# A Compositional Approach to the Specification of Multimedia Objects Using Petri Nets

*Nikolay A. Anisimov, Alexey A. Kovalenko, Pavel A. Postupalski*
Russian Academy of Sciences

*Son T Vuong*
Universisity of British Columbia

**Abstract.** In this paper we propose a compositional Petri net model, called CoPN, for multimedia synchronization specifications. The salient features of this model, including *macroplaces* and *PN entities*, are presented and the application of the CoPN model to multimedia synchronization specification is discussed via some simple examples. This compositional approach enables the compact and readable specification of complex, large-scale specifications while preserving the fine granularity as well as supporting user interactions.

## 1 Introduction

An important problems in multimedia communication is the synchronization of both static (data, text, image) and dynamic (video, audio) objects. To some extent this problem can be successfully solved with the aid of Petri net models such as Object Composition Petri Nets (OCPN) [3], Extended Object Composition Petri Nets (XOCPN), Dynamic Timed Petri Nets, Time Stream Petri Nets (TSPN) [4] and its extension for user interaction $TSPN_{ui}$ [6]. The comprehensive review of these approaches can be found in [6]. A common problem with the existing specification methods lies in the complexity of the specifications. A viable way to cope with this problem is to apply the notion of compositionality and modularity which was recently introduced in the Petri nets framework [2].

In this paper, we review the salient features of the Compositional Petri net (CoPN) model: *macroplaces*, *Petri net entities* and *access points*; and discuss their applications in multimedia synchronization specification via some simple examples. We also briefly describe a compositional Petri net tool, called $PN^3$-Tool [1], which can be used to graphically generate CoPN specifications.

## 2 Macronets

In a recent work [2], the classic Petri Net model has been extended with the notion of *macroplace*. Such extended nets are called *macronets*. In this section we will show how the use of macroplaces and macronets improves the process of multimedia synchronization specification.



0-8186-7585-3/96 $5.00 © 1996 IEEE

Macronet is a Petri net augmented with a set of macroplaces. First, we introduce the notion of a *simple* macroplace. In general, a macroplace is defined as a set of so-called *internal* places associated with the macroplace. One of the internal places is marked as a head place. Macroplaces are connected with transitions in the same way as a simple (non-macroplace) places. So, a macroplace is considered to be marked if at least one of its internal places is marked. Adding the token to a macroplace is defined as putting a token into its internal head place. Removing a token from the macroplace implies the removal of a token from one of the internal places of the macroplace. The transition of macronet is considered to be enabled if its each input (macro)place has at least one token. Then, firing an enabled transition involves removing a token from each input (macro)place and adding a token to each output (macro)place.

Intuitively, the above definition of a macroplace makes sense only if its associated internal places form a state-machine net (SM-net) (i.e. a Petri net in which each transition has exactly one incoming and one outcoming arc). A macroplace corresponding to a SM-net is called a *simple* macroplace.

Graphically, a macroplace is depicted as a large circle surrounding the part of the net formed by the set of its associated internal places. Incoming and outcoming arcs are attached to a macroplace in the same way as to an ordinary place.

Macronets allow system designers to specify a complex structure in a compact and natural way. For example, the macroplace in Figure 1 represents a cyclic (repeated) background presentation (e.g. audio, video, etc.) which can be terminated by the master stream regardless of the (internal) state (marking) of the background stream.

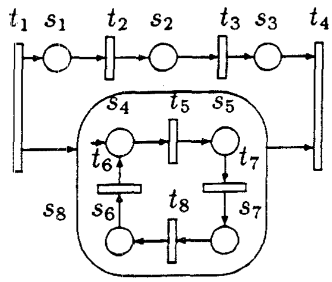


Figure 1: Cyclic background specification

It should be noted that macronets are only compact representations of classic Petri nets with complex internal structures. It is relatively straightforward to transform a macronet to its equivalent normal Petri net.

Using macroplaces also provides us with an additional advantage of supporting user interactions such as **`terminate`**, **`restart`**, **`forward`**. Such user interactions are necessary for video-on-demand applications.

We can see that macronets offer a way of multimedia synchronization specification with more descriptive power while maintaining the compactness of the specification. Using macronets it becomes possible to specify new types of constructs such as cycles as illustrated in Figure 1. Moreover, some user interaction operations such as restart, terminate and forward can be specified in a natural and compact way.

# 3 Compositionality

The notions of *Petri net Entities* and *access points*, for compositional Petri nets were proposed in [2]. In this section, we review these notions and discuss their applications in the compositional and hierarchical specification of multimedia synchronization requirements.

Essentially, a *Petri (PN) net entity* is a generalization of the well-known labelled Petri net where instead of one labelling function we allow for several ones. These labeling functions are called *access points*.

Graphically, a PN entity can be represented in two complementary ways: in a Petri net form and in a diagram form. In the Petri net form, each transition of a classic Petri net is labeled by the set of expressions of the form $id_i : \xi$. This means that the transition is visible as (interaction) $\xi$ at access point (gate) named $id_i$. In the diagram form, a PN entity is represented as a rectangle with small bold-face markers placed on its perimeter, where each marker represents an access point.

We define an operation of composition of PN entities through their access points. The composition of two PN entities involves disjoint union of corresponding nets, followed by the merging of transitions visible at the access points with the same (interaction) names. The formal definition of the operation can be found in [1].

Thus, the architecture of a specification can be described as a set of hierarchical composite Petri net entities where the lowest level entities have the Petri net equivalent representations. We can consider such hierarchical specification as a useful, conceptual one, whereas the formal semantics of such a specification can be defined as the result of performing the compositions of all the entities into an overall normal Petri net.

The main advantage of the compositional specification approach lies in its general and flexible support of compositionality and modularity of specification; it allows systems designers to work in terms of blocks and their interconnection which is gerenally more convenient than working with complex unstructured Petri nets. At the same time, a compositional diagram can be represented as a Petri net at any point of the development process.

Using the compositional approach, we can specify each stream as a PN entity and define the composition of entities according to the interstream synchronization. In order to synchronize the streams we have to label each transition which needs to be synchronized.

The user interaction operations such `stop` and `resume` can be specified in an elegant way. We need to define an extra controller entity to model the user interactions, as illustrated in Figure 2. Each transition of the net which must be controlled (synchronized) must be visible at one access point by the same (interaction) name (e.g. $\zeta$) in order to be synchronized with a special transition in controller object. There is only one transition visible as (interaction) $\zeta$ at the $I$ access point in the controller entity. This transition will be merged with every transition in the main PN entity which is visible as $\zeta$ at the synchronized access point $I$. After firing the `stop` transition every merged transition will be disabled until the `resume` transition firing.

# 4 PN$^3$-Tool

The CoPN model, described in the previous sections, is supported by a special Petri net tool called PN$^3$-Tool. A detailed description of the tool can be found in [1].

The PN$^3$-Tool has a number of modules. The basic editor module supports generation of specifications in Petri net form graphically. The editor includes the standard Petri net editing facilities such as putting, removing, resizing Petri net elements (place, transitions, arcs, marking, labels). Another (algebraic) editor presents the set of algebraic operations for

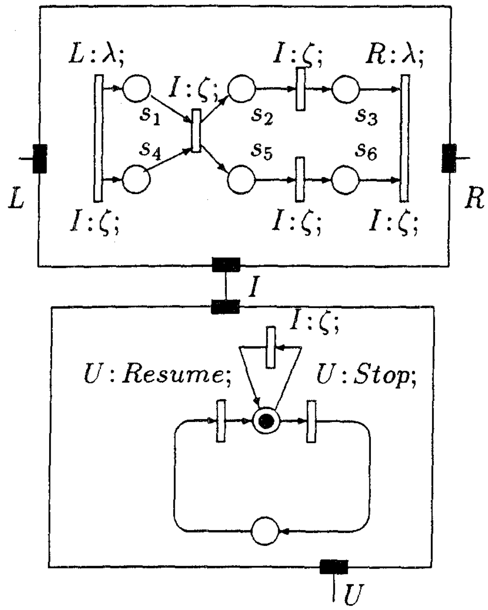


Figure 2: User interaction

Petri net specifications. On the top level is the architectural editor which allows the user to manipulate Petri net entities in diagram form, i.e. in the form of interconnected boxes.

Additional facilities implemented in the PN$^3$-Tool include the visualization module for Petri net simulation, which can used for Petri net analysis for a better understanding of the dynamic behavior of the Petri net; and the verification module for verifying syntactic properties (e.g. boundless, deadlocks, etc) via the construction of Petri net reachability trees.

# 5 Conclusion

In this paper we propose a compositional Petri net model, called CoPN, for multimedia synchronization specifications. This compositional approach enables the compact and readable specification of complex, large-scale specifications while preserving the fine granularity as well as supporting user interactions. While discussing the features of CoPN, we have focused on the compositional aspects of the multimedia synchronization specification, without explictly addressing the time parameters of the specification. We are currently examining the interplaying of times and compositions in Petri nets, and expect to report the results in a future report.

# References


[1] N.A. Anisimov, A.A. Kovalenko, P.A. Postupalski, "Compositional Petri Net Environment", *Proc. of the 1994 IEEE Symposium on Emerging Technologies & Factory Automation: ETFA'94*, November 6–10, 1994, Tokyo, Japan, 420–427.

[2] N.A.Anisimov, M.Koutny, "On compositionality and Petri nets in Protocol Engenering", *Proc. of the 15th Int. Symposium on Protocol Specification, Testing and Verification* June 13–16, 1995, Warsaw, Poland, 57–72.

[3] Little T., Ghafoor A., Synchronization and storage models for multimedia objects, *IEEE Journal on Selected Areas in Communications, pp 52-61,* Volume 8, No 3, April 1990.

[4] Diaz M. and Senac P., "Time stream petri nets: a model for multimedia streams synchronization", *Proceedings of the International Conference on Multi-Media Modelling,* Singapore, November 1993.

[5] S. Vuong, K. Cooper, and M. Ito, "Formal Methods for Modeling Multimedia Synchronization Requirements" *Proc. of ICNP'95 - Int. Conf on Network Protocols*, Tokyo Nov 7-10, 1995.