

Music Analysis and Modeling Through Petri Nets

Adriano Baratè, Goffredo Haus, and Luca A. Ludovico

LIM-DICO University of Milan,
Via Comelico, 39,
20135 Milano, Italy
{barate, haus, ludovico}@dico.unimi.it

Abstract. Petri Nets are a formal tool for studying systems that are concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. They were used in a number of real-world simulations and scientific problems, but seldom considered an effective means to describe and/or generate music. The purpose of this paper is demonstrating that Petri Nets (enriched with some peculiar extensions) can well represent the results of a musicological analysis process.

1 Introduction

This paper represents the results recently obtained at LIM (Musical Informatics Laboratory, State University of Milan) in the area of Music Petri Nets (Music PNs). This mathematical formalism, shortly introduced and defined in the following sections, can be applied to music field according to different meanings. Roughly, we can recognize two possible categories of applications: analysis and composition. However, the former and the latter aspects cannot be considered completely independent. In fact, PN-oriented analysis would provide poor results, if not aimed at the comprehension of the original composition or even at the generation of a new music piece that shares some common features with the one previously analyzed. Moreover, PN-based composition itself would produce insignificant results, if not supported by a deep comprehension of the underlying structures, which involves – after all – an analysis process. In this paper, we will concentrate on the analytical possibilities and limitations related to the application of Petri Nets to music field. The possible consequences in music composition will not be explored, and will represent one of the subjects of our future work.

When applied to music analysis activities, the adoption of Petri Nets should not influence the approach of the researcher. In fact, Petri Nets should be thought as a way (one of many possible ways) to express the results of the analytical process. From our perspective, this formalism cannot limit or influence the analysis, which is an activity that obviously precedes the representation of its results.

We said that Petri Nets should not constitute a limitation, but this is not sufficient: we want to demonstrate that they are useful and effective. We will show that Petri Nets are a promising tool to represent and read music analysis results. Petri Nets were born to describe concurrent, asynchronous, and parallel processes, and these characteristics can be found in music as well. Other characteristics, such as non-determinism,

can be useful for music composition through Petri Nets, but are not very significant for analysis.

In the following discussion, no constraints will be imposed about music to analyze: music works can belong to different genres, and come from different cultures, geographical areas and historical periods. On the contrary, we will underline the adequacy of Petri Nets formalism according to different degrees of abstraction in analysis, which will be the subject of next section.

2 Music Analysis and Grouping Structures

In this section we introduce the formal concept of *grouping structure*, as defined in [6]. A *group* can be constituted by any contiguous sequence of pitch events, undetermined beats or rests. Only contiguous sequences can constitute a group. A group can contain smaller groups, and in this case the subgroups must be completely contained in the former. Finally, if a group contain at least a smaller group, it should be possible to partition it exhaustively in smaller groups. These conditions define a strict, non-overlapping, recursive hierarchy, and constitute a set of grouping well-formedness rules. Intentionally, we don't introduce at the moment a set of grouping preference rules.

In this context, a music piece as well as a single note can constitute a group. Besides, the identification of grouping structures allows extracting from the score music objects such as episodes, themes, rhythmic patterns, or harmonic cadences.

Fig. 1 illustrates in a hierarchical fashion some possible groupings for a melody.¹ To reflect hierarchies, groups are represented by slurs placed beneath the music notation.



Fig. 1. Examples of grouping structures

Considering only 8 measures imposes serious limitations to the reachable degree of analysis: it doesn't allow the segmentation in episodes of the whole first movement, or the identification of recurrences of the music object represented in Fig. 1. Nevertheless, at least three categories of grouping structures can be identified. The most comprehensive structure (i.e. the largest group) embraces a whole period, whereas the two subsumed groupings reflect the subdivision of the 8-measures period in two 4-measures phrases. More interesting, the third proposed grouping structure tries to highlight relationships among smaller music objects: for

¹ W. A. Mozart, *Sonata in F major KV 332 (300k) – Allegro (I movement)*, bars 1-8 [8]. All the music examples in this paper have been extracted from this piece.

instance, the first 3 groups (measure 1, 2, and 3 respectively) present rhythmic similarity; group 5, 6, and 7 (in measures 5, 6, 7, and 8) have a similar melodic behaviour. Of course, other segmentations could be realized, by using different criteria for grouping structure identification.

It's worth to note that our concept of grouping structure can embrace also vertical slices of a music piece, which allows harmonic analysis and the segmentation of a piece in episodes.

In the simple example shown in Fig. 1, we considered only a small portion of a voice. Of course, the identification of grouping structures should be extended to the whole piece or even to a set of pieces that constitute an overall music work (e.g. the movements of a sonata or the episodes of a symphonic poem). According to our previous definitions, the analytical process for a music work can be thought – in general terms – as the identification of grouping structures together with their relationships inside the piece. Intentionally, this statement is quite vague. In fact, our purpose is specializing such definition in a number of different ways (creating different degrees of abstraction in music analysis) in order to demonstrate the adequacy of Petri Nets to the different cases.

From the musicologist's perspective, identifying grouping structures is the first step to establish relationships among them, highlighting similarities and differences. For instance, the music form known as *sonata* is characterized by the presence of two contrasting themes (the principal and the secondary theme), that are reposed in a literal or slightly varied form during the piece, according to given rules and to the inspiration of the composer. After identifying those grouping structures, it is possible to show the alternations and recurrences of themes and other transition music objects: in other words, the analysis focuses also the relationships among music objects.

The analytical process can be conducted at different degrees of abstraction, addressing movements in a complex composition, macro-episodes in a piece, themes in an episode or even atomic music events in an elementary music object. In this work we will show how music structures can be highlighted and also processed by means of a more abstract kind of representation than the staff one. Petri Nets, a formalism that will be soon described, are able to represent effectively the results of music analysis depending on the degree of abstraction we want to introduce in the process.

3 An Introduction to Petri Nets

A Petri Net (PN) is an abstract and formal model to represent the dynamic behaviour of a system with asynchronous and concurrent activities ([4], [9], [10]).

PNs consist in a set of basic objects: places, transitions and arcs, whose graphical representations are circles, rectangles, and oriented lines respectively. Places and transitions are also called nodes.

In Fig. 2, an example of an elementary PN is shown. P1, P2, P3, P4 are places, T1, T2, T3 are transitions, and the oriented lines represent arcs. The number associated to arcs is called *arc weight*.

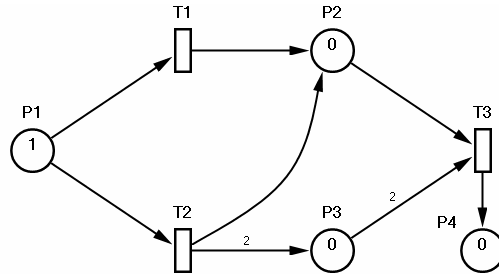


Fig. 2. An example of Petri Net

When building a PN, some constraints about elements layout must be respected. An arc can connect only nodes of different kind, i.e. a place to a transition or a transition to a place. However, two or more arcs having the same orientation can connect two nodes. From a graphical point of view, such a behavior can be summarized using the concept of arc weight, which represents the multiplicity of the arc.

A key concept of PNs is the idea of *marking*, realized using tokens. Any place can hold a certain number of tokens, usually represented by little black circles; however, in a computer-oriented description, a better formalism is adopted: a numerical value inside the place indicates the total number of tokens in that place at a given time. This is the meaning of the value inside the places in Fig. 2.

Tokens let PNs evolve and self-modify. They can be transferred from place to place according to policies known as firing rules. The dynamic evolution of a PN is determined by the following rules:

- When all the incoming places of a transition present a number of tokens greater or equal to the weights of the corresponding incoming arcs, the transition is enabled.
- When a transition is enabled, the fire drops from the incoming places a number of tokens equal to the weights of the incoming arcs and adds to each outgoing place a number of tokens equal to the weights of the corresponding outgoing arc.

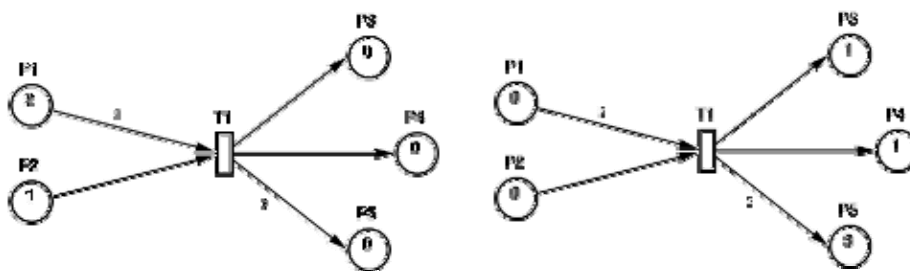


Fig. 3. Evolution of a Petri Net

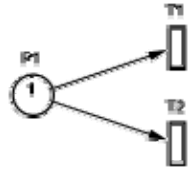


Fig. 4. Alternative transitions

When a transition is enabled, its fire is not automatic. For instance, looking at Fig. 4, T1 and T2 are both enabled, but there is only one token in input, so they both cannot fire. In such a case, we call them *alternative transitions*, and only one of the two transitions will fire. This is a kind of non-determinism.

A net execution is formed by subsequent transition fires, and terminates when no more transitions are enabled.

3.1 Extensions

In order to use PNs as an effective tool to describe music, we have to define further extensions to the given definition. Not all the extensions presented here are original: many of them are already in use in PNs general applications.

Capacity. The capacity is a property of places, and indicates the maximum number of housed tokens. This attribute creates a new condition for transition enabling: a transition cannot be enabled if the marking of at least one output place would become greater than its capacity after the fire of the transition.

The introduction of the capacity concept adds a new type of non-determinism (see also Fig. 4), called *conflict*. A conflict occurs when two (or more) transitions are enabled, but the fire of one transition prevents the fire of the other one(s), according to their capacities (see Fig. 5).

In our graphical representations, the upper value inside a place represents its present marking (number of tokens), whereas the lower one indicates its capacity.

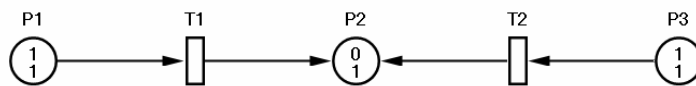


Fig. 5. An example of conflict

Refinement. The theory of morphisms is complex, and this is not the place for an exhaustive treatment. However, we introduce the concept of *refinement*, a simple morphism used to describe complex PNs in terms of simpler ones, in a hierarchical way. A refinement, called *subnet* in this context, represents an entire PN that replaces a node. For instance, in Fig. 6, the place P2 subtends the PN on the right; the expansion of such node would generate the lower global PN. A subnet must have an input node and an output node of the same type of the refined node.

Temporization. As stated above, PNs are a good way to describe concurrent processes together with their synchronization. From a temporal perspective, when a transition is enabled, the duration of the fire is supposed to be null, so a PN execution can

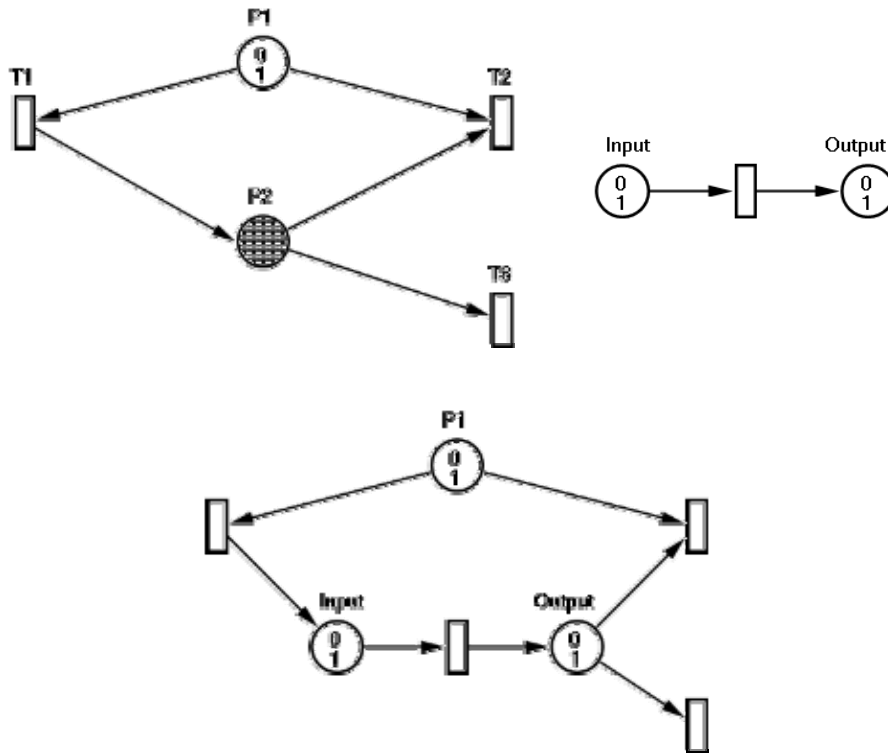


Fig. 6. Refinements and subnets

be considered instantaneous. In Music PNs, we will necessarily introduce the concept of temporization, as music processes are temporized and music objects present a temporal dimension.

Probabilistic Weight. In order to face conflicting or alternative situations, we have to introduce another extension: the *probabilistic weight* of arcs. When many transitions are enabled, the choice depends on the probabilistic weight of the arcs involved, in relation with the total sum of their weights.

For example, let us consider a PN with 3 arcs: A_1 (probabilistic weight $W_1 = 5$), A_2 ($W_2 = 10$), and A_3 ($W_3 = 300$). If at a given time t_1 the choice is between all the three arcs, A_1 shall have a probability of $5/315$ (1.6%) to fire, A_2 a probability of $10/315$ (3.2%), and A_3 a probability of $300/315$ (95.2%). At the time $t_2 > t_1$, let only A_1 and A_2 be enabled: their new probabilities will be $5/15$ (33.3%) and $10/15$ (66.7%) respectively.

A particular situation occurs when an arc has a probabilistic weight equal to 0. In this case, the associated transition will fire only if there are no other alternative or conflicting arcs with greater probabilistic weight.

Probabilistic weight will be graphically represented by a numeric value over the arc, in square brackets.

3.2 Music PNs

In Music PNs, we can associate music objects to places. According to the definition in [2], a *music object* may be anything that could have a music meaning and that could be thought as an entity, either simple or complex, either abstract or detailed. Such entity will present some relationship with other music objects. When a place containing an object receives a token, the music object is executed. Fig. 7 shows two simple music objects that will be helpful to understand the following examples about transitions' behavior.

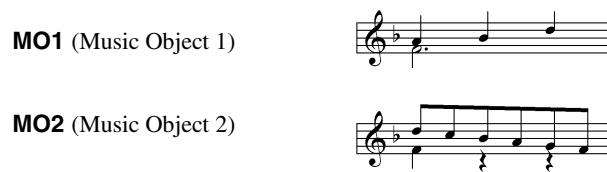


Fig. 7. Two music objects: MO1 and MO2

Transitions play an important role: they determine – together with tokens – the evolution of the net. In our extension of PNs, namely Music Petri Nets, two categories can be found: transitions *with* and *without* associated music operators. We can consider the latter category as transitions having a null operator associated.

When there is no music operator associated, transitions present a simpler behavior: they are only devoted to net evolution. Their role is dropping tokens from input places and adding them accordingly to output places. As noted before, when a token arrives at a place, if the place has an associated music object, this object is played. In Music PNs, the temporization is performed accordingly to the durations of the music objects (eventually) associated to the places. When a place receives a token from an incoming transition, the (eventually) associated music fragment is executed, and the token cannot be used until such execution is completed.

An example is provided in Fig. 8, where MO1 is associated to the left place and MO2 is associated to the right one. The first measure comes from the execution of MO1, which took place when the token arrived at the left place. Then, the token is moved to the right place, originating the subsequent execution of the second measure. The overall result is noted in the score.



Fig. 8. The *sequence* structure

Even by using transitions *without* music operators, it is possible to create peculiar structures. Fig. 9 provides five simple nets in order to illustrate respectively a fusion (two objects generating one object), a split (an object generating two objects), an alternative (a non-deterministic choice between two objects), and a joint structure (a logical connection between two objects).

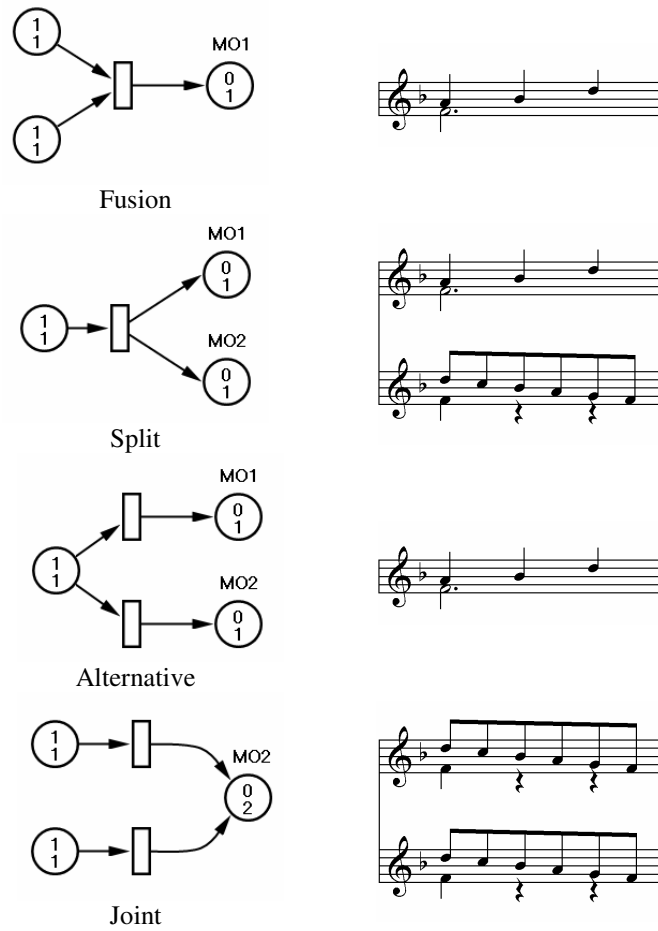


Fig. 9. Some PN structures and the corresponding executions

In the aforementioned approach, we said that transitions might have also an associated music operator. The purpose of music operators is providing changes to input objects (i.e. objects coming from input places), and passing the transformed objects to output places. Typical operators associated to transitions reflect common music operators, such as inversion, retrogradation, and transposition. The behavior of the last operator is shown in Fig. 10.

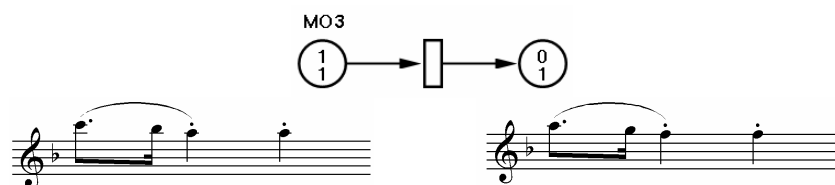


Fig. 10. Music Object 3 before (left) and after (right) the transposition

4 PN-Oriented Analysis and Grouping Structures

The main questions this paper wants to answer are the following: Are PNs a good formalism to collect the results coming from an analytical process, at various levels of detail and abstraction? And are PNs a valid tool to provide a new way to read music structures and to create relationships among music objects? These questions were partially answered by previous works and papers, such as [2].

Apparently, the applications of Petri Nets to music analysis lead to contradictory results. If [3] demonstrates a great efficacy in describing the structure of Ravel's *Bolero* through Petri Nets, on the other hand [1] points out some limitations of this approach, on the base of opportune counterexamples.

An accurate choice of the piece to be analyzed could generate excellent results: this is the case of music works constituted by a few music objects or having a very simple structure. For instance, the music form known as *canon*, based on the (almost) slavish repetition of the same music objects in different voices at different instants, can be represented in a very compact way by using Petri Nets. Provided that music objects are suitably identified, most counterpoint production (including baroque and serial music) originates very compact PN-based descriptions. Another field of application is music with a very simple structure, such as pop songs or deliberately trivial pieces. Ravel's *Bolero* belongs to the latter category: its structure is intentionally simple and repetitive. Probably it would be very difficult to describe a romantic piece or a jazz improvisation by Petri Nets.

In our opinion, PNs can be more or less appropriate to provide a readable and compact description of analysis results depending on the level of detail the analysis wants to reach. This statement justifies the contradictory results obtained within the same research group at LIM when considering different music pieces.

Thus, Petri Nets are more or less efficient and effective depending on the possibility to identify a few music objects and simple relationships among them. According to the aforementioned definition of music object, this concept can embrace whole episodes of a music work as well as single atomic events. As noted before, the identification of music objects is strictly related to the grouping structures we chose for analysis.

Next section will take into consideration the first movement of a sonata by W.A. Mozart. Intentionally, this case study takes an intermediate place between a strongly structured piece and a completely unstructured one: the score can not be described as a sequence of few repetitive music objects (like a canon or fugue), however musicologists agree about the presence of characteristic grouping structures (principal

theme, secondary theme, middle themes) and about its macroscopic segmentation (exposition, development, repeat, and coda). This case study will constitute a serious benchmark for Petri Nets application in music analysis.

5 A Case Study: 1st Movement of a Sonata by W.A. Mozart

On the base of E. Surian's text [11], we have analyzed the 1st movement of piano *Sonata KV 332* by W.A. Mozart. The purpose of our analysis is the application of PNs to the structure of the music piece, at different degrees of abstraction.

A typical sonata-form movement consists of a two-part tonal structure, articulated in three main sections [7]. In this piece, the optional introduction and coda are not present.

The first section, called *exposition*, divides into a "first group" in the tonic and, after transitional material, a "second group" in another key. The piece we considered is in major key, so the second group - according sonata form's rules - is presented in the dominant degree. Both groups may include (and in this case actually include) a number of different themes. The first group of themes includes the principal one, namely the *main theme*; the second group of themes introduces the *secondary theme*, together with other thematic material.

The second part of sonata structure is represented by the *development*, which elaborates material from the exposition in a variety of ways, moving through a number of keys. It also prepares the return to the main theme and to the tonic key which begins the following part.

Finally, the third section is named *recapitulation*, and restates the themes of the exposition, usually in the same order. The second group is now heard in the tonic.

Interesting differences occur in the transitional measures, namely the part between the first and the second group of themes: in fact, in the exposition the two groups are in the tonic and in the dominant degree respectively, whereas in the recapitulation they both are in the tonic degree.

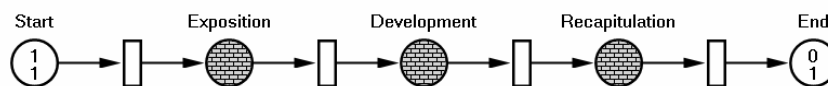


Fig. 11. A high-level representation of sonata form

The aforementioned structure is shown through PNs in Fig. 11. The three music objects that come into play are: **Exposition** (meas. 1-93), **Development** (meas. 94-132), and **Recapitulation** (meas. 133-229). This first example originates a very simple Petri Net, which is a trivial sequence of three steps. The description is very compact and clear from a graphical point of view, but it conveys poor musicological information. Besides, intrinsic characteristics of Music PNs (such as parallelism) are not exploited. The example in Fig. 11 will be taken up later, after showing how recapitulation can be derived from exposition, and will originate a more interesting net.

Key:

— First Group (FG)	— Transition (T)	
..... main theme, first theme in FG (1FG)	1T — · · —	2T
- · - · - second theme in FG (2FG)	3T - - - -	4T
	5T — — — —	6T
	7T - - - -	

Key:

- | | |
|--|--------------------------|
| Second Group (SG) | Close Group (CG) |
| secondary theme, first theme in SG (1SG) | first theme in CG (1CG) |
| second theme in SG (2SG) | second theme in CG (2CG) |
| third theme in SG (3SG) | |

At a lower degree of abstraction, we now provide a closer look at the inner structure of the exposition. In Fig. 12, the exposition follows the upper branch of the net, whereas the lower one represents thematic transformations applied in the recapitulation. As noted before, in sonata exposition there is a first group of themes (**1FG** and **2FG** are the first and the second theme of the group, respectively), then a transitional segment (**T(1)** and **T(2)**), and finally a second group of themes (**SG**), followed by a close group (**CG**). The lower branch shows the evolution of the same thematic material in the context of recapitulation. In this case, our PN illustrates in a clear and compact way the similarities between sections of a sonata. We notice that:

- **1FG** and **2FG** recur in both sections, without variations.
- The transitional material is slightly different, and those dissimilarities will be further investigated.
- Finally, **SG** and **CG** are identical in the exposition and in the recapitulation, but, in the second case, they undergo a transposition. In order to apply this operator to the music object only in the recapitulation, we first have to load it (**Load SG/CG** place) and then to execute it, with or without transposition (**Exec SG/CG** place).

The grouping structures we are using refer to smaller blocks of measures, coming from the analytical process of segmentation proposed in [11]. This process has identified the main theme, the secondary theme and other themes from first, second, and closing groups.

In order to make the reading of Music PNs easier, in Figure 12 and following places with an associated music object are gray colored, whereas places without an associated music object (as either inherit a transformed music object from the preceding places or simply make PN evolve without musical consequences) present a white background color.

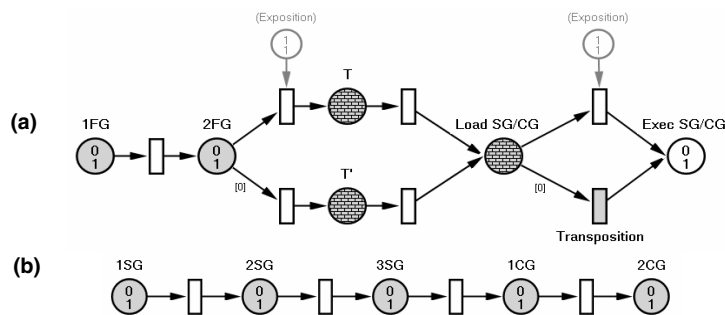


Fig. 12. (a) Themes evolution in Exposition and Recapitulation; (b) SG/CG subnet

Thanks to the aforementioned considerations, now we can revise Fig. 11 in order to synthesize **Recapitulation**'s dependency from **Exposition**. The result is shown in Fig. 13, where high-level structural representation is even more compact.

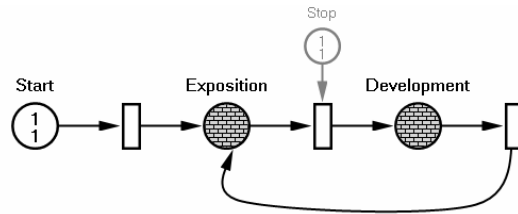


Fig. 13. A revised representation of sonata form

Finally, we take into consideration what happens in the transition between the first and the second group of themes. In the exposition and in the recapitulation a different behavior will occur, as tonal relationships between the first and the second group of themes change (passing from tonic-dominant to tonic-tonic).

The complete example, shown in Fig. 14, illustrates the expressive power of PN formalism. The choice of music objects to consider reflects only one of many alternatives: our goal here was keeping graphic complexity low, showing – nevertheless – a number of relationships among original music materials. Of course, other grouping structures are conceivable, and they would embrace other properties and relationships of music material. For instance, we identified **1T** and **2T** as different objects, because they are treated in a slightly different way in exposition and in recapitulation sections, and not for musicological reasons. Likewise, within **1T** the rhythmic and melodic figures played by the left hand at measure 23 and 24 are very similar, but this relationship was intentionally ignored in order to design a more compact Music PN. At a lower degree of abstraction, it is worth noting that many other relationships can be caught; unfortunately, this would imply a more complex (and less readable) Petri Net.

Fig. 14(a) illustrates music evolution in the first occurrence of transition, i.e. the transition segment within exposition. After 6 measures (**1T** and **2T**), we had to split music score in two parts, denoted by suffix **rh** (standing for right hand) and **lh** (for left hand) respectively. Concerning **3T**, it is considered a single music object within exposition, while it is managed as two separated parts during recapitulation section.

The splitting mechanism previously introduced is very effective when applied to **4T**, **5T** and **6T**: in fact, it lets us describe measures 31-36 as the transformation of the same elementary music material. We underline that – over the two branches – a music object is associated only to **4Trh** and to **4Tlh** (grey places), whereas the subsequent white-colored places inherit music objects from the preceding ones. Of course, transitions operate some changes: concerning right hand, music material undergoes a number of pitch substitutions (e.g., passing from **3Trh** to **4Trh**, G becomes Ab); for the left part, the minimal grouping structure is made of only one measure, always literally repeated twice. In our models, the transitions that have an associated algorithm are grey-colored. In this case, the musical operators associated to transitions are: diatonic transposition **Diat(Ab-3)**, two degrees (i.e. a third) down along Ab major scale, and complement **Compl(-2)** to obtain the second inverted chord. After **6T**, the evolution of Music PNs brings to converge at **7T**.

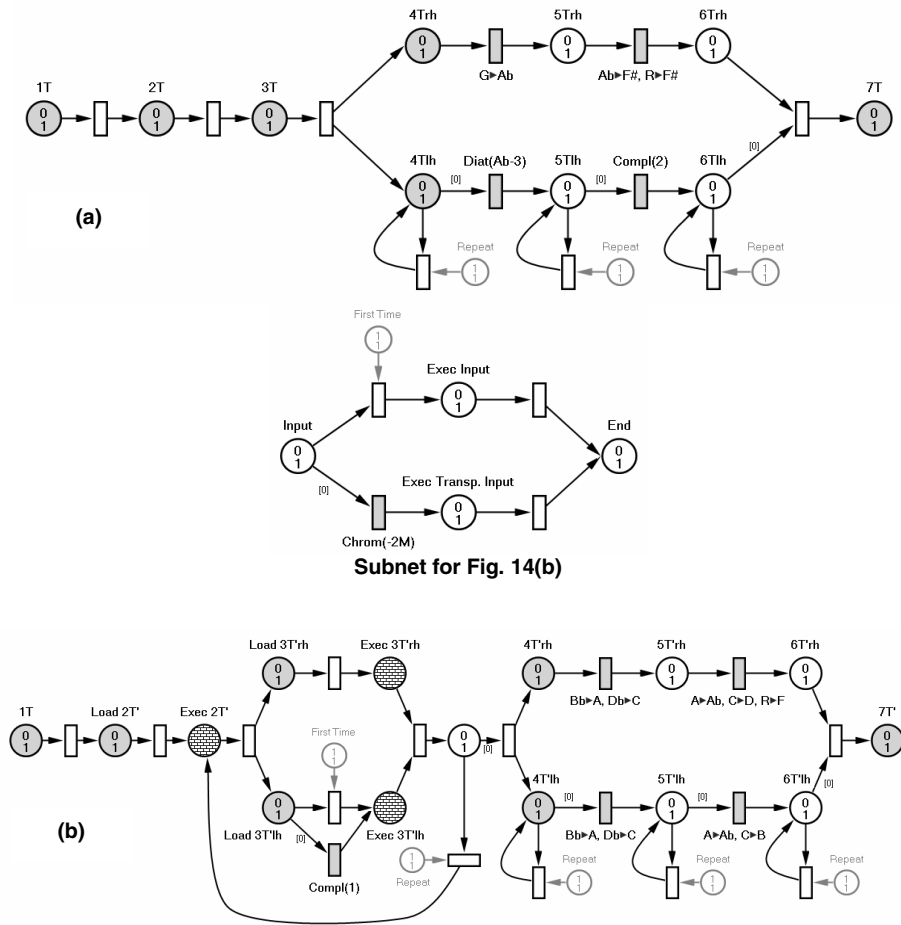


Fig. 14. (a) Transition from Exposition; (b) Transition from Recapitulation

In Figure 14 we provide two distinct models (a and b respectively) to illustrate how transition evolves in exposition and in recapitulation. However, the two parts of Figure 14 are not independent: the original material is the same in both cases, as indicated by the place names. **1T** remains unchanged, whereas many other music objects undergo a transposition from the dominant degree to the tonic degree. The harmonic grid cannot be the same in both cases, and this consideration emerges clearly looking at the different music operators associated to PN transitions in Figure 14(a) and (b). Together with **Compl** and **Diat** operators, already introduced for Figure 14(a), in the subnet related to Figure 14(b) we find a **Chrom(-2M)** operator, that transposes all the corresponding intervals a major second below.

A single PN describing the two occurrences could be created, as we did in Figure 12, but the result would be less readable.

In order to appreciate the operation of PN transitions, the following example (Figure 15) relates a significant part of Figure 14(b) to the corresponding measures in the recapitulation.

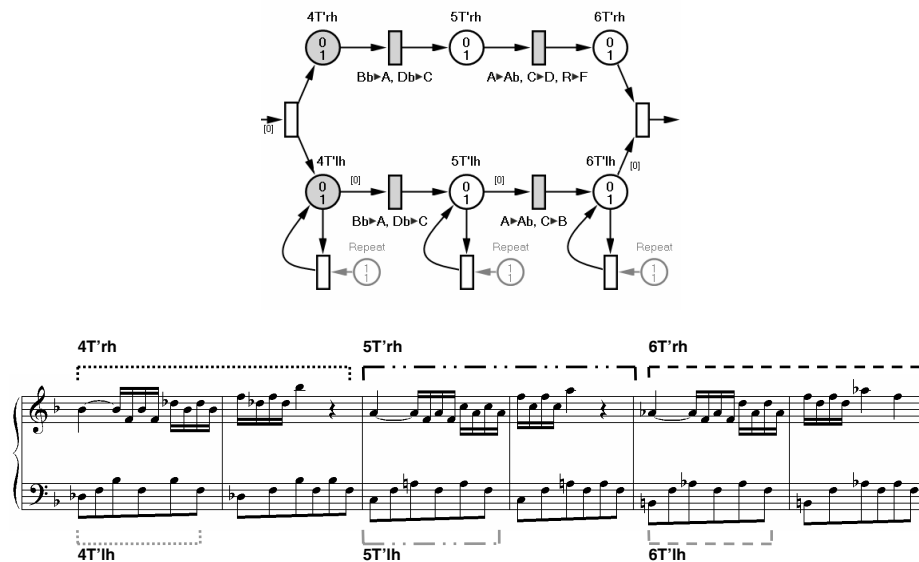


Fig. 15. Operations performed by PN transitions

6 Formats to Collect PN-Oriented Analysis Results

The Petri Net Markup Language (PNML) is a proposal of an XML-based interchange format for Petri Nets. The standardization effort originated the working draft of ISO/IEC 15909-2 for a Transfer Format for High-level Petri Nets. At present, the PNML format is supported by several PN tools, which facilitates the exchange of PN models. The goal of this format is the possibility to add new extended PN types, maintaining a set of basic features. This allows, for example, the export of a PN model to another tool that, adopting the PNML format, simply ignores the unsupported features without losing all information.

The PN design tool we have used, namely ScoreSynth (see next section), adopts a PNML-based approach to save and load PN models. In ScoreSynth we have developed an extension to PNML basic PN types to incorporate all the extensions we have to use for Music PNs.

In order to collect the results coming from a PN-oriented analysis, our purpose is to encode both music symbolic information and the corresponding structural information in a unique data structure. XML provides an effective way to represent musical information at different levels of abstraction. Thanks to the file format we are developing at LIM², namely MX, it is possible to represent notational symbols as well as

² Laboratorio di Informatica Musicale, Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano.

music objects and related structures. MX is an XML-based format that describes music information according to a multi-layer structure, where each layer is specific to a different degree of abstraction in music information. As described in [5], MX is constituted by General, Structural, Music Logic, Notational, Performance and Audio layers.

Our encoding format is particularly suitable to describe information coming from a manual or automatic segmentation process. Thanks to its multi-layer layout, themes and other musical objects (Structural layer) can easily refer to organized symbols in score (Music Logic layer). Besides, MX format encodes also the relationships intervening among music objects.

Another important characteristic of MX encoding is the possibility to import and/or support fragments from other XML-based formats, such as PNML.

7 Related and Future Works

The application used to design, execute and debug all the PNs involved in this paper is called ScoreSynth, and was developed at LIM. Since 1980, we have been using PNs as the basic tool for music description and processing, creating some applications to support our approach. In our recent work, we have switched to Windows operating system (Macintosh was the preceding platform), and we have developed two new applications: ScoreSynth and MediaSynth. Both are based on the same interface, which allows drawing, editing, and executing PNs. In ScoreSynth, the material associated to places is symbolic and represented in MX format, while MediaSynth is dedicated to multimedia processing, with media associated to places and effects associated to transitions. The PN-based analysis presented in this paper was performed and validated by ScoreSynth. Thus, in our implementation of a tool to design and executing Music PNs, the format chosen to represent music objects is MX, an XML-based encoding that can incorporate all levels of music representation.

Concerning future works, an interesting idea is exploiting analysis results to write new music compositions. For instance, it is possible to maintain the same structural characteristics of the analyzed piece, but changing music objects associated to places and/or music operators associated to transitions. These possibilities will be further explored in the next future.

8 Conclusions

In our opinion, after some opportune extensions, Petri Nets can effectively represent the results of a musicological analysis process. This representation can be more or less effective, readable, and compact depending on the music objects we choose and on the degree of abstraction we want to reach. Music PNs work very well when applied to general structures (e.g. sonata form), provide interesting and peculiar results when applied to thematic segmentation (e.g. themes in a section), and finally present a too complex layout when atomic music objects are involved.

Their full potentialities are exploited when a number of relationships occur in a music piece. PNs are very effective and efficient to describe concurrent, distributed,

parallel processes, and an *ad hoc* choice of music objects can highlight such peculiarities. The application of this formal tool to music analysis can show surprising results.

Acknowledgements

The authors wish to acknowledge the partial support of this project by Italian MIUR (FIRB “Web-Minds” project N. RBNE01WEJT_005) and the Italian National Research Council, in the framework of the research program “Methodologies, techniques, and computer tools for the preservation, the structural organization, and the intelligent query of musical audio archives stored on heterogeneous magnetic media”, Finalized Project “Cultural Heritage”, Subproject 3, Topic 3.2, Subtopic 3.2.2, Target 3.2.1. We also want to acknowledge the members of the IEEE MX WG (PAR1599) for their cooperation and interest in our work. This work has been made possible by the efforts of researchers and graduate students of LIM.

References

1. De Matteis, A., Haus, G.: Formalization of Generative Structures within Stravinsky’s The Rite of Spring. In: Journal of New Music Research, Vol. 25, N. 1. Swets & Zeitlinger B.V., Amsterdam (1996) 47–76
2. Haus, G., Rodriguez, A.: Music Description and Processing by Petri Nets. In: 1988 Advances on Petri Nets, Lecture Notes in Computer Science, N. 340. Springer-Verlag, Berlin (1989) 175–199
3. Haus, G., Rodriguez, A.: Formal Music Representation; a Case Study: the Model of Ravel’s Bolero by Petri Nets. In: Haus, G. (ed.): Music Processing. Computer Music and Digital Audio Series. A-R Editions, Madison (1993) 165–232
4. Haus, G., Sametti, A.: Modeling and Generating Musical Scores by Petri Nets. In: Languages of Design, Vol. 2, N. 1. Elsevier Publ., Amsterdam (1994) 7–24
5. Ludovico, L.A., Haus, G.: Music Segmentation: an XML-Oriented Approach. In: CMMR 2004 Post-symposium Proceedings, Lecture Notes in Computer Science. Springer-Verlag, Berlin (2004)
6. Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music. The MIT Press, Cambridge (1983)
7. The New Grove Concise Dictionary of Music. Macmillan Publishers, Basingstoke (1994)
8. Mozart, W.A.: Sonata in F major KV332 (300k). In: Complete Piano Sonatas. G. Henle Verlag, München (2005)
9. Petri, C.A.: General Net Theory. In: Proceedings of the Joint IBM & Newcastle upon Tyne Seminar on Computer Systems Design (1976)
10. Peterson, J.L.: Petri Net Theory and the Modelling of Systems. Prentice Hall, New Jersey (1981)
11. Surian, E.: Manuale di Storia della Musica, Vol. 2. Rugginenti Editore, Milano (1992) 511–513