rehearsal marks,
- etc.

have to be modeled as „Abstract Events" to fit into our framework.

C. Parameters and Scales

Since no further statements can be made on these morphisms, all semantic discussion must take place w.r.t. the parameter functions.

For sake of inter-operability, compositionality and precise semantics we suggest to refine the *range* side of the parameter functions by splitting it into to parts, which may be specified *separately*: The first of which indicates an appropriate *scale* which governs the parameter, the second contains a vector of some scale-specific numeric values:

$$P_p : E(M_n) \nrightarrow K_p \times \overline{V_p}$$

$$\overline{V_p} == V_{p,1}, \ldots, V_{p,k}$$

$$\begin{aligned} \forall\, p, n \quad \bullet \quad & V_{p,n} \equiv \mathbb{Z} \ \vee\ V_{p,n} \equiv \mathbb{Q} \\ \vee \quad & V_{p,n} \equiv \mathbb{R} \ \vee\ V_{p,n} \equiv \mathit{Text} \\ \vee \quad & V_{p,n} \equiv \mathbb{P}\mathbb{Z} \ \vee\ V_{p,n} \equiv \mathbb{P}\mathbb{Q} \ \vee\ V_{p,n} \equiv \mathbb{P}\mathbb{R} \end{aligned}$$

...where K_p is the set of appropriate *scale objects* for the parameter described by P_p, and $\overline{V_p}$ is a tuple of corresponding arguments into this scaling definition. Please note that these arguments can be restricted to be *simple*, i.e. no free types or structures are required on this level of construction.

It is a central requirement that the scale objects are themselves *definable within* any musical score! Consider e.g. figure 3, which shows schematically an example very common in avant-garde style notation. It is easily seen that there are three

[9] Using XML-speak as a metaphor, this is like using `IDREF` attributes with corresponding values, without ever using a corresponding `ID` attribute.

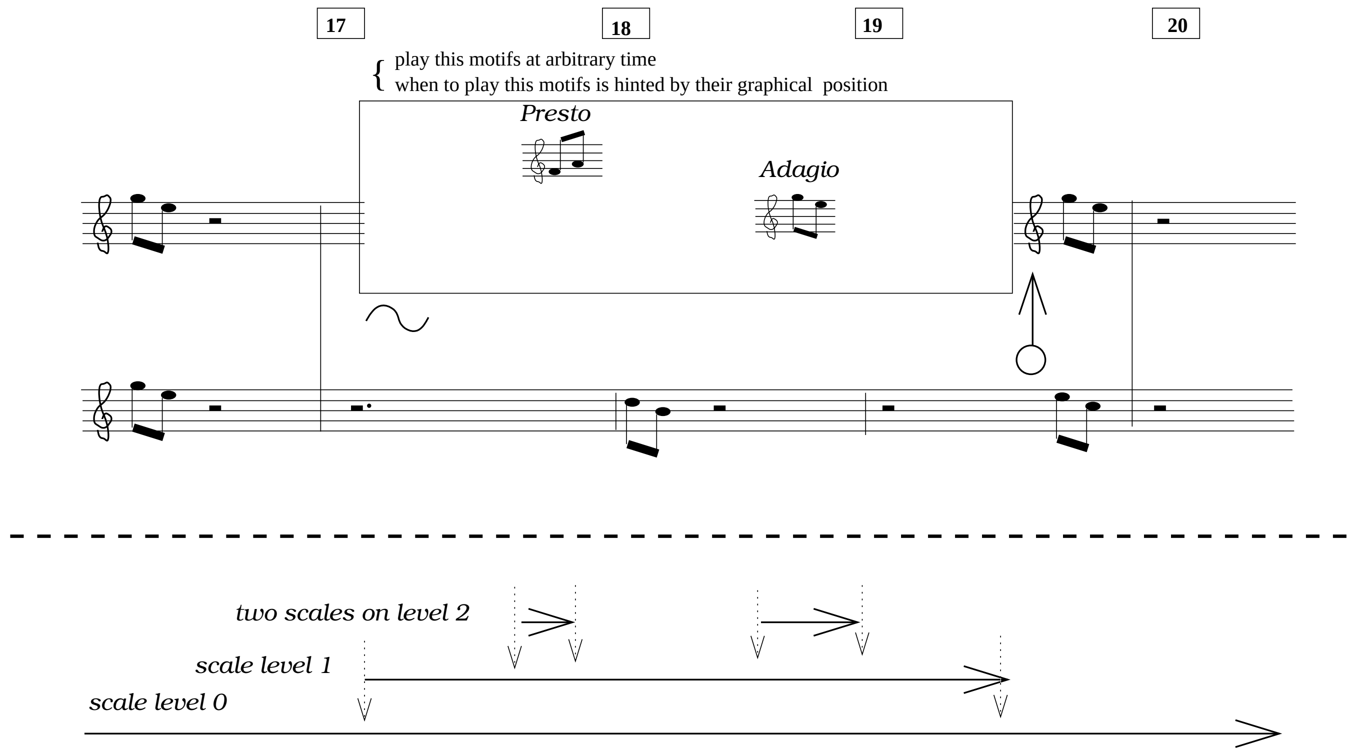


Fig. 3. A simple example of nested time scales

levels of scales, to which events are related, and which are defined in a nested manner[10 11].

So we have to provide means for defining scale objects *within* a musical document, which can be accomplished by treating them similar to abstract events, yielding ...

Definition 2 *It is always true that* $K_p \in A$.

Now we can define scale objects by parameter functions P_k, similar to „normal" events, by ...

$$P_k : K \nrightarrow K \times \overline{V_p}$$

... where K is the union of all K_p, so that we need to additionally impose the requirement of non-circularity[12].

Practical Application: Initially there must be of course some predefined scales, which serve as the semantic foundation for the scale definition mechanism. These *a priori* scales will in most cases be oriented on some predefined „physical" models, cited from existing non-musical standards.

Additionally we define a meta-parameter-function called `constr`, pointing for each scale into a set of predefined *scale constructors*:

$$\texttt{constr} : K \rightarrow \mathcal{K}$$

In the example of nested scales, the values of the parameter functions `start` and `end` where given by recurring to abstract events. To express those kinds of equalities, we allow the „scale and arguments" pairs to be deputised by „parameter function and abstract event" pairs.

Thus the overall information structure which has to be modeled[13] turns out to be

$$P : A \nrightarrow \Big((K \times \overline{V}) \;\; \cup \;\; (A \times P) \Big)$$

D. A Possible Syntax for the Generic Semantics

Some possible realization which can be mapped to our semantic model is given in figure 4. Please note, that this grammar constitutes a generic superset, since it does not reflect the

[10] The two variants of the interpretation directive („when to play this motifs is hinted by their graphical position" vs. „play this motifs at arbitrary time") gives different semantics to the position parameter function w.r.t. audio interpretation, but does not affect the semantics w.r.t. score rendering.

[11] The author had to face in his work as composer an even more complicated situation: From a tape recording of BACH, KdF, Cp XVIII, a sequence of fragments had to be extracted, the starting point of which was determined by regular triolic/quintolic division on the **notation** level, while the duration was determined in „centimeter of tape". These fragment were arranged in time according to some other, again metric raster.

[12] The requirement of well-foundedness would suffice, but since complicated fix-point semantics seem not to be necessary, we make this harder requirement.

[13] Of course only the „correctly typed" expressions are part of the language. Since the overall typing requirements are too complex to be expressed by mere indexing, the indices have been omitted.

context conditions the parameter scales and the scale constructors impose on the types of their numeric parameters. Only the documents conformant with these are of course valid.

Please note that grammar in figure 4 is meant as the grammar of an *abstract* syntax, ie. does abstract from precedences etc.

Practical Application: The construct *EquivalenceMap* reflects the fact, that sometimes during a refinement process $1 : n$ or $n : 1$ identities have to be stated between abstract events, but that $m : n$ equivalences should be avoided :

E.g. a chord, which on notation level or as a notion of harmonic analysis is a *single* event with a pitch parameter of a `set`-type range, has to be modeled as a multitude of events w.r.t. time and intensity, when describing a certain piano *interpretation*, — a single node with a triller ornament must be split to many events in a Midi realization, etc.

E. Stacked Semantics and Packed Representation

According to our experience in language transformation, eg. using XSLT-scripts (cf. [2]) and visitor concepts, it seems outmost important that the semantic model (a) is „flat", i.e. realizes nothing more than an orthogonal function similar to

$$AbstractEvent \times ParameterIdent \;\nrightarrow\; Scale \times ParameterValues$$

and (b) that it does not make any use of implicit context information on the coding level, beyond the dependencies given explicitly by scale object creation and reference.

On the *coding* side the situation is different, and some hierarchical constructs can be quite useful, e.g. for (a) saving disk space and transmission time, and (b) to increase readability when working directly on the XML-representation, e.g. for debugging purposes.

Our approach offers two layers of hierarchy, namely (a) scoping of parameter functions and (b) track overlaying.

1) Scoping of Parameter Functions: Since normally certain parameter functions do have *identical values* for a certain range of events (or even for all), our grammar provides a means for scoping parameter values over *sets* of events and further parameter function specifications. So you can write (with the constructor ranges indicated schematically by indentation) :

```
scale 17
 http://ISO/music/pitchcodes/welltempered
scale 18
 http://ISO/music/durations/metric
           /sumsOfRationals

track 0  2002/05/30-15:12:11  mytool
param track 0
  param scale pitchclass 17 on
    param scale duration 18 on
      param value duration 1 4 on
        param value pitchclass c on 211
        param value pitchclass d on 212
        param value pitchclass e on 213
      param value duration 3 16 on
        param value pitchclass f on 214
      param value duration 1 32 on
        param value pitchclass g on 215
        param value pitchclass f on 216
  param value octave 1 on 211 212 213 214
                          215 216
```

2) Stacking of Tracks: Since in most cases transformations and further processing of a given musical document by a second tools implies only the modification of certain parameters, or even only attach additional information, leaving existing data untouched, we make use of the concept of *data tracks*: Each track may define parameter functions, while „inheriting" all parameter function values not overridden in this track.

This is especially useful for attaching eg. graphical rendering hints, results of analysis or interpretation data to a given musical score.

Together with the appropriate constructors for scales, and with the mechanism of defining scales and parameters separately, we can eg. create a *transposition* of the score above by just adding . . .

```
scale 19  http://ISO/music/pitchcodes
                /pitchclasstransposition

track 1 modifies sameDoc 0
   2002/05/30-15:12:11  mytool
param track 1
  param scale 19 +1 on  211 212 213
                        214 215 216
```

The same mechanism can be used (1) to derive instrumental parts from whole scores by exchanging the *geometric* scales, (2) to keep different *versions* of a composition in one document, (3) to create *interpretations* from notations by adding microintervalic derivation information and micro-timing, etc.

F. Mapping it to XML

Mapping of the language given in figure 4 to XML is straightforward. In a first step all nonterminals beginning with a capital letter are transformed into an appropriate *free type* definition, then for each constructor of this free type an XML tag is chosen.

Practical Application: Of course this mapping is in reality nontrivial, and different aspects have to be considered, some of them listed in the following:

1 —

The structure of XML-objects following the XML kernel specification is

$$Node == Tag \times (QIdent \nrightarrow Text) \times \mathtt{seq}(Node \cup Text)$$

If it were

$$Node == Tag \times (QIdent \nrightarrow \mathbf{Node}) \times \mathtt{seq}(Node \cup Text)$$

then XML-ATTRIBUTEs would be perfect for encoding all the nonpositional and associative part of the information, while the „content" would carry the position and context-dependent part. But since the designer of SGML decided thirty years ago to restrict the range of all attribute functions to mere text, all pure-associative information which targets into *structured* domains cannot be expressed by the XML-ATTRIBUTE mechanism.

So whenever we want an open architecture which also allows the encoding of *expressions* of some sub-language to describe parameters, and we do not want these expressions to „fall out" of the structure of the document (of course you can always put expressions into pure

MusicalDocument	::=	`eventHi` $\mathbb{N}$ `trackHi` $\mathbb{N}$ (*dataItem*)+
dataItem	::=	*TrackDeclaration* | *ParameterSpec*
	|	*EquivalenceMap* | *ScaleDeclaration* | *UriAliasDeclaration*
ScaleDeclaration	::=	`scale` $\mathbb{N}$ *scaleConstructor ParameterSpec*
TrackDeclaration	::=	`track` $\mathbb{N}$ (`modifies` *TrackId*)? *Date ToolIndication*
TrackId	::=	(*URI* | `sameDoc`) $\mathbb{N}$
ToolIndication	::=	*Vendor* (*Toolname*)? (*Version*)?
EquivalenceMap	::=	`equiv` $\mathbb{N}$ `to` $\mathbb{N}+$
	|	`equiv` $\mathbb{N}+$ `to` $\mathbb{N}$
UriAliasDeclaraion	::=	`alias` *Ident URI*
ParameterSpecs	::=	`param track` $\mathbb{N}$ *ParameterScope*
ParameterScope	::=	*ParameterSpec* `on` (*ParameterScope* | *abstractEventId*)+
ParemeterSpec	::=	`param scale` *pIdent* $\mathbb{N}$
	|	`param value` *pIdent* ($\mathbb{N}$ | $\mathbb{Q}$ | $\mathbb{R}$ | *Text* | *abstracEventId* | *Set*)+
Set	::=	`set` $\mathbb{N}+$ | `set` $\mathbb{Q}+$ | `set` $\mathbb{R}+$
abstractEventId	::=	$\mathbb{N}$
pIdent	::=	*URI* | *Ident*
scaleConstructor	::=	*URI* | *Ident*

Fig. 4. A possible language for Musical Documents

text, but all XML-tools will not be able to process the expressions automatically any more), we should minimize or even avoid the usage of ATTRIBUTE-based encoding. Indeed these are not necessary from the viewpoint of mathematical semantics, since there are always equivalent representations using ELEMENT tags.

2 —

Further more we stress again that there should be minimal usage of *context* information, e.g. when giving entry time by relative distances, the references to the pre-going event should be made *explicitely*. This will substantially facilitate „external" processing of musical documents by standard processing tools, and since the XML encoding has to be converted into some internal format anyway, we do not see any advantage in giving any semantics to the document relative position of an item, which is just defined on the coding side.

3 —

The reader may have noticed that this paper, while speaking on XML, does not contain a single pair of angle brackets.

This is by intention.

According to our experiences with several standardization processes we strongly recommend to use professional devices of *sufficient abstractness*whenever discussing language specifications, like „abstract syntax", „denotational semantics" etc. Even the *concrete* front-end representation is much more easily specified as eg. a „constructor algebra" or by EBNF, the mapping of which into XML tags and content models is almost canonical.

IV. Filling in the Genericity

What is **not** covered in this paper and by our approach, is the „substantial" standardization, ie. the definition of (a) the semantics of the different scale constructors and (b) the meaning of standardized parameter identifiers.

The existing standards or standardization efforts (eg. [3], [4]) contain valuable definitions which should be taken as a starting point.

The track concept presented herein allows to mix standardized and proprietary notions *ad libitum*. So the standardization process could be performed in an incremental way. Different ways of encoding the same parameter function may even be permitted to co-exist, because sufficiently precise definitions of their semantics will always induce a canonical mapping between them.

Further more it is possible to attach arbitrary *ad-hoc* parameters to any given structure: E.g. a composer might want to introduce a parameter like „`color`" to the printing of his notes, or even a parameter „`taste`", taking values from `sweet` to `salty`[14].

A more senseful application would be to attach *arbitrarily* defined results of musical analysis or auxiliary parameters for a special synthesizing algorithm to any existing, predefined or „downloaded" musical structure, so lifting the expressive power of compositionality, which is inherent to XML-documents and -declarations, to the level of æsthetic application.

References

[1] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, and Eve Maler. *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation, `http://www.w3.org/TR/2000/REC-xml`.

[2] James Clark. *XSL Transformations (XSLT) Version 1.0*. W3C Recommendation, `http://www.w3.org/TR/1999/REC-xslt`.

[3] ISO/IEC `ftp://ftp.oml.gov.pub/sgml/WG8/SMDL/10743.ps`. Standard music description language (sdml). Technical Report DIS10743.

[4] ISO/IEC `http://www.musique.umontreal.ca/pesronell/Belkin/NIFF`. Notation interchange file format. Technical report, 1995.

[5] Guerino Mazzola. *Geometrie der Töne - Elemente der Mathematischen Musiktheorie*. Birkhäuser, 1990.

[14] `;-)`