SAMM: An Integrated Environment to Support Multimedia Synchronization of Pre-orchestrated Data[1]

Ronaldo Mussauer de Lima — Flavio Paiva Junqueira

Paulo André da S. Gonçalves — Otto Carlos Muniz B. Duarte

Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro
COPPE/EE - Programa de Engenharia Elétrica/Departamento de Eletrônica
P.O. Box: 68504 - 21945-970 - Rio de Janeiro - RJ - Brazil
E-mail: {mussauer, flavio, apaulo, otto}@gta.ufrj.br

Abstract

This paper presents the development of the SAMM integrated environment (Multimedia Presentation System), which has been proposed to support synchronized real-time communication and presentation of pre-orchestrated multimedia data. The employed multimedia synchronization technique was based on the eXtended model of the Object Composition Petri-Net (XOCPN). The integrated environment is presented by the description of its main components: the multimedia synchronization service; the modules that automatize processes concerning the synchronization mechanism; and the Composition *graphical application that works as a specialized authoring tool and helps users that want to compose multimedia presentations.*

1. Introduction

The growing interest in distributed multimedia applications which integrate different types of media, such as text, audio, and video [1], has been stimulating the development of systems that support several services required by this class of application. These services include capturing, integration, storage, communication, and a synchronized presentation of multimedia data. The need of integration derives from the dependencies among media, such as temporal, spatial, and causal relations [2]. The temporal composition of several media requires specific synchronization techniques to support multimedia data communication and coherent presentations. Thus, the synchronization techniques use mechanisms to coordinate the temporal and spatial event ordering, once the different media types have specific performance requirements.

[1]This work has been supported by UFRJ, FAPERJ, FUJB, CNPq, REENGE, CAPES, and COFECUB.

Synchronization mechanisms have been proposed in order to fulfill different synchronization requirements [3]. The mechanism to be chosen will depend on the purpose of the multimedia system.

Some data might have a synthetic temporal relation with another data, thus, it is possible to explicitly formulate a temporal relation between them. In this case, these data are denominated as *pre-orchestrated*. The temporal relation is formulated and the data, along with their relation specifications, are stored for further retrieval (e.g., a slide presentation with explaining audio).

A Multimedia System can be defined as a system or an application that supports the integrated processing of several media types with at least one time-dependent medium [4].

This work presents the SAMM integrated environment (*Multimedia Presentation System*), proposed and developed by the Computer Network Group (GTA/UFRJ), that provides support to synchronized presentation of pre-orchestrated multimedia data stored in remote databases. The synchronization mechanism is based on the XOCPN model, proposed by Qazi et al. [5] and Woo et al. [6], because it is well suitable for the purpose of this work.

The paper is organized as follows. Section 2 presents the synchronization specification models. Section 3 presents the SAMM integrated environment, including the automatic XOCPN generation and the graphical application *Composition*. Conclusions are presented in section 4.

2. Multimedia Synchronization

In the multimedia context, the synchronization concept is broad, including the access scheduling of shared resources and the problems of specifying and controlling joint activities of cooperating processes [1]. Using a restricted definition, synchronization techniques are mechanisms used to coordinate the temporal and spatial ordering of the application-related events.

2.1. The Target Application

The target application of the present system aims at providing different multimedia presentations that can be selected and played out by the end user. The presentation QoS parameters can also be set by the user [7]. The system accessed by the end user is defined *as Local System*. The multimedia objects related to all composed presentations are previously stored in a database. This database is accessed by the other system defined as *Remote System*. The complete specification of each presentation is also stored in the database and it is identified and accessed by the Remote System. After the selection of a presentation by the user, the Remote System retrieves the information concerning the respective multimedia presentation and, afterwards, retrieves the multimedia objects to communicate them to the Local System where the presentation will take place.

2.2. Composing Multimedia Objects

Little et al. [8] have proposed a method to specify the synchronization and the relations among data, that uses a model to compose multimedia objects based on timed Petri-nets (*Object Composition Petri-Net* – OCPN), because this model has representation attributes of concurrent and asynchronous events. In the OCPN model, places are related to resource (audio, video, text) utilization and have a duration. Firing of a transition in OCPN is an instantaneous event. Through an OCPN it is possible to specify all temporal relations among multimedia objects [8], such as 'before', 'during', and 'after'. The OCPNs are serial-parallel nets (without feedback), represented by graphs and specified by tuples.

2.3. The eXtended Model - XOCPN

The OCPN model does not take into account the communication requirements in a distributed environment. Considering the demands for communication over a network in a distributed system, a model to specify communication and synchronization mechanisms for preventing and/or compensating anomalies due to the random delays introduced by the network is extremely necessary. In this regard, Qazi et al. [5] proposed an extension model called eXtended OCPN (XOCPN).

The XOCPN model divides objects into sub-objects to coordinate and maintain the continuous temporal relations among media. Each sub-object is called *Synchronization Interval Unit* (SIU). The extended model is generated from an OCPN specification and provides a fine-grain synchronization of isochronous data. The XOCPNs are also represented by graphs and specified by tuples. The XOCPN is defined in the Local and Remote Systems, including functions for synchronization of multiple flows.

3. The SAMM Integrated Environment

For the purpose of the system development, an integrated environment to support multimedia synchronization of pre-orchestrated data has been defined. This section aims to present the main components related to the SAMM integrated environment.

3.1. The Multimedia Synchronization Service and the XOCPN Model Implementation

In the present system, a protocol layer, called *Synchronization*, has been proposed and implemented. The Synchronization layer offers services to applications that demand synchronized communication and presentation of pre-orchestrated data, stored in remote databases, and performs functions related to intra and intermedia synchronization, using specific strategies based on the XOCPN model. This layer is responsible for the synchronized communication between the Local and Remote systems through a computer network [7].

The implementation of the XOCPN-based synchronization mechanism included the representation of Petri-net place and transition by fixed size structures, as well as the manipulation with these elements. Several algorithms have been developed to execute the receiver and transmitter XOCPNs [9].

3.3. Automatic XOCPN Generation

An OCPN object related to isochronous data is divided into a string of SIUs, as mentioned in the section 2.3. Each SIU refers to a part of the object and has the same duration as the other SIUs related to the same object. Moreover, special places, called *Interstream Pacing Points* (IPPs) [5], are inserted in the receiver XOCPN to implement an intermedia synchronization policy. Figure 1(a) shows an OCPN of a simple multimedia presentation, with 25s of total duration, including audio, video, and text objects. From the OCPN specification, it is possible to generate the respective receiver and transmitter XOCPNs. Figures 1(b) and 1(c) show, respectively, the receiver and the transmitter XOCPNs, considering independent channels for each medium. These XOCPNs are related to the OCPN in Figure 1(a), which includes the specification of each object type and the respective duration.

In order to make it a simple process, a module has been implemented to generate automatically the transmitter and the receiver XOCPNs from the OCPN model specifications, [10]. Before the XOCPN generation, a planning procedure must be performed. This procedure is divided in two parts: the retrieval/transmission planning of stored data and the identification of paths or flows in the OCPN that need intermedia synchronization.

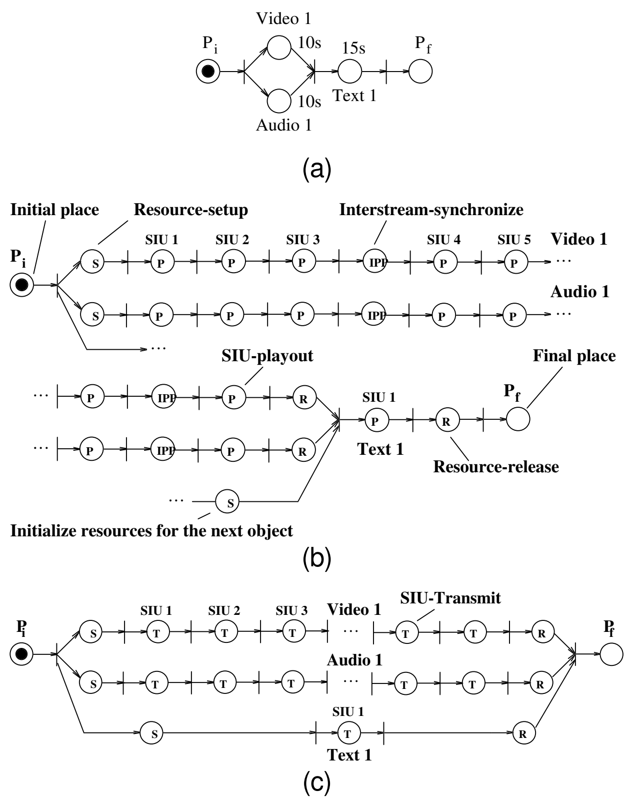

Figure 1: Representation of (a) OCPN, (b) receiver XOCPN, and (c) transmitter XOCPN.

Each multimedia presentation is specified by an OCPN that is stored in a database. A presentation can be identified and accessed by the Remote system through an owner ID. Thus, the Remote system is informed about which presentation was chosen by the user, then it retrieves the respective OCPN information, generates the transmitter XOCPN and sends the OCPN information to the Local system to generate the receiver XOCPN. Afterwards, the Remote system tracks the retrieval planning performed for the transmitter XOCPN in order to retrieve the objects from the database and to send them to the Local system.

The transmitter XOCPN planning refers to two procedures that must be previously performed. Some specific information related to the objects and to the presentation composition must be obtained from the OCPN specification for the transmitter and receiver XOCPN generation process. The first procedure is performed for the transmitter XOCPN generation and refers to the retrieval/transmission planning of the stored data to order the sub-object transmission. The other procedure identifies the paths that need intermedia synchronization, and supplies information used in the transmitter and receiver XOCPN generation. This information is used to split OCPN objects into sub-objects and insert IPP places.

The planning refers to the retrieval time determination, related to each SIU playout time (π_n) of the transmitter XOCPN. Several reasons contribute to the latency (λ) during the data communication from the source to the destination. Some of them are data generation, encoding, compression, transmission delay, and random delays introduced by the network [11]. If a control time T is chosen, such that $T \geq \lambda$, then scheduling the retrieval of some data at T time units prior to the presentation time, it is possible to guarantee a successful presentation.

The algorithm that establishes the minimum control time for each object is based on the procedures proposed by Little and Ghafoor [8]. In these procedures, object retrieval planning works backward on the flow determined by the Petri-net. First, the retrieval time of the last object (ϕ_n) is calculated through the equation $\phi_n = \pi_n - T_n$, thus, it will reach the destination in time to its presentation. Afterwards, the retrieval times of the remainder objects are calculated in an iterative way, still working backwards, up to the time calculation of the first object. It should be emphasized that the objects must be divided into sub-objects, according to the resource type and the intermedia synchronization criterion, before executing this algorithm. This algorithm is shown as follows.

Procedure Transmission_Retrieval_Plan
$\phi_n = \pi_n - T_n$
For all i ***from*** 0 ***to*** n-2 ***do***
If $\phi_{n-i} \leq \pi_{n-i-1} - D_p$
Then (busy channel)
$\phi_{n-i-1} = \phi_{n-i} - T_{i-1} + D_p$
Else (idle channel)
$\phi_{n-i-1} = \pi_{n-i-1} - T_{n-i-1}$
End-Procedure

The information obtained from the OCPN specification is not enough to identify the Petri-net objects that need intermedia synchronization. However, the multimedia presentation model given by the OCPN contains this information implicitly. Moreover, some parallel flows (for instance, audio and video flows) suggest the application of intermedia synchronization, but they might not necessarily need such synchronization. In this sense, it is important to establish a criterion to identify the OCPN objects that really need intermedia synchronization [10].

To identify the intermedia synchronization between two flows, the following decision criterion has been adopted:

- *Two parallel paths must be synchronized if neither of them includes at least a different medium from audio and* *video, and if neither of them becomes separated in two or more paths.*

From this criterion, a procedure was proposed and implemented. Suppose an OCPN in which no place was visited. During the procedure execution, places are set as visited in order to prevent loops. The procedure is described as follows, supposing N a given OCPN, p a place of N, t a transition of N, A(t) a set of output places of the transition t, B(t) a set of input places of the transition t, F(t,p) a flow which starts in transition t and finishes in place p, Sync_Policy is the set of all flows that match the adopted criterion, and R(p) the resource related to p.

```
Procedure Intermedia_Sync(t)
  For all p∈A(t)
      If p is not visited Then
        If R(p) = (audio, video) Then
         Reverse_Path(t1,t,p),p∈A(t1)
         If p∈B(t1) and t ≠ tf
          Intermedia_Sync(t)
      Else
        Set p as visited
        If p∈B(t1) and t ≠ tf Then
          P ∉ Sync_Policy
        Intermedia_Sync(t1)
End-Procedure
```

The execution of the `Intermedia_Sync(t)` procedure must begin with OCPN initial transition, and, afterwards, all OCPN structure must be tracked until the final transition. For all transitions that have more than one input place, the algorithm must track backwards, taking a different path, in order to identify the need of intermedia synchronization. For such task, the procedure `Reverse_Path(t1,t,p)` was implemented. For each transition given as a parameter to the `Intermedia_Sync(t)` procedure, it must be verified if there is any place started by that transition which resource type is 'audio' or 'video'.

```
Procedure Reverse_Path(t1,t,p)
   For all p1∈A(t1)
     If all p2 ∈ F(t,p1)=(Audio,Video)
      Then If all t2 ∈ F(t,p1), A(t2)=1
       Then F(t,p1),F(t,p) ∈ Sync_Policy
     For all p2 ∈ F(t,p1)
         Set p2 as visited
End-Procedure
```

The next section presents the *Composition* graphical application, proposed to help users to compose multimedia presentations, producing and saving the essential information to the XOCPN generation.

3.4. Multimedia Object Composition

The creation of a multimedia presentation refers to the different ways in which multimedia objects can be composed. As mentioned earlier, the OCPN model has been chosen to specify the temporal relations among several kinds of multimedia presentation elements. Thus, the OCPNs related to the multimedia presentations are stored in a database. In such case, all the information about each OCPN is stored in a file. Through a corresponding table it is possible to identify which file contains the OCPN data related to the presentation selected by the user. This file contains a list of all OCPN information. Through another table it is possible to associate the identified file with the location, in the database, of each resource specified by the OCPN, as well as the characteristics of its respective media.

The automatic XOCPN generation procedures require the above-mentioned file. In order to make the multimedia object composition easier, generating and storing automatically the required files to the XOCPN generation, the *Composition* graphical and interactive application was proposed and developed. For the integrated environment presented in this paper, this graphical application works as a specialized authoring tool, based on the OCPN model, and helps specialized users that want to compose multimedia objects and to store the composition information in a database. Thus, the system end user can select and playout the presentations from a remote system.

The *Composition* graphical has been developed using the Tcl/Tk toolkit [12]. The window for composition and edition of multimedia objects is shown in Figure 2.

An important feature of this software is to perform the necessary syntax verifications in the OCPN diagram composed by the user. It is essential to assure the diagram correctness, and the definition of the respective duration and specification file of each object. The following verifications are performed in the diagram: whether the initial and final places have been defined or not; if all places have defined properties; if there are places without linkage with the remainder OCPN diagram; if all transitions have at least one input and one output direct arcs; if all places (except the initial and final) have necessarily input and output direct arcs; and, finally, whether the parallel objects have the same duration or not. If any of these conditions fails, all objects related to the failure are automatically highlighted in the window and an error message is returned to the user. After these verifications, the diagram correctness is guaranteed.

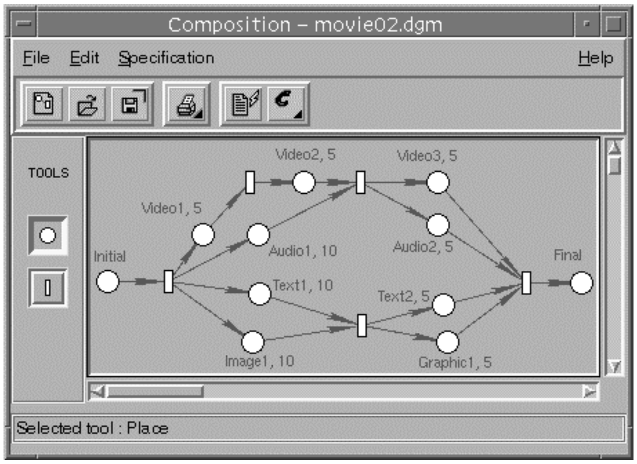

Figure 2: Window for composition and edition.

Figure 3 shows a diagram of the *Composition* modules and other modules related to the integrated environment. The diagram also represents how these modules are interconnected to the Local and Remote systems. *Composition* has four modules, as follows. The first one, called User Interface, defines the interface with the user, interactively controlling and monitoring the composition of the OCPN diagram. The second one, called Property Definition, is responsible for assigning the multimedia object properties. In this module, the place type (initial, final, object or delay) is specified, and in case the place type is 'object', the resource type, name and duration of such object are identified. The object specification file is also identified and the database is accessed for reading the object properties. The third one, called OCPN Generation, is responsible for the required processing to generate the OCPN specification. After the syntax verifications, this last module accesses the multimedia objects database to read the identification number of each object [10] and to generate the OCPN specification. The fourth one, called Management of OCPN Diagram Files, manages all procedures for opening and saving diagram data files.

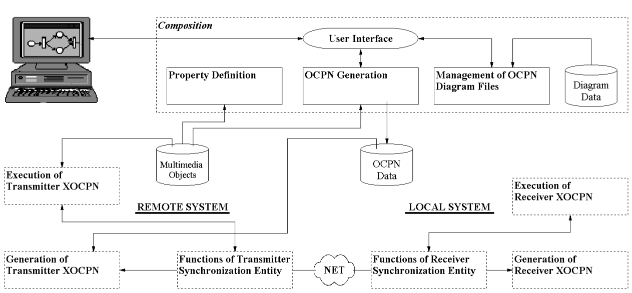

Figure 3: Diagram of the *Composition* and integrated environment modules.

4. Conclusions

This paper has presented the SAMM integrated environment for supporting synchronization of pre-orchestrated multimedia data. The synchronization mechanism has been based on the XOCPN model and takes into account synthetic temporal relations among stored multimedia objects. The implemented XOCPN transmission/retrieval planning allow multimedia application execution that matches network conditions.

The *Composition* graphical application has been implemented in Tcl/Tk and benefits from the portability of these interpreted languages. The main feature of this application is the provision of a graphical and simple environment that helps authors to compose multimedia presentations.

5. References

[1] Steinmetz, R., "Synchronization Properties in Multimedia Systems", *IEEE J. Select. Areas Commun.*, vol. 8, nº 3, April 1990, pp. 401-12.

[2] Courtiat, J.-P.; Carmo, L.F.R.C. and Oliveira, R.C., "A General-Purpose Multimedia Synchronization Mechanism Based on Causal Relations", *IEEE J. Select. Areas in Commun.*, vol. 14, nº 1, pp. 185-95, January 1996.

[3] Ehley, L.; Furht, B. and Ilyas, M., "Evaluation of Multimedia Synchronization Techniques", *Proc. of the International Conference on Multimedia Computing and Systems*, May 1994, pp. 514-19.

[4] Steinmetz, R. and Nahrdstedt, K., "Multimedia: Computing, Communications & Applications", *Prentice-Hall*, 1995.

[5] Qazi, N.U. et al., A., "A Synchronization and Communication Model for Distributed Multimedia Objects", *Proc. ACM Multimedia '93,* Anaheim, CA, August 1993, pp. 147-56.

[6] Woo, M. et al., "A Synchronization Framework for Communication of Pre-orchestrated Multimedia Information", *IEEE Network*, vol. 8, nº 1, January 1994, pp. 52-61.

[7] De Lima, R.M., "Um Serviço de Sincronização Multimídia de Dados Pré-orquestrados", *Master Thesis*, PEE-COPPE/UFRJ, June 1996.

[8] Little, T.D.C. and Ghafoor, A., "Multimedia Synchronization Protocols for Broadband Integrated Services", *IEEE J. Select. Areas in Commun.*, vol. 9, nº 9, December 1991, pp. 1368-82.

[9] De Lima, R.M. and Duarte, O.C.M.B., "Multimedia Synchronization System for Pre-orchestraded Data Communications", *SBT/IEEE International Telecommunications Symposium (ITS'98)*, São Paulo, SP, August 1998, pp. 267-72.

[10] Junqueira, F.P., "Geração Automática de XOCPN para um Serviço de Sincronização", *Senior Project*, Escola de Engenharia-DEL/UFRJ, February 1997.

[11] Little, T.D.C. and Ghafoor, A., "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks", *IEEE Computer*, October 1991, pp. 42-50.

[12] Ousterhout, J.K., "Tcl and the Tk Toolkit", *Addison-Wesley Publishing Company,* 1994.