Approaches to Using Rules as a Composition Method
Öörjan Sandred

Available online: 10 Nov 2009

To cite this article: Öörjan Sandred (2009): Approaches to Using Rules as a Composition Method, Contemporary Music Review, 28:2, 149-165

To link to this article: http://dx.doi.org/10.1080/07494460903322430

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.tandfonline.com/page/terms-and-conditions

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan, sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
Approaches to Using Rules as a Composition Method

Örjan Sandred

This paper will give an overview of the author’s experience in using rule-based computing in music composition. After discussing both imposed and voluntary constraints in existing music, a very short description of a rule-based computer system will be given. The concept of musical dimensions will be discussed, and used to illustrate the complexity of implementing musical structures into a computer system. A central section in the article will focus on different approaches to the design of musical rules. This will be followed by a discussion of the relationship between pitch and rhythm, and the role of motifs and gestures. Finally, two examples from one of the author’s compositions will illustrate how rules can be used to formalize music.

Keywords: Computer-assisted Composition; Rule-based Composition; Constraint-based Composition; Constraint Programming; Music Analysis

Constraints and Tradition

Music can only exist within a framework of constraints. Composers voluntarily utilize some of these constraints. Other constraints are forced by common practice in music. The notation system commonly used in Western Art music imposes several constraints for how composers write music. Two obvious restrictions are the use of 12 pitches per octave, and the concept and organization of whole and half tone steps (the origin of this system for organizing pitch is older than the notation itself). It is, however, possible to stretch how we use the current notation system by extending the number of pitches per octave by using accidentals for microtonal pitches, and it is possible to ignore the modal organization of pitch. By extending the system we are not able to eliminate the constraints a notation imposes; we only redefine its limits.

The notation of musical time is also restricted by tradition: musical time is commonly thought of as being proportional. Durations in a traditional score have a very precise relationship to each other. They relate as fractions of a whole duration unit (i.e. the whole note). By using proportional notation, a composer chooses a concept for time representation that will force restrictions on the music.
Composers have used their creativity to stretch the framework of constraints. According to Moles (1966, p. 105), ‘The evolution of music seems to be a methodical violation of previously accepted rules.’ The fact that composers tend to stretch what is acceptable within a framework does not mean that a framework is not desired. For example, when composers abandoned tonality as a framework, it was soon replaced by other systems with other rules (for example, serial or stochastic techniques).

**Constraints as a Method**

Anyone who has studied music theory has been exposed to the method of describing a musical style in terms of rules. These rules are typically simple, and only describe a single stylistic detail. A single rule will therefore have limited value; only the combination of all rules gives a valid representation of the style.

Fux’s (1943) theoretical treatise *Gradus ad Parnassum* originally dating from 1725 is a good example of how a set of simple rules can explain complex polyphonic relationships where pitches, durations and meter are interdependent. The combination of all Fux’s rules will give a good description of the sixteenth-century vocal polyphonic style.

When computers were introduced in composition, composers were challenged with how to formalize musical structures. Hiller’s early experiments in his *Illiac Suite* applied rules inspired by Fux’s theoretical work (Hiller & Isaacson, 1957, 1958). A later and musically more mature example of a rule-based formalization is Lindberg’s piece *Engine* for orchestra (Lindberg, 1996). Lindberg’s method is also inspired by Fux’s work in the way the individual rules are designed (see below). Just as Fux’s rules give a complete description of a musical style, Lindberg’s rules do too (but in Lindberg’s case the rules are only valid for one piece). Lindberg’s rules constrain how pitches can be selected for events in a pre-composed rhythm score.

**Rule-based Computing**

If musical structures can be formalized by sets of rules, computers can generate them. A programming paradigm called constraint programming can be used to find values for sequences of variables that fulfil given rules. The variables can only be assigned a value from a domain of possible candidates. For example, if a domain contains the candidates (1 2 3), the solution to a sequence with 5 variables could be (2 2 1 3 1). In a musical application, we can represent a sequence of pitches as a sequence of variables. The domain will then be a set of all pitches that can exist inside the sequence. For example, a domain that contains the pitches (D E F# G A B C#) will result in a sequence of pitches within this mode (for example, a sequence with 6 variables could be D D A B G F#). The solution to a musical constraint problem can in this way be a melody.

A rule can define a relationship to a fixed value; for example, ‘the melody has to start on F’. Rules can also state relationships between variables; for example, ‘pitch
classes are not allowed to be repeated’ or ‘melodic intervals are not allowed to appear more than once’. The latter two rules are context dependent, and less trivial to solve. We need to step through our sequence of pitches and step-by-step check that these rules are fulfilled.

If we let a constraint program search for a sequence of 12 pitches with the above three rules, it will generate 12-tone all-interval series (Figure 1).

A domain can represent other musical parameters than pitch. If the domain contains durations, we would search for a sequence of durations (i.e. a rhythm).

![Figure 1](image_url) Three different solutions from a constraint program programmed to generate 12-tone all-interval series.

**Musical Dimensions**

The 12-tone example in Figure 1 is a simple problem for a computer to solve. It only constrains a sequence of pitches. Rhythm is not part of the formalization, and there are no other voices that influence the pitch order.

Musical structures are generally very complex. The main reason for this is that there are interdependencies between different musical parameters. Typically the choice of a pitch depends on its duration and its position within the metric hierarchy. For example, in the study of traditional vocal polyphony one rule is that half notes on unaccented beats may have dissonance, but only as passing notes. This rule explains a detailed relationship between duration (half notes), metric position (unaccented beats) and pitch (a dissonance as a passing note depends on the preceding and the subsequent pitches as well as the simultaneous pitches in other voices).

In this article I will use the term *musical dimensions* to describe the different parameters for musical events. A musical dimension (the way I use the term) can be thought of as a sequence of values or quantities. A melody can be built of a sequence of pitches and a sequence of durations. These two dimensions often also relate to their position within the metric dimension. If there is more than one voice, the other voices create additional dimensions. Two-part music will then have five dimensions: the first voice has one pitch dimension and one rhythm dimension, the second voice has one pitch dimension and one rhythm dimension, and they both share the same metric dimension.
The complex dependencies between musical dimensions make it hard to master musical structures. Students of vocal polyphony who study the above-mentioned rule for consonance and dissonance for half notes could considerably simplify an exercise by only using half notes when writing their melodies. This is one of the pedagogical ideas behind species counterpoint: by making two dimensions static (rhythm and meter), the student can focus on mastering another dimension (pitch as consonance and dissonance).

Musical structures are complex to assemble for humans, and they are also complex for computers to process. Regardless of computational technique, it is hard to generate two or more interdependent dimensions at a time algorithmically. As a consequence, computers have often been used to generate pitch or chord sequences (i.e. only one dimension) for which a composer later superimposes a rhythm. In some cases (for example, in Lindberg’s Engine mentioned above) a rhythm structure was first generated independently of pitches. As a second step, the computer selected pitches for the rhythm structure.1

When composing my own music, I rarely follow a scheme where first one dimension is composed and then another dimension is superimposed. Dependencies are often more complex than that: a pitch might depend and adjust to a rhythm, but just as often a rhythm depends on and will have to adjust to a pitch. Therefore the method of creating one dimension before (and independently of) the others is not optimal for my own work. I would prefer a more flexible system that allows for all dependent dimensions to be created in parallel.

**Multi-dimensional Rule-based Computing: PWMC**

To be able to compute musical structures with more than one dimension, I developed the PatchWork Musical Constraint system (PWMC) in 2004. PWMC is an extension of PatchWork GL (PWGL), a visual programming language specialized in Computer Assisted Composition developed at the Sibelius Academy (Laurson, Kuuskankare & Norilo, 2009).

PWMC is a system that is dedicated to rule-based computing. It is based on conventional constraint programming techniques, and extends these with the knowledge of musical dimensions. A conventional constraint solver would only be able to handle one musical dimension. PWMC makes it possible to search for polyphonic musical structures where pitches, rhythm and meter are all unknown. PWMC uses the same basic structure as a conventional constraint solver: there is a domain of allowed values (in the case of PWMC the domain is a mix of pitches, durations and metric units), and there is a set of rules that state the relationship between the variables (i.e. between the durations, the pitches and their metric positions).

Finding solutions to multi-dimensional constraint problems is a complex task. The constraint solver that the current version of PWMC uses builds a solution step-by-step. The exact order for how the pitches and their durations are added to the
solution is context-dependent. The user can interact to find the optimal strategy for this order. A bad strategy can result in that it takes a very long time to find a solution.

All examples in this article were generated using the PWMC system, but it is beyond the scope of this article to fully explain and describe the PWMC system.

**Rule Design in Musical Constraint Programming**

An important strength of working with constraints in computer-assisted composition (CAC) is the modular nature that rules have: a complex musical description can be broken down into many sub-descriptions with simple rules that are easier for the human mind to grasp.

There are several approaches to the way rules can be designed. All rules below are discussed from the perspective that they are implemented in a rule-based computer system (as described above, in Rule-based computing). Only the first group of rules is commonly used in constraint-based music programming today (although they can be found in other programming techniques).

**Rule Design 1**

The rules used in Fux’s description of sixteenth-century vocal polyphony restrict details within a musical structure. Fux typically focuses on the individual notes: What melodic interval can I choose to start from a quarter note? How can I position a dotted half note within a bar? The network of rules illustrates how the treatment of consonance and dissonance depends on an event’s metric position and on its duration. The rules also outline how melodic intervals relate to their metric and rhythmic context (Figure 2). Fux’s rules are, however, sparse on the description of form and overall progression. In order to be able to create a good stylistic imitation, students need to develop a sense for qualities outside of Fux’s rule system. If the rules are implemented by a constraints program, the result would be more successful if there would be some additional information regarding the form of the sequence; for example, if the rhythm would be given and only the pitches need to be found.

![Figure 2](image-url)

*Figure 2* In this example, only one rule-type is used to restrict how pitches and durations can be combined. The rule is a typical ‘Fux-style’ rule: it restricts what melodic intervals that are allowed to start from notes of certain durations. The rule in this example restricts melodic intervals from quarter notes to only be octaves, from eighth notes and dotted eighth notes to only be fourths, from sixteenth notes to only be minor or major seconds and from eighth note triplets to only be minor or major thirds. There is also a rule restricting syncopations and the subdivision of metric beats. Beside these restrictions, pitches and durations were picked at random.
Rule Design 2

A contrasting rule design to Fux’s approach is to look at the overall qualities of a sequence. We might have a preference for certain characteristics. For example, if we want a rhythmically active music, we might prefer sixteenth notes to quarter notes. We can formalize the distribution of durations and estimate that of all the durations in our piece, we would like 50% to be sixteenth notes, but only 10% to be quarter notes. This type of rule gives many options for the individual note lengths, but forces the overall rhythmical character in a certain direction. As a consequence, the rule will not give a detailed stylistic control, but it can help in shaping the character of a longer sequence.

There are several ways to design a rule for the distribution of events. One way would be to always keep the sequence as close as possible to the ideal distribution. Another way would be to estimate the maximum number of each of the note lengths that is allowed in the sequence, and not to allow them anymore when their maximum numbers are reached. In Figure 3, the pitches follow a distribution rule: the low C is the most common pitch and the high B is the least common (see Figure 3 for the exact distribution of pitches). The distribution rule is designed according to the first method described above. The rule is set to accept ±2% deviation from the given distribution. The result is a sequence that gravitates towards the lower pitches; for more information, please see Sandred et al. (2009).

Figure 3 A distribution rule assures that the distribution of the pitches is close to a given probability table: 15.4% of all pitches should be C, 14.1% should be C#, 12.8% D, 11.5% Eb, 10.3% E, 9.0% F, 7.7% F#, 6.4% G, 5.1% Ab, 3.8% A, 2.6% Bb and 1.3% B. The rule is set to accept a deviation of maximum ±2%.

The distribution of events is just one out of many possible techniques for controlling the overall character of a sequence. Other stochastic techniques such as Markov processes can be transformed to suit rule-based computing. There are also techniques that are not based on stochastic principles that fall into this category. One example is profiles. A profile can outline the melodic or rhythmic progression. A rule that forces a dimension to follow a given profile will influence the musical direction. An example of the latter is a method proposed by Schilingi: a melodic profile derived from a musical example is imposed heuristically on a sequence. Other rules (that for example restrict melodic intervals or harmonic relations to other voices) are imposed as strict rules on top of the profile rule. The result is a sequence that has a profile that resembles the original musical model, but also fulfils additional restrictions.

Rule Design 3

A musical structure is often constrained by hierarchical relationships between events within the structure itself. Music is full of hierarchies. The nature of the metric
dimension is to define rhythmic hierarchies, where some events are more important and therefore often more restricted than others (many of Fux’s rules above refer to the metric hierarchy). A composed metric dimension is not necessarily perceived; a composer needs to be aware of how meter is expressed through pitches and rhythm. The notation system allows very complex metric structures to be notated, and rules are well suited to limit the types of metric subdivisions and syncopations that are acceptable in a music style (Sandred, 2000).

We can also create other types of hierarchies than metrical ones. Letting one voice restrict another can create hierarchies between voices. In Figure 4, the two top voices (played on two drums) are both constructed as rhythmic patterns: after four durations the rhythms repeat themselves (the two patterns have different lengths). The two percussion voices restrict the rhythm in the bass line: every starting point for a note in any of the two percussion voices will force a new note to start at the same time point in the bass line. The bass line is built of eighth notes and sixteenth notes. As a consequence of the hierarchies, eighth notes are rare in the bass line. An additional rule restricts eighth notes to only be performed as a low C (other pitches are picked at random).

![Figure 4](image)

**Figure 4** Hierarchical rules between the percussion voices and the base line restrict the rhythm in the base line.

**Rule Design 4**

Existing music examples can serve as a model for new music sequences. The examples can be generalized through music analysis (for example, an analysis of the harmonic progression). Rules can refer to the analysis and apply its characteristics to the new music sequence.

It is important to understand that all methods for music analysis have their limitations: only a limited number of characteristics can be captured through analysis. An analysis might contain very limited information of a musical style. Often it is necessary to combine a rule that is generated from analysis with other types of rules to make musical sense.

There are several analysis techniques that are suitable for computers, and therefore can be used in rule-based computing. These include analysis of the application of traditional counterpoint rules, and analysis of the distribution of events as discussed
under Rule Designs 1 and 2 above. For example, a computer can count how many
times given rules fail in an existing score and in this way give an analysis of whether
or not they are valid. An analysis of the event distribution is simply achieved by
calculating the probability table for the events in a sequence.

A recent development in music analysis is the morphological descriptions of
melodic profiles (Schilingi & Voisin, 1997). Two examples of techniques for
morphological descriptions are the new/old-analysis and the energy profile analysis
(Aralla, 2003). The new/old-analysis relates to information theory. In information
theory, the predictability of an event is a measurement of its information content. If
an event is predictable, it does not carry any new information. An event that is
repeated over and over again will be perceived as more and more predictable. The
opposite situation is equally predictable: if every new event in a sequence is different
and unique, it will be of no surprise if the next event also is unique. In the context of
everything in a sequence is unique, an added, unique event does not carry any new
information.

The measurement given by the new/old-analysis gives can be understood as the
predictability of events in relation to the context in which they appear. The quantity
that is measured is labelled ‘newness’ (i.e. how new an event will be perceived). A
sequence with a repeated value will result in a new/old-analysis where the third event
has zero ‘newness’. The opposite case (where every new event is different from all
existing events) will result in a new/old-analysis with a gradually falling ‘newness’.

In most melodies, unique pitches are mixed with re-occuring pitches. The new/
old-analysis is a tool to describe the impact the individual pitches will have in their
context. The necessary algorithmic processing of a sequence in a new/old-analysis will
not be discussed in this article; for more information, please see Aralla (2003).

An energy profile is another way to display the outcome of a new/old-analysis: it
describes the degree of difference in ‘newness’ between neighbouring events (Figure 5).
A sudden change in the new/old-analysis will be visible as an energy peak in the energy
profile. The energy profile is often able to point out differences in melodic behaviour. It
can also point out where a melody modulates or where new pitches are introduced. The
energy profile has proven to be useful both for tonal and atonal music.

An energy profile can be used in rule design. The rule would then force the
characteristics of an existing energy profile on a new sequence. The energy profile
influences the event ordering. It does not carry any information regarding tonality,
dissonance/consonance or metric hierarchy. It is technically possible to apply an
analysis of the energy profile to the rhythmic dimension; however, the musical
meaning of this is less explored.

Rules that have a weak definition on their own can provide valuable additional
directions to other rules. Figure 5 illustrates this: the rule for the energy profile only
has an impact on the order of events. Most stylistic characteristics, such as tonality
and voice leading, are not inherent in this rule. Only by supplementing the rule for
the energy profile with two simple rules regarding pitch (Figure 5), is it possible to
rebuild the Bach theme that the energy profile is based on.
The four examples of approaches in rule design above are not an exhaustive list of possibilities. They only point out that there are many more approaches in rule-based computing than what has commonly been explored in musical constraint programming.

The Symbiosis of Pitch and Rhythm

In rule-based composition, the constraints we impose on our musical structures must have the potential to shape aesthetic qualities. For this reason, I occupied myself for several years with the question of what a good rhythm is. It is not obvious how to formulate rules for durations that result in a satisfying rhythm.

One of my conclusions was that rhythm should not be discussed as a phenomenon in its own right. Rhythm comes very rarely in such a pure form that it is unaffected by other musical dimensions. Even in pure percussion music, timbre plays a role. The colour of the events will have an affect on how a rhythm is perceived. A good rhythm must therefore be seen in the context of the pitches/timbres it will carry.

My belief is that rhythm in many cases has been composed as a result of choices made in the pitch dimension. There are clearly exceptions; if rhythm is separately generated from an algorithm, there is no conceptual link to the pitch dimension. However, in the end, the rhythm will be performed with timbres or pitches, and the listener’s perception of rhythm will be linked to these parameters. For example, in an isorhythmic composition technique, rhythm is treated separately from pitch. An important aspect of isorhythmic structures is that when the rhythm pattern (the

Figure 5 The graph shows the energy profile of the theme from Bach’s Fugue XVI from *Das Wohltemperirte Klavier*. The main rule in this example forces the characteristics from this energy profile on a melody with 22 pitches. The energy peak is also constrained to have the pitch c sharp. Two more rules restrict the sequence: one rule forces the pitches on accented metric points to be consonant to a given harmonic progression (gm/cm/D/gm/gm/c/gm/gm/D/gm/A/D), and one rule specifies that melodic intervals from sixteenth notes can only be minor or major seconds. This information is enough to recreate the pitches for the theme from Bach’s Fugue XVI from *Das Wohltemperirte Klavier*. The rhythm was not generated but given as a restriction.

The four examples of approaches in rule design above are not an exhaustive list of possibilities. They only point out that there are many more approaches in rule-based computing than what has commonly been explored in musical constraint programming.
talea) repeats, it will be colored by new pitches: each repetition of the rhythm will therefore be perceived differently. By separating rhythm from pitches, isorhythmic techniques explore the effect of recombining the two dimensions again.

To clarify what I mean with the symbiosis of pitch and rhythm, I will give an example: In Figure 2, a single rule inspired by ‘Fux-style’ rules is all that is used to generate a simple melody (there is an additional rule to avoid complex positions of durations in relationship to the metric dimension). This seemingly simple rule creates a surprisingly elastic melody. The melody has its obvious weaknesses (form being one), but the single link between the pitch and rhythm dimensions already gives a first fragment of a musical expression.

The example in Figure 2 illustrates how music is perceived as a combination of musical dimensions. The melodic octave leaps give the quarter notes a characteristic musical expression. In this example, the rhythmic qualities can only be understood in the light of the melodic intervals they express. This is why the rhythm has its elastic quality. The rhythm is here the result of decisions in the pitch dimension (and vice versa).

Other types of links between pitch and rhythm give different types of expressions (see for example the bass line in Figure 4).

Musical Building Stones

Up until now, we have assumed that the only constraint in our rule-based composition method, beside the notation system itself, has been the set of rules we apply to pitches, durations and metric units. The secret of a successful musical language would then exist in well-designed rules. Composers have, however, had other approaches when structuring the musical material than using rule restrictions. One possible strategy has been to view the musical building blocks on a higher level than on the level of single notes. Higher-level groupings of pitches and durations create motifs and gestures.

In musical styles where motifs and gestures are valid concepts, their use can simplify the composition process: it is often easier for a composer to assemble a piece of music out of gestures and motifs than to build the whole piece note-by-note. The same approach could be possible in rule-based computing: rules often give a very detailed, low-level control of the musical result. If motifs and gestures could be outlined before the low-level control is applied, it would increase the possibility to reach a desired result.

Gestures and motifs represent fragments of the musical language in a piece. Their design will have a great impact on the final musical structure. The concept of gestures and motifs can be applied by rules, but it is more efficient to implement musical shapes already in the domain. The domain would then contain groupings of pitches and durations instead of single pitch and duration values. A group in the domain cannot be broken down into its individual elements. For example, if the pitch C3 only exists in a group (C3 E3 G3) in the domain, then C3 will always be followed by E3.
and G3 in the final piece. The first rule-based computer system that implemented the concept of motifs in the domain was the OpenMusic Rhythm Constraints library (OMRC) by the author (Sandred, 2000, 2003).

It is possible to combine pitch and rhythm motifs with individual pitches and durations in a domain. Pitch motifs may or may not specify exact pitches. A pitch motif that does not specify the exact pitches only outlines its melodic contour. A pitch motif can for example be defined as a minor third up followed by a pure fifth down (i.e. a broken triad). If this shape is part of the domain, the melodic line can at any point follow it (if the rules allow it). Examples will follow below.

Examples from Whirl of Leaves

Two music examples will illustrate how music can be composed using a rule-based method. The two examples both come from my composition Whirl of Leaves for flute and harp. The piece was premiered at the UltraSchall Festival in Berlin, January 2009. Whirl of Leaves was entirely composed using rule-based computing.

Example 1

The first example is taken from the third section of the piece, starting at measure 50. The meter is fixed to 4/4. The domain contains all the pitches and durations that can exist in the flute and harp parts: the flute can have any pitch between F4 and Ab6, and the harp can have any pitch between C2 and C6. These ranges give access to both low and high registers in both instruments (avoiding the extreme ends). The durations in the domain are grouped into rhythm motifs. These motifs can be seen in Figure 6: the idea is to give the harp fast-moving gestures with different subdivisions of the beat (all rhythm motifs have the total duration of a quarter note). The flute domain has two motifs that will match the subdivision of the first and fourth motifs in the harp domain. The domain for the flute also contains four different rests to choose from.

In addition to pitches and rhythmic motifs, the domain contains information about chords. There are five chords with corresponding modes that constitute the base for the harmonic structure (Figure 7). Three long durations will be the building blocks for the harmonic rhythm (see Figure 6). The program will determine the order of the durations and the chords for the harmonic progression.

There are 12 rules restricting the score. Two types of rules are used in this example: strict rules and heuristic rules. Strict rules have to be followed without exception. Heuristic rules only give preferences to certain characteristics of the music.

Hierarchical rules (described above) define relationships between the harmonic rhythm, the rhythm in the flute part and the rhythm in the harp part.

Rule 1: Every chord change in the harmonic rhythm triggers a new event in the flute part.
Rule 2: The events in the flute part trigger events in the harp part.
The two rules force every harmonic change to be expressed as new events in both the flute and the harp. The second rule also forces the rhythm in the harp part to follow the flute when the flute plays, but when it has rests the harp is unaffected by this rule.

The rhythm in the flute part is controlled by two rules balancing each other. Rule 3 is heuristic and can only be followed when rule 4 allows it.

**Rule 3**: At the points of chord changes in the harmonic progression, the flute has to play a motif using the first rhythm in Figure 6.

**Rule 4**: Rests are always preferred in the flute.

The result is that the flute will accentuate every chord change by playing a sextuplet gesture. Most of the other time the flute will have rests.

The two rules force every harmonic change to be expressed as new events in both the flute and the harp. The second rule also forces the rhythm in the harp part to follow the flute when the flute plays, but when it has rests the harp is unaffected by this rule.

The rhythm in the flute part is controlled by two rules balancing each other. Rule 4 is heuristic and can only be followed when rule 3 allows it.

**Rule 5**: It is preferred that the rhythm motifs (Figure 6) are not immediately repeated.
Pitches in both instruments are linked to the harmonic progression.

**Rule 6:** Only pitch classes from the current mode in the harmonic progression (Figure 7) can be used in the harp.

Every mode in Figure 7 is designed so that it is possible to play all its pitches without changing the pedals on the harp.

Pitches on beats are more restricted in the harp.

**Rule 7:** Only pitches from the current chord in the harmonic progression can be used on beats or in-between beats (i.e. events appearing on an eighth note grid) in the harp.

Rules 6 and 7 create a harmonic structure that is expressed through the metric structure. The chord progression will be clear thanks to these two rules.

Pitches in the flute are linked to pitches in the harp.

**Rule 8:** Simultaneous pitches in the flute and the harp have to be of the same pitch class.

Since the flute mainly will play the sixteenth note sextuplet at chord changes, this rule will only have an effect at these points. The flute will thus accentuate chord changes (Rules 3 and 4) by doubling the pitches in the harp (Rule 8). Both instruments will play a sextuplet at these points (Rules 2 and 3).

A few rules restrict the melodic line in the harp.

**Rule 9:** Pitches are not allowed to be immediately repeated.

**Rule 10:** Two steps in the same direction have to be followed by a step in the contrary direction.

**Rule 11:** Melodic intervals have to be smaller than a fifth unless the harp plays in homophony with the flute. In the latter case the melodic intervals can be at the maximum an octave + fourth.

The last two rules create a melodic line that stays in a narrow register until it is doubled by the flute. At that point it may jump to a new narrow register.

One additional rule restricts pitches in the flute.

**Rule 12:** Pitches are not allowed to be repeated immediately.

Since the pitches in the flute follow the harp (Rule 8), Rule 9 already has some effect on the flute line. Rule 12 ensures that immediately repeated pitch classes will be put in different octaves.

The final score for this example can be seen in Figure 8. The two sustained, low C sharps in the flute were added by hand after the computer had generated the score.
Example 2

The second example is taken from the second section of the piece. It illustrates how a domain may contain pitch motifs only specified by their profiles (i.e. they can be transposed to any pitch).

The rhythm domains for both voices in this example are built of individual durations (and one rest), shown in Figure 9. The pitch domain for the harp is close to identical to the previous example (the same range of individual pitches), but the pitch domain for the flute is different: it contains nine transposable pitch motifs (Figure 10). All motifs have 4 pitches (=3 melodic intervals) and the distance from the highest to the lowest pitch is always an octave + fourth. Transposable pitch motifs restrict the melodic shapes. From the viewpoint of the domain, any pitch can exist in the flute part as long as it fits into one of these shapes.

In addition to the design of the pitch motifs in the domain, four rules control the flute melody.

**Rule 1:** Pitches cannot be lower than C4 (the lowest pitch on the flute) or higher than F6.

**Rule 2:** Every sixth event has to be a rest.

**Rule 3:** Pitches higher than G5 have to be at least a half note long.

**Rule 4:** Pitches G5 or lower have to be shorter than a half note.
The first rule is necessary to assure that the pitch motifs do not walk out-of-range of the flute. The second rule divides phrases. The third and fourth rules link the choice of rhythm to pitches (as discussed under ‘The symbiosis between pitch and rhythm’ above): they will create melodies that ‘hang’ in a higher register.

There are two rules for the melodic line in the harp.

**Rule 5:** Pitches cannot be repeated immediately.
**Rule 6:** Melodic intervals have to be smaller than a major sixth.

The main melody will be in the flute: the pitch motifs in the domain are the most important restriction for shaping the pitches in the flute. The melodic line in the harp is less restricted. In addition to Rules 5 and 6, the main constraint for the pitches in the harp can be found in Rule 8 below.

![Figure 10](image_url)

**Figure 10** The pitch domain for the flute part. The motifs can be transposed to any pitch.
A few rules restrict the relationship between the voices. There is a hierarchical rule between the flute and the harp, similar to the one in the first example.

**Rule 7**: Every event in the flute triggers an event in the harp (the harp might have more events in-between the events triggered by the flute).

A difference to the previous example is that the domain does not contain any information regarding the harmony. Instead, the harmony is implemented through a rule. This rule compares all harmonic intervals and chords that exist in the score to a predefined list of chords.

**Rule 8**: Only harmonic intervals and chords that exist in the predefined list of allowed chords can exist in the score.

The chords to which Rule 8 refers are fixed in pitch and cannot be transposed.

In the final score we can see the effect of the rules. The phrases in the flute are always five notes long, and are all divided by an eighth note rest (the only rest available in the domain). When the flute melody goes above G5, it becomes more sustained. The exact shape of the melody is the result of linking pitch motifs from the domain: the melody starts with the ninth motif (Figures 10 and 11), followed by the sixth motif, then the ninth motif again, etc. The rhythm in the harp is linked to the flute. The pitches in the harp are a result of harmonic restrictions (Rule 8) and melodic restrictions (Rules 5 and 6).

I feel that I have reached the musical result I intended for *Whirl of Leaves*. The problem with the rule sets presented above is of another nature: there are very few considerations regarding playability within the rule sets. The first example creates a music that can be very demanding for the performers, while the second is in general very easy to play.

**Figure 11** Measures 23–30 from the final score for the second example from the composition *Whirl of Leaves* by the author. The extra pitches in the chords in the harp part are added by hand after the computer generated the score. These added pitches all fit into the chords.
Conclusion

A composer can use rule-based computing to generate a musical score. Rule-based computing has the strength of breaking down a complex problem into simple subproblems.

A score can be assembled from musical elements such as pitches and durations, or from higher-level groupings of these elements (i.e. motifs and gestures). A composer can interact with the system by shaping the motifs and gestures, and by designing the rules. A rule-based system allows the composer to experiment with the impact individual rules, motifs and shapes have when they are combined into a complex system of dependencies.

Notes

[1] As a composition student of Lindberg in 1996 (at the time he composed Engine), one of the topics we studied was the Score-pmc constraint engine inside PatchWork (a visual music programming language). The nature of Score-pmc is that it needs a complete rhythm structure before the program can assign pitches to the predefined rhythm score. The piece Engine was composed with (among other tools) the Score-pmc.

[2] This can be found in the JBS-constraints library inside PWGL.

References


