THE FORMALIZATION OF GENERATIVE STRUCTURES WITHIN STRAVINSKY'S "THE RITE OF SPRING"

Adriano De Matteis, Goffredo Haus
L.I.M. - Laboratorio di Informatica Musicale
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano

Introduction.

In this paper we show how we can formally describe generative structures of musical pieces. We consider as a particular case study "The Rite of Spring" by I. Stravinsky.

The concept of Music Object (MO) has a fundamental role in our research. An MO may be anything that could have a musical meaning and that we think of as an entity, either simple or complex, either abstract or detailed, an entity with a name and some relationship with other musical objects.

Our formalization has been made by means of an "ad hoc" arrangement of Petri Nets (PNs) and a music algebra (Haus and Sametti, 1992). Music objects are associated to places; music transformations are described by algorithms associated to transitions and are coded by expressions which are based on the set of operators and the syntactic rules we have defined. This approach allows us to describe music objects and their transformations at various levels of representation.

Our aim is to show that:

1) it is possible to find a set of general PNs that can work on a number of music pieces,

2) a set of MOs and few of these PNs can partially or completely well formalize the musical material within a composition,

3) this formalization is closely related to musical thinking and practice, because it preserves and shows the hidden structures within music scores.

A further case study can be considered the model of Ravel's "Bolero" by Haus and Rodriguez (1993) which is based on an approach very similar to this one.

The research could be useful not only as a musicological analysis but also as a demonstration of the power of this formalization approach for composing new material: for example, according to the transformations we have found.
The models we have produced have been implemented by the ScoreSynth, so we can have musical executions "testing" our modeling.
The contents of this paper include:
a) a description of the basic macro PN structures which are of common interest in the development of the Rite model,
b) a comprehensive example excerpted from the model which exemplifies how to use the macro structures described previously;
c) a discussion on how it is possible to get the orchestral score from the model of the piano score;
d) some concluding remarks about both the method and results of our research.

Basic Macro PN Structures.

In this paragraph we briefly discuss the basic PN macro structures we have found analyzing Stravinsky's "The Rite of Spring". In the following paragraphs we will show how these nets can be used for describing music structures which are hiddenly contained within the Rite score. It is suggested to the reader who is not well acquainted with ScoreSynth syntactic features to read the paper by Haus and Sametti (1992) in which all the implementation of music Petri Nets and the associated music algebra are completely described.

Juxtaposition.

This subnet concerns the model of a characteristic process in Stravinsky's music: the dynamic juxtaposition of two or more MOs. It is undoubtedly the most interesting net relative to our "The Rite of Spring" model.
The compositive procedure we want to formalize is reported by Nicolas Nabokov (1951) as he relates a session in Stravinsky's study being Stravinsky describing one of his works (Orpheus) at the piano:

"See the fugue here" he would say pointing to the beginning of the Epilogue "Two horns are working it out, while a trumpet and a violin in unison sing a long-drawn-out melody, a kind of *cantus firmus*. Doesn't this melody sound to you like a medieval vielle [a viol]? Listen..." And his fingers would start figdeting again on the keyboard. Then, coming to a passage in the Epilogue where a harp solo interrupts the slow progress of the fugue, he would stop and say: "Here, see, I cut off the fugue with a pair of scissors." And he clipped the air with his fingers "I introduced this short harp phrase like two bars of an accompaniment. Then the horns go on with their fugue as if nothing has happened. I repeat it at regular intervals, here and here again." Stravinsky added with his abitual grin, "You can eliminate this harp-solo interruptions, paste the parts of the fugue together and it will be one whole piece".

Even if Stravinsky's description does not concern the Rite directly the important fact is that this attitude was present in his mind. Besides, we know that all the analysis made on different Stravinsky's pieces prove that many of them are composed in a similiar way. Consider, for example the Boulez well-known analysis (1953).

So, the general meaning of this process consists in having a certain number of MOs each following its particular development. These developments are cut here and there and then set by juxtaposition: that is, none of the cuttings modifies what is happening in the developments, apart from interrupting their natural development; when the suspended development begins again it is as if nothing has happened. For example, if we name the MOs A, B and C we can make some cuts within them to obtain the objects A_1, A_2, B_1, B_2, C_1, C_2 and then juxtapose these results in the form

A_1 - B_1 - A_2 - B_2 - C_1 - C_2

Looking at this structure, we may think that a sequential net works well (see Figure 1). This is certainly true, but such a net would not contain information about how the piece is really structured, only about how the piece appears to us. This kind of net would lose the fact that we have stressed.

We want to show that some models are possible not only in the sense that they work, but moreover that they preserve information about the construction of the piece which is completely lost in common music notation.

We believe that this appproach is closer to music thinking. So let's see in Figure 2 the PN that realizes what we are saying. It is formed by three parallel sequences each one regarding the particular evolution of a certain MO. They can be seen easily in our figure because they begin with a place named `A_path`, `B_path`, `C_path` respectively. The different stages of development of MO A are

labeled A_n, the ones concerning B B_n, and for C C_n. These places are *play-places* ; instead the other ones have a logical function.
The actuation of the process is guided by the connections with the transitions in output from a play place with an Sgn (the name is derived from SiGNal) corresponding to the next MO we want to play. So if in the figure transition Out(A_1) is connected with the place Sgn(B_1) this means that B_1 follows A_1 in the process of juxtaposition. These Sgn places are necessary not only to synchronize the MOs evolutions but also to clear out the temporary memory from the current MO and to avoid interference between the different parallel sequences. We want to remember in fact that the interferences exist only in the sense that each part interrupts the other but do not change it. Notice that if we eliminate the connections indicating the interferences we could have three different self-sufficient pieces.
Let's see the token evolution. When the net is invoked a token is put in place In. So transition T may fire and we have a token in the Sgn place corresponding to the MO that we want to be the first playing. In our case the place is Sgn(A_1), so the first MO in our realization is A_1. Since both A_path and Sgn(A_1) have a token, transition T1_A fires, and the process associated to A_1 can occur. At the end of A_1 transition Out(A_1) fires and a token is put in both Sgn(B_1) and Carry_A. Place Carry_A has only to take into memory object A and carry it forward in its binary. A token in place Sgn(B_1) means that the process associated to B_1 is ready to be executed. Notice in fact that the only transition which may fire is T1_B since there is a token in both B_Path and Sgn(B_1), the input places of T1_B. After the occurrence of the process associated to B_1, we can again make considerations analogous to the previous ones. When all the MOs are played in the desired order a token is put in place Out and the whole process ends.
The graphic of the net makes us think of a sketch that Stravinsky himself drew when Robert Craft asked him to express his music graphically (Craft and Stravinsky, 1959). See Figure 3.

Loop.

The subnet in Figure 4 describes a typical musical structure, widely used in Stravinsky's works: the repetition of the same musical process for a certain fixed number of times.

The net has as input a certain MO and actuates the repetition. We want to underline that the presence of an MO is not indispensable. The net can be used to repeat other kind of processes whatever they may be: for example, MIDI Exclusive messages or entire sections of music.

Place `Cnt` (the name is derived from CouNTer) contains a number of tokens equal to the required repetitions. In our net they are four. Notice that this number is equal to the capacity of place `Basket` and to the multiplicity of the arc connecting `Basket` to `Way`. We see why later.

When transition `T` fires a token is placed in `Aux`. So, since transition `Alg_1` has two input places both with at least one token, it can fire and we have a token in `Obj`.

Place `Obj` can be simple or associated with a subnet. In any case the presence of a token in `Obj` determines the execution of the input MO, whether it has been modified or not.

Immediately afterward transition `Alg_2` is enabled. Its firing causes the loss of one token from `Cnt` and one from `Obj`, which consequently become empty. Moreover we have one token in both `Basket` and `Aux`. Notice that `Way` cannot fire because in `Basket` there is just one token, while the input arc has multiplicity four (that is, transition `Way` asks `Basket` for four tokens to fire). So the only transition that may fire is `Alg_1`, and we come back to execute `Obj`. Then what happens is the same as we have seen before: after the execution of `Obj`, `Cnt` loses another token while `Basket` acquires a new one, and one token appears in `Aux`. `Way` still cannot fire. We come back in `Obj` and so on till `Cnt` contains only one token and so transition `Alg_1` fires for the last time. Afterwards, this firing the last repetition of `Obj` is executed and after the fire of `Alg_2` we have one token in `Aux` and four in `Basket`. The only transition which then may fire is `Way` and we go out of the net.

Notice that such a net is useful for repeating a chosen musical process a certain number of times. But we can easily see that by using only this net we would not have the possibility of variating the repetitions, thus making them different fromone each other, except in the case we would apply the same transformation, contained in `Alg_1`, iteratively. What we can instead considered a characteristic feature of Stravinsky's music is continuous variation. Over and over Stravinsky repeats something while changing at each iteration the transformations applied. These variations do not necessarily derive from one another by an iterative process; rather each requires an independent processing. To realize this by PNs

we must have the opportunity of eventually invoking different nets at each round.
We have solved the problem by creating the net `OrdPathSel`. We discuss this net in the following paragraph.

OrdPathSel.

The net in Figure 5 gives us the possibility of coming back in a certain place of a net and actuating a process which is different, completely or partially, from the one we did the first time in the same place. Let's describe how this net works.
At the first invoking we have a token in place `In` and one in `Start` while `Sgn_1` and `Sgn_2` are empty. So the only transition that can fire is `T_1`. The result of this firing is the elimination of a token from `In` and `Start`, and the placing of a token in `Path_1`. `Path_1` is associated to a subnet that realizes an eventual first transformation of our MO. After having executed `Path_1`, the only transition that may fire is `Exit_1`; when it fires, it allows us to leave the net and to have a token in `Sgn_1` at the same time. The presence of a token in `Sgn_1` signals that the first transformation has happened and that the second one can happen when there will be a new token in place `In`, that is, when the net is called another time.
The second time the net is called we have a token in place `In` and one in `Sgn_1`. So `T_2` may fire. After the firing of `T_2`, a token is placed in `Path_2`. The second variation is executed and then, with the firing of `Exit_2`, we go out of the net pointing out by means of a token in `Sgn_2` both that the execution of `Path_2` has occourred and that the net is ready for another call.
We go on till the required number of repetitions is achieved and then come out definitively from the net.

ThemeMod.

The macro we are talking about here is represented in Figure 6. This subnet receives as input an MO and determines three different executions, with the transformations and in the order desired, of this MO. The three algorithms that realize the three transformations eventually partially or totally coincident are associated to the transitions labeled `Transf_n`.

When the subnet is called, input MO is associated with place `In`. The firing of transition `T` causes the creation of three equal copies of the MO that are associated to places `Copy_1`, `Copy_2`, `Copy_3`. Each of these places also receives a token.
Notice that the transition `Transf_1` is the only one which can fire because `Transf_2` and `Transf_3` have as input places `Synchro_1` and `Synchro_2`, respectively, which have no tokens. To place `Mod_1` is associated the effective execution of the transformation's result which has occoured in `Transf_1`. After this first repetition has been executed, place `Synchro_1` receives a token. Thus transition `Transf_2` fires and the associated algorithm generates a new repetition of the MO with the necessary variations. We can easily see that place `Synchro_1` has a logical function: the presence of a token in `Synchro_1` means that the first exposition is finished and that the second one can begin.
The other repetitions development is similar. At the end, place `Out` takes us to the main net that has invoked `ThemeMod`.
In the net we have described we have made an effort to isolate the different stages of the musical developments of the basic idea to play different executions of the same theme one after another.
In fact, from the net structure it is understood that the same MO is associated to the places named `Copy_n`; and the transformations associated to transitions `Transf_n` are working on the same MO. Logically, we first develop the three different transformations; then in a second step, places `Synchro_n` tell us how these variations should be ordered. At the end we have the three different executions played according to the requested order.

Polyphony.

The following net realizes the possibility of working with polyphonic material. In input we have a monophonic MO and in output we have a certain number of parallel voices.
A possible use of this net in Stravinsky's music consists in creating a succession of parallel chords, the movement of which is guided by the melodic line of a monophonic MO. We will see an example later. The net of Figure 7 is an example of how to realize four-part chords.
As input we have a melodic line. When transition `Transf` (which may have an algorithm associated to it) fires, a token is placed in places `P` and Obj. So both transitions `T` and `Obj_Transf` are enabled to fire, and their firing puts a token in

each of the places P_1, P_2, P_3, P_4 and Obj_Out. Transition T_Out cannot fire because one of its input places, namely Sgn, is empty. Instead all transitions Voice_1, Voice_2, Voice_3 and Voice_4 are enabled for firing. Associated to them we have four different algorithms.

In the first: P:1,$,? Leaves the input MO as is
in the second: P:1,$,?+4 Transposes a major third higher
in the third: P:1,$,?+7 Transposes a perfect fifth higher
in the fourth: P:1,$,?+12 Transposes an octave higher

When all four transitions have fired we get a token for each place Play_n. The global result is a major chord with repetition of the fundamental.

After the voices are played, transition End may fire and a token is put in place Sgn. Finally, transition T_Out fires and the net is left.

The upper process which does the Obj_Transf task has a complementary role: that is, the movement of the original MO (either transformed or as is) from the input to output of the net.

Invoking this net with different suitable algorithms associated to transitions allows us to get every kind of four-pitch combination. Adding more parallel developments allows us to get more parallel voices.

It is also possible not to use some of the available voices of the net. This is simply done by not associating any algorithm in the corresponding transitions Voice_n. When there are variations in the four-part setting of the net we point out the fact referring to the net as a n-voices Polyphony.

SynchroSplit.

The net shown in Figure 8 is useful for a compositional process which is very often used and not only in Stravinsky's music.

In this process two MOs, A and B, are logically and temporally consequent. However, the second one does not wait for the first to finish its own evolution, but partially places itself above the first one. The matter is not trivial in PNs, because when two objects are logically consequent they are also temporally consequent.

So, let's suppose we have two objects A and B. To realize a partial superimposition we use a net called SynchroSplit. This net is called by place A, anywhere in another net. As input the net has object A. When transition Split fires, a token is found in Obj_Play and then the whole MO A is played.

Simultaneously a token is placed in `Obj_Carry` and then transition `Ahead` can fire. This transition contains an algorithm of the following kind:

`S:1,n` Saves the first n notes

where n stands for the order number of the note after which object `B` is required to play outside this net immediately after the truncated MO `A` we have now in `Obj_Synchro`. In other words, in `Obj_Synchro` we have a partial execution of `A`, the part of `A` without `B` superimposed. This part superimposes itself, with no effect, on the entire object `A` played in `A_play`. So at the firing of the last transition we come to place `Out`, leave the subnet, return to the main net. So `B` can be played with the desired anticipation, while `A` is still playing.
Remember that the firing of a transition has a null duration in ScoreSynth, thus we have no loss of time in the execution of the algorithms and have effective parallel processes.

Harmonization in Stravinsky's works.

In describing Stravinsky's music we must be cautious in using the traditional harmonic language. We can think that the music of Stravinsky is a music organized around tone centers but not tonally structured in a strict sense. So when we use musical terms of the harmonic tradition we must forget the associations of these terms with their historical structural function in traditional music. At the current stage of development we have the lacking of a theory which could be considered comprehensive for Stravinsky's music. The problem is well expressed by Joseph Strauss (1982), who made a meaningful step in the direction of creating this lacking theory.
For example, the indications of the key-tonality can mislead the musicologist. We mean that Stravinsky has chosen a diatonic set of notes, but not that these have been tonally structured. This set can be used to construct chords overlapping several fourths, or incomplete chords not well classifiable in the tonal system. The first chord of the passage we have chosen and will discuss later may, for instance, be thought of as a minor ninth chord on the subdominant of Bb minor. But since we cannot find a tonic or a dominant chord and the progression at the end of the piece is quite non-tonal, we must conclude that if some material of the early style is used this is because of its complexive sound taken in itself and does not refer to its traditional structural meaning.

Inside this set there are surely some attraction centers, such as the Bb in our example, but not necessarily of tonal attraction. Stravinsky himself points out the problem (1947):
"Having reached this point beyond classical tonality, it is no less indispensable today to obey, not new idols, but the eternal necessity of affirming the axis of our music and to recognize the existence of certain poles of attraction. Diatonic tonality is only one means of orienting music towards these poles. The function of tonality is completely subordinated to the force of attraction of the pole of sonority. All music is nothing more than a succession of impulses that converge towards a definite point of repose".
So our choice is not to explain the sonorities we encounter. We simply take them as sound phenomena given in themselves without describing harmonic structures whose interpretation is not yet generally established and accepted.

A Comprehensive Example: "Spring Rounds".

Now we will see how the nets we have described in the above paragraph work on a musical example. First we make a musical analysis of the piece and then we build a Petri Nets model of the music structures we have identified in the analysis step. The piece we have chosen for our purpose is taken from "Spring Rounds"; it can be found from rehearsal 49 to 54 of the score.

Musical Analysis.

The piece consists of two parts. Each of the component parts is processed in the same way, with only some variances that we will point out at the suitable moment.
This common treatment is the dynamic juxtaposition of three fundamental sections that we have called A, B, C. The result of this juxtaposition gives in both parts the sequence: A B A B C.
All materials are derived from only one melodic fragment, which has already made an appearence in the "Augurs of Spring", that takes direct part in the passage. We have called this germinal cell Theme (see Figure 9).
Let's examine how the elements of the first part have been generated and structured. The A section is formed by pervasive low chords functioning like an ostinato underneath a melodic figure. We can easily notice that this figure is

simply the first bar of the Theme, transposed an octave lower and harmonized with only one repeated chord formed by a fifth higher and a major third lower (see Figure 10).
The ostinato is derived from the theme by the elimination of the Bb repetitions and the last two notes. Then the initial rest is exchanged with the first note of the Theme. This determines a contrast with the part above that has a beginning rest in correspondence with an effective playing note in this ostinato. Moreover, the rest is contracted till the half of its value. This determines with the superimposed figure a phase-displacement of a crotchet and reinforces the contrast with the superimposed part of the theme. This figure undergoes other transformations from a rhithmic viewpoint: the quavers become crotchets and viceversa.
At the end the entire obtained figure has been harmonized by simply doubling it a fifth lower in tonality. The exception regards the first note in which this fifth is doubled an octave higher, and all the notes of this first group are transposed an octave lower. In Figure 11 we have the result of all the transformations we have made, i.e. the ostinato.
We know that this explanation may seem artificious. It is not our aim to reconstruct the creative process as it occourred. We know that creation undergoes many unconscious processes. We have only made an attempt to create a formal model of structural and numerical-algorithmic relationships in a partition that undoubtedly are there, and that are the only mechanisms we can investigate from a mathematical-informatical viewpoint.
So we must think of the fact that the musical result of some transformations, whether conscious or unconscious, can be modeled by the composition of elementary algorithms that we may not perceive but that undoubtedly give the same result. Notice how the contrast between the obstinato and the superimposed theme is confirmed and revealed by numerical analysis. These are the two elements that contribute to the construction of the A section in both first and second parts of the entire passage.
The B section is in sharp contrast with the A section.
It is formed by a melody doubled in octaves and deduced from the Theme by an exact transposition a fifth higher, by the elimination of some notes and by the uniformation of all the rhythmic values on the crotchet (see Figure 12). This melody is harmonized by a chord obtained from the Bb, the pole of attraction of the piece, by a fourth higher and a fourth lower. Moreover two notes of the chord, Eb and Bb, are dwelled on conjuncted grade higher and lower

respectively. Another interpretation may be a chord of Bb minor with some *appoggiaturas* on its tonic and its third (see Figure 13). Both interpretations are possible, but ideally we may consider the chord like a sound phenomenon given for itself and made of the chosen diatonic set (notice how Stravinsky uses in this figure the first five of the seven notes of the diatonic set Bb Cb Db Eb F Gb Ab), and not always reconductable to the traditional harmonic theory.

At the second occurrence of B there are some variations. There is still the transposition and the chord below, but the entire group is now repeated twice. Moreover the eliminated notes in the melody are different in the two repetitions. At the end notice that the melody of this second B section superimpose itself on the next section with its last note, while the first was simply juxtaposed. Even this fact is a characteristic of Stravinsky's music: a continuous oscillation between perfect (what comes undermines what goes) and superimposed (what comes lives a little while with what goes, in large part only with the first note) juxtaposition.

Let's come to section C. In this section the Theme makes its first complete appearance. The ostinato goes on repeating and above it the Theme spreads by successive varied repetitions.

These variations are in the repetition of the theme head and in some truncations here and there. The Theme is harmonized by parallel chords in tonality, as shown in Figure 14.

In addition there is a minor theme (in the orchestral score played by piccolo and small clarinet in Eb), deriving from the main theme. For the sake of brevity, however, we will not discuss it here.

We have already said that the second part of the piece is constructed like the first, that is,by dynamic juxtaposition of the three fundamental figures we have pointed out. Even the result of the juxtaposition is the same: the sequence A B A B C.

The changes are to be found in the inner processing of the single section. In the A section the only differences between the single sections within the two main parts are in the number of repetitions involving the whole group. These numbers are: 3 2 (first part) 2 1 (second part).

Notice how Stravinsky acts both by asymmetry and by decreasing: what is repeated must be contracted so as to not render redundant its development to listeners.

In the two B sections of this second part the processing is inverted in respect with the first; the first section has a superimposition with what follows and the

second is simply juxtaposed. In the first part remember that the processing was the same but inverted.
Now we neglect the little differences between the A and B sections of this second part in respect with the first. This is because the C section is the truly interesting one.
As in the first part, here we have the superimposition of the theme on the ostinato. But what is new is that we also have the appearence of a counter-theme in opposition to the theme. Even in the ballet representation there were two big masses moving in a sort of big scenic counterpoint.
First of all we can notice that the Theme is processed in the same way we have seen previously, that is: it is repeated, with variations, a certain number of times and is harmonized by parallel chords in tonality. These parallel chords are the same as in the first part trasposed an octave higher and lacking a note.
The counter-theme can be produced by some transformations from the Theme. These are composed by some other simpler transformations:
1) widening of the distance between the original notes and the ones in the counter-theme as the theme goes on playing; this causes a complexively descending melody.
2) repetition of the last two notes and temporal contraction (the duration become half) of the new ones.
See in Figure 15 how these transformations work on the first four bars of rehearsal 58. The other bars are produced on the same principles. To complete the analysis we must say that the whole counter-theme is simply harmonized by real parallel chords doubled in octaves. We can see in Figure 16 the chord applied on a Bb note.

The PN Model

Our PN model has a sequential structure at the higher hierarchical level (see Figure 17). Both places `Part_1` and `Part_2` of the net are associated to the same macro we have described before and have called `Juxtaposition`. For the description of the net behaviour see the previous discussion about basic PNs. In this special case we have associated places `A_n`, `B_n`, `C_n` to the developments of the corresponding sections A, B, C discussed above.
The places labeled `A_n` both in the first and in the second call of `Juxtaposition` are associated to a two-part subnet `Polyphony`. These two

parts, which we have associated to two places named `Clock` and `UpBeat`, represent the upper part and the ostinato of the A section. To realize the verticalization of these objects they call a three-voices `Polyphony`. The parameter call will be:

Clock

`T:{P:1,$,[Theme,1],?-12`	Takes into the volatile memory the MO Theme
`S:1,4`	Takes only the first four notes
`M:1,$,3}`	Repeats three times the MO obtained
`Voice_1 :{P:1,$,?+7}`	Transposes a fifth higher
`Voice_2 :{P:1,$,?}`	Takes the MO as it is
`Voice_3 :{P:1,$,?-4}`	Transposes a major third lower

UpBeat

`T:{P:1,$,[Theme,1],?-24`	Loads the MO Theme, and transposes it two octaves lower
`K:2,3`	Eliminates Bb repetitions
`K:$-1,$`	and the last two notes
`D:1,$,24`	Lets all the durations be quavers
`I:1,2`	Exchanges the rest with the first real note
`P:1,1,?-12`	Transposes an octave lower the first note
`D:2,2,?/2`	The duration of the rest is halved
`D:$,$,?/2`	The duration of the last note is halved
`M:1,$,3}`	Repeats three times the obtained MO

`Voice_1 : {P:1,$,?-7}`	Transposes a fifth lower
`Voice_2 : {P:1,$,?}`	Takes the MO as it is
`Voice_3 : {P:1,1,?+5`	Transposes the first note of each bar a fifth higer
`P:6,6,?+5`	
`P:11,11,?+5}`	

We could have modeled the same passage with the net `Loop`. We have chosen not to follow this way in order to have in this case few hierarchical levels. We can see in the model the use both of the macro `Loop` and of more hierarchical levels.

We have already said that places `A_n` behave in the same way in both tparts of the piece. So, in the model they call the same subnet `Polyphony`. The only difference will be in the parameter list, and precisely in the algorithm of transition `T`, where the instruction

```
M:1,$,k
```

depending on the number of repetitions of the ostinato and the above melodic figure ; we have for k the numbers k=3 2 2 1.
Look how the calls preserve the information of the asymmetric-decreasing attitude of Stravinsky: it is enough to see the `M` instruction to point out the sequence 3 2 2 1 that we have already noticed as the characteristic of the A processing.

As we have already said section B is treated quite differently from A.
`B_1` calls a `Polyphony` subnet using the first two voices only; we have associated these two voices to two places named `Theme_2` and `Harmony`. `Theme_2` calls `Polyphony` to be doubled in octaves. The melody of `Theme_2` is derived from `Theme` in the net itself, in transition `Transf`. The parameters are:
Theme_2

`Transf: {S:4,$`	Considers the notes of the Theme from the fourth to the end
`P:1,$,?+7`	Transposes the Theme an exact fifth higher
`D:1,$,12`	Makes all durations be equal to a crotchet
`M:1,1,3`	Repeats the first note three times
`R:1,$,6`	Rotates all the MO obtained six places towards right
`P[C#]:$,$,?-1}`	Makes the last note descend a conjuncted grade The last three algorithms are for constructing the tail of Theme_2, formed by the first note and its immediate lower note in the tonality of Bb minor.
`T : {P:1,$,?}`	Takes the modified MO as it is
`Voice_1: {P:1,$,?}`	Leaves the MO as it is
`Voice_2: {P:1,$,?-12}`	Transposes the Mo an octave lower

To construct the harmony of this passage we must call a subnet `Polyphony` with three voices. The net has in input the MO `Theme`. The parameters are:

Harmony

`T: {S:2,2}`	Preserves the second note of the Theme (the Bb)
`Voice_1: {P:1,$,?-5`	Transposes the note a fourth below

```
          M:1,$,4            Repeats it four times
          D:$,$,?/2}         The duration of last repetition is halved
Voice_2: {M:1,1,2            Repeats the first note
          P[C#]:1,1,?+1      Elevates the first note of a degree
          M:1,$,7            Repeats the group obtained seven times
          K:$,$              Eliminates the last note

Voice_3: {P[C#]:1,$,?+4      Transposes the note a fourth higher
          M:1,1,2            Repeats the note two times
          P[C#]:1,1,?-1      Transposes the first note a degree lower
          M:1,$,7            Repeats the whole group seven times
          K:$,$              Eliminates the last note
```

In the processing of the B_2 object the differences lie in two facts. The first is that the melody is repeated twice with some changes in the second repetition. Second and more importantly, this second B section superimposes itself on the following section.

In regard to the first point, we speak only about the transformations that are necessary to realize the variation on the melody. In fact the harmonic part is exactly the same; what varies is only the number of times the basic structure is repeated. So, `B_2` receives MO `Theme_2` created in `B_1` (that is because they are following each other within the same process). And in the parameter call of the subnet `Polyphony` associated to place `Theme_2` we find:

```
Transf: {M:1,$,3      Repeats the theme three times
         K:$-4,$      Eliminates all the third part excepting the first note
         K:10,10}     Eliminates the third note of the second repetition
```

To realize the superimposition which we have mentioned before, this net has associated to place `Play_1` subnet `SynchroSplit`. So in addition of what we have already said, in the parameter list of `Theme_2` there is the instruction:

```
Play_1 : subnet SynchroSplit
```

The net `SynchroSplit` (see how it works in the discussion above) has in transition `Alg_1` the algorithm

```
K:$,$
```

that killing the last note allows the superimposition of the following section on the last note of current `Theme_2`.

In the second call of `Juxtaposition` this scheme is inverted: `B_1` uses `SynchroSplit` to superimpose itself on the following section, and `B_2` is simply juxtaposed against the C section. We want to point out that this fact is shown by the parameter calls of net `Polyphony`: the absence or presence of instruction `Play_1 : subnet SynchroSplit` reveals to us what processing will be done.

Section C is processed in a more complex way. We break the entire section C in two parts. This is to have an idea, by looking at subnet `Juxtaposition`, of the proportion put in the battlefield. Let's see how the model works in the case of the first `Juxtaposition` net. C_1 is composed of two elements: the pervasive low chords and the Theme above it. So it calls a subnet `Polyphony` with two effective parts, which we can call `Ostinato` and `ThemeMod`. Ostinato is processed like we have already seen. We want to show how to deal with the theme. We simply have three varied repetitions of the theme, and these variations concern with the theme's head. The first time the head is repeated three times, the second time only once, the third the head is truncated. To represent this process we use subnet `ThemeMod`. All the melody derived by means of `ThemeMod` is harmonized by parallel chords in tonality. So `ThemeMod` will have, associated to places `Mod_n`, macro `Polyphony` with four voices.
Summarizing the parameter list of `ThemeMod` is:

`Transf_1 : {M:1,4,3}`	Repeats three times the first four notes
`Transf_2 : {P:1,$,?}`	Plays the MO as it is
`Transf_3 : {K:1,2}`	Eliminates the first two notes
`Mod_1 : subnet Polyphony`	Associates to the places `Mod_n`
`Mod_2 : subnet Polyphony`	the subnet `Polyphony`
`Mod_3 : subnet Polyphony`	

We do not take into account the parameter calls of subnet `Polyphony` since we think how this net works is now clear.
The development of the C section in the second `Juxtaposition` is quite different from the first. We again have the repetition of the theme with variations on its head. In the first and second repetition within C_2 we also have some pitch variations. The theme is harmonized with parallel chords in tonality slightly different from those within section C belonging to the first part (the

same incomplete indeed). The really different and interesting thing in the passage is the second melody that counterpoints the first. To model this last part we use a `Loop-OrdPathSel` structure with three repetitions, both for `C_1` and for `C_2`. Each one of the places `Path_n` of `OrdPathSel` calls a `Polyphony` net with three voices. One is used for the ostinato, one for the Theme, and one for the Counter-Theme.
We have already seen how to process the ostinato and the Theme, so let's talk about the Counter-Theme.
We show how to construct the guiding melody and how it will be harmonized by parallel chords. So in places `Path_n` of the first macro `OrdPathSel` we have these parameter calls of the subnet `Polyphony`:

Path_1

`Transf: {M:1,$,2`	Repeats all the MO twice
`K:5,5`	Eliminates the rest of the second bar and by repetition
`M:5,5,2`	of the successive note transforms it in the Bb note
`P:5,$,?-1}`	Decreases the pitch of the second repetition of a half step

Path_2

`Transf: {K:1,1`	Eliminates the first note (the rest) and
`M:1,1,2`	substitutes it with a repetition of the second one
`P[C#]:2,$,?-(!-2)}`	From the second note on decreases the pitch of each notes a number of degree lower equal to the order position of the note diminished by two

Path_3

`T: {K:1,1`	Eliminates the first note (the rest) and
`M:1,1,2`	substitutes it with a repetition of the second one
`P[C#]:1,$,?-(!-1)`	Decreases the pitch of each note of number of degrees lower equal to the position of the note diminished by one
`P:4,4,?-1`	Decreases of an half step the pitch of the last note obtained
`M:$-1,$,2`	Repeats the last two notes and
`D:$-1,$,?/2}`	stretches their value of an half

All these subnets `Polyphony` have not only these algorithms but also all the algorithms associated to their `Voice_n` places. So each of these nets must be called with the following parameters in addition to the others we have only just mentioned. These calls are the same for all the nets because the Counter-Theme is always harmonized by effective parallel chords. We show the algorithms of only four of the eight notes that compose this chord. The remaining four are only a transposition, an octave higher, of these ones.

`Voice_1: {P:1,$,?}`	Play the MO as it is
`Voice_2: {P:1,$,?-5}`	Transposes an exact fourth below
`Voice_3: {P:1,$,?-8}`	Transposes a major sixth below
`Voice_4: {P:1,$,?-11}`	Transposes a major seventh below

Place `C_2` is treated along the same principles. We not consider all the complete transformations since they are quite similiar to the ones shown previously .

The relationships between piano and orchestral scores.

Throughout our study a fundamental role has been played by the four-hand piano reduction made by the composer himself. All our models have been made first of all on this score. The main reason for this approach lies in the comparison among the three documents we consider fundamental if we want to make an analysis of the Rite of Spring.
These documents are:
1) the orchestral score
2) the piano reduction made by the composer;
3) the Sketches.
This comparison has produced the following result: the piano reduction is similar enough to the Sketches to take this score as a library and/or effective radiograph of the rhythmic, harmonic and structural intuitions within the Rite. In the sketches we find something more, such as indications of folk tunes or instrumentation projects. Moreover, the piano score is already an orchestration of the bare naked elements in the sketches, for the piano itself is an instrument and requires an instrumental processing. Remembering that Stravinsky composed at the piano, we can conclude that this score is closer to the frame of the work than the orchestral one and rather than the sketches, represents a definite form This is why we have chosen this score to work on. Now we show

how it is possible to model the orchestral score as an enrichment of the piano model.

The orchestral score

As we have mentioned before many sources available in music literature (White, 1966) (Stravinsky, 1935-1936) (Craft and Stravinsky, 1959) seem to confirm that Stravinsky first composed a kind of "sketch" piano score with annotations on possible instrumentation and only in a second moment orchestrated it.

In our approach we can do something similar. We have said that our model is strongly based on the piano reduction. We want to show how it is possible to realize the orchestral score from the piano one, naturally using PNs.

To realize this purpose a relevant goal of our research has been the identification of relationships among MOs belonging to the piano duet (Stravinsky, 1947) or the orchestral score (Stravinsky, 1967) or both.

These relationship lead to two kinds of generative structures:

1) Essential Generative Structures (EGSs),

2) Additional Generative Structures (AGSs).

We call EGS a structure that determines the development of music material only in a temporal sense (Horizontal development). That is, in Stravinsky the opposition and/or juxtaposition of two or more periodical and variable melodic-rhythmic patterns; the synchronization and matching of periods in order to give asymmetric effects, and so on.

The concept of AGS concerns the orchestration of EGSs. We think that orchestration must be thought of in a more general sense as a means of creating new material, not only at a timbric but also harmonic level by superimposition of new generated material (Vertical development). Overall we can say that the same EGS occurs both in the piano reduction (Stravinsky, 1947) and in the orchestral score (Stravinsky, 1967), while AGS only in the latter.

An EGS is formalized by a PN in which the input MOs represent melodic or rhythmic ideas, and the PN determines the development of the generated material only in the horizontal sense. The previously seen model of "Spring Rounds" is based exclusively on EGSs. No mention is made of the orchestration of the material we have exposed. A model made only of EGSs is strictly based on the results of the already discussed comparison among the piano and the orchestral scores with the Sketches.

An AGS realizes PN morphism which, following a top-down procedure, gives an EGS+AGS net preserving the time structures stated by the EGS net.

Often Stravinsky's instrumentation is a simple assignation. This causes no problems with PNs. It is enough to expand a place regarding a playing MO with a splitting net such as the one shown in Figure 18. We have the parallelism to as many MIDI Program Change messages as output channels of the structures (i.e. the instruments).

Sometimes the instrumentation process is constructive: new material is generated from the one present in the piano reduction. And this one can disappear or superimporse itself on the new bearings.

Here we show only one example. Many others are available from the authors. People interested can contact them. We just want to show how it is possible to make an instrumentation of a score previously modeled by PNs.

The instrumentation method we want to consider here is the following: we have as the input MO a melodic line, from which we want to create two melodic voices that interleave one another and give the impression of being independent and processed like in a counterpoint style. An example of what we are saying can be seen in the introduction to the second part at rehearsal 86 of the score.

On page 62 of the Sketches we can find the melody that is shown in Figure 19. In both the orchestral and piano scores we encounter this MO transformed as shown in Figure 20. Referring to the orchestral score, we can see that the original MO is transposed a fifth higher and then that a trumpet in C plays the note in odd order positions of this fragment while another trumpet in C plays the notes in the even order positions. Moreover the notes are prolonged to cover the order positions left by the notes played by the other instrument. In this way we have the perception of two distinct parts, one going with the beat and the other with the upbeat, independent in their development but structured in a certain way. Moreover from a linear MO we come to a bidimensional MO.

Summarizing: the process consists of playing some notes of an MO with a certain instrument and the remaining with another instrument. The process can be extended to more parts and/or groups of instruments. To model this process we have created the net `Interleaving` (see Figure 21). As the input the macro receives the linear, melodic or rhythmic MO on which the process will be applied. The two MOs that will be obtained are associated respectively to places `Obj_1` and `Obj_2` and the net's segment which follows them. In transition `Alg` we can find a characteristic algorithm that indicates which are the notes to be eliminated in order to realize the interleaving among these and the remaining ones. In our example the characteristic algorithm is:

```
K:8,8
K:6,6
K:4,4
K:2,2
```

Its effect is simply to suppress all even notes.

When the net is invoked a token is put in place `In`, at the firing of transition a token reaches each one of places `Obj`, `Carry` and `SGN_1`. `Rot` cannot fire because `SGN_2` and `Aux`, two of its input places, do not have tokens. On the contrary `Empty` and `Alg` can fire. The firing of `Empty` cleares out the temporary memory from the MO associated to place `In` so to avoid superimpositions with the MO, a rest in our case, associated to place `Synchro_2`.

When transition `Alg` fires, the characteristic algorithm is executed and three tokens are distributed, each in the correspondent places `Obj_1`, `Obj_2`, and `Aux`. Transition `Alg_2` cannot fire because place `Obj_2` has only one token, and since the input arc of `Alg_2` has multiplicity two, the transition firing is inhibited until `Obj_2` has two tokens. `Alg_1` on the contrary can fire. It contains an algorithm that controls the durations of the notes which have not been suppressed, to get the superimposition we pointed out before. Each note is protracted the requested time:

```
D:1,1,?*(3/2)
D:2,3,?*2
D:4,4,?*3
```

When `Alg_1` fires a token reaches `Play_1` and another one reaches `SGN_2`.

Then in `Play_1` we have the execution of the first voice on MIDI Channel 1.

`Carry`, `Aux` and `SGN_2` have each one a token and so transition `Rot` can fire. This transition contains the algorithm:

`R:1,$,7` Rotates all the notes seven places towards right

Then we come back to `Alg` that repeats the characteristic algorithm. What happens now is the elimination of the notes that are complementary to the first ones eliminated as an effect of the rotation made in `Rot`. So in `Obj_2` comes an MO formed by all the notes of the initial MO that has not been used in `Obj_1`.

Notice that before `Obj_2` a rest of a quaver is executed to synchronize `Obj_2` with `Obj_1`. This rest is associated to place `Synchro_2`. Now `Alg_2` can fire because `Obj_2` contains two tokens and the following algorithm is executed:

`D:1,$,24` Makes all the notes last a quaver

See how the instruction suggests that this voice will be on the beat. This algorithm has the same function as the one associated to `Alg_1`, but it is applied on the second voice. In place `Play_2` the second voice is played on the MIDI Channel 2.
It is important to notice that the firings of the transitions have a null duration and so the executions of `Obj_1` and `Obj_2` happen at the same time, even if in the net they are consequently under a logical point of view.

Concluding Remarks.

Mapping the Rite score by PNs.

What we want to take into account in this paragraph is the possibility offered by PNs to map the Rite of Spring score. We have examined a certain number of score rehearsals, and from the models made out of these sections we have created the PN collection we have discussed in the first part of this paper.
It is not so important to say how many times a certain net occurs, but what we can say is that if we add to this set of PNs a certain number of sequential PNs the model covers all the pieces examined up till now.
The modelled rehearsal are:

From	*To*	*How Many*
1	4	3
13	15	2
18	22	4
46	57	9
67	78	11
91	93	2
104	121	17
142	201	59

So, we have modeled 50% approx of the whole work. An analysis done on the remaining part of the work reveals that most of the material can be modeled with the same set of nets.

Discussion of the PN model.

Notice that the algorithms whicht realize transformations in order to create new MOs are sometimes quite heavy. That is simply understood because the further we go from the beginning material, like during the creation of a new theme, the heavier must be the needed transformations. Besides, transformations that we consider simple from a musical viewpoint are actually the composition of many elementary operations. These elementary operations are described well by our music algebra.

Certainly this approach is not convenient from an information viewpoint, for it is easy to take as new anything we find in a piece!

Our goal is to bring into a model more information about how the piece is structured, at any hierarchical level we think necessary. By introducing directly a new theme we would have less information introduced but we would also lose many hidden structures of the passage. So a convenient viewpoint must be reached considering a description and modelling approach which:

1) is easy to understand;
2) is easy to handle;
3) reveals the hidden structures of music;
4) is an unitary means of description for musical analysis;
5) is close to musical thinking and practice.

Conditional Structures.

Furthermore, we want to discuss a relevant limitation in the use of our PN modeling approach. This limit concerns the fact that ScoreSynth has neither recognition nor conditional structures. So any kind of features which can be expressed by the form

```
IF ... THEN ...
```

cannot be represented according to this form.

Why is this so important? In our models we have often found many problems of that kind. For example, some of the following situations can arise:

1) Eliminate all the rests of a certain piece,
2) Truncate a certain section if it is longer than another one parallel and taken as leading,

3) Avoid certain musical processes. For example in the Rite we can notice the almost complete absence of dotted rhythms. Stravinsky avoided them because they are characteristic of romantic music, which he wanted to surpass.
We could add many other examples. The fact is that an instrument which can tell the composer and the analyst where certain structures are is necessary. We are not talking about a complex pattern recognitor, which could probably be a complete answer to our problem, but about a simple operator capable to finding elementary elements (such as a certain pitch, a simple rhythmic pattern, a chord) and then operating on these objects to get the composer's intention.
In all the examples we have given it is evident that the centre of the question is the need to find some elementary structures, make confrontations among some of them or some structures provided, and in consequence of the results of these confrontations do one thing or another. Even in the example we have discussed in this paper we have encountered some of these situations. In the B section of our model, in both the first and second parts, the harmony was truncated when the upper melody finished its natural development. To realize this intention we used the instruction `K:$,$`. This instruction is somewhat "manual": we must know where we want the cut. Using the conditional structure even if we do not know where the cut is the job is well done. A structure needed in this case will have the form:

```
    IF MO1 play no THEN MO2 mute.
```

Notice moreover (and surely this is the really true centre of the problem) the instruction `K:$,$`, does not express that the MO is interrupted because something elsewhere has finished playing. The model loses the essential information linked to this operation, precisely what we want to avoid.
Another fact discussed in the preceding model was about the construction of the ostinato. We have used operations like:

```
    IF a note has the duration of a quaver THEN transform it
in
    a crotchet
    IF a note is a crotchet THEN transform it in a quaver
    IF you find a pause THEN exchange it with the next
sounding
    note
```

All this information about how we have constructed the piece is completely absent in our model. We have been forced to use "manual"operations such as, in the case of exchanging the pause with the first sounding note:

```
I:1,2
```

The fact that two notes are exchanged is revealed. But the most important fact that the first of these notes is a pause (the reason why we made the exchange) is completely hidden.

A possibility for further research lies in investigating other kinds and extensions of PN models. The use of Predicate-Transition Nets has already been considered in (Pope, 1986), for istance. It could be a primary goal of our research in the near future to upgrade the PN modeling approach by means of these descriptional features.

Acknowledgements

Many thanks to A. Sametti for discussions and suggestions about the realization of Petri Net models by means of the ScoreSynth program. Moreover, special thanks are due to the L.I.M.-Laboratorio di Informatica Musicale staff for the scientific and technical support provided all throughout our research.

This research has been partially supported by the Italian National Research Council in the frame of the MUSIC Topic (LRC C4): "INTELLIGENT MUSIC WORKSTATION", Goal C: SISTEMI AVANZATI DI PRODUTTIVITA' INDIVIDUALE, Subproject 7: SISTEMI DI SUPPORTO AL LAVORO INTELLETTUALE, Finalized Project SISTEMI INFORMATICI E CALCOLO PARALLELO.

REFERENCES

Boulez, P. (1953). Stravinsly demeure. *Musique Russe*, Vol. I, Paris: Presses Universitaires de France.

Craft, R.; Stravinsky, I. (1959). *Conversations with Igor Stravinsky*. New York: Doubleday Publ.

Haus, G.; Rodriguez, A. (1993). Formal Music Representation; a Case Study: the Model of Ravel's Bolero by Petri Nets. In G. Haus (ed.), *Music Processing* (pp.165-232). Madison, WI: A-R Editions.

Haus, G.; Sametti, A. (1992). SCORESYNTH: a System for the Synthesis of Music Scores based on Petri Nets and a Music Algebra. In D. Baggi (ed.), *Readings in Computer Generated Music* (pp.53-78). Washington: IEEE Computer Society Press.

Nabokov, N. (1951). *Old Friends and New Music*. London: Hamish Hamilton.

Pope, S. T. (1986). *Music Notation and the Representation of Musical Structure and Knowledge*. Perspectives of New Music, 24(2),156-189.

Strauss, J. (1982). *Stravinsly's Tonal Axis*. Journal of Music Theory, 26(2), 261-290.

Stravinsky, I. (1947). *The Rite of Spring - Reduction for Piano Duet*. London: Boosey & Hawkes.

Stravinsky, I. (1967). *The Rite of Spring*. London: Boosey & Hawkes.

Stravinsky, I. (1969). *The Rite of Spring. Sketches 1911-1913. Fac-simile reproductions from the Autographs*. London: Boosey & Hawkes.

Stravinsky, I. (1935-1936). *Chroniques de ma vie*. Paris: Les Editions Denoël et Steel.

Stravinsky, I. (1947). *Poetics of Music*. (Knodel and Dahl trans.) Cambridge: Harvard University Press.