Multimedia Synchronization System for Pre-orchestrated Data Communications[1]

Ronaldo Mussauer de Lima Otto Carlos Muniz B. Duarte
mussauer@gta.ufrj.br otto@gta.ufrj.br
Grupo de Teleinformática e Automação (GTA)
COPPE/EE - Programa de Engenharia Elétrica/Departamento de Eletrônica
Universidade Federal do Rio de Janeiro
P.O. Box 68504
21945-970 - Rio de Janeiro - RJ - Brazil
phone: +55-21-260-5010 fax: +55-21-290-6626

Abstract

This work presents a multimedia synchronization system, that supports real-time communication and presentation of pre-orchestrated data. The synchronization mechanisms and the developed multimedia synchronization service are based on the XOCPN model. The paper describes aspects concerning the system implementation, including the implementation architecture and environment, the XOCPN data structures and the specialized modules related to the synchronization mechanism. The presented system provides a complete support to applications that communicate multimedia data and helps users to compose multimedia objects. The proposed implementation seems to be very efficient.

1. Introduction

The advance in communication systems, including high-speed networks and protocols, has enabled the development and implementation of distributed multimedia applications, which integrate several types of media such as audio, video, text and graphics [1]. The need of integration derives from the dependencies among media, such as temporal, spatial and causal relationships [2]. Each media type has specific presentation, storage and communication requirements and may experiences random delays during transmission and retrieval. In order to support multimedia data communications and coherent presentations at the destination, it is essential to assure the synchronization of data elements.

Several synchronization techniques have been proposed [2]-[6], each one presenting advantages and disadvantages, depending on the type of application, the data sources, the communication system, the required quality of service (QoS), the available resources, etc. [7]

This work focuses on synthetic temporal relationships among media. The synchronization mechanism is based on the eXtended Object Composition Petri-net (XOCPN) model [5,8], which provides a complete presentation and communication schedule for pre-orchestrated multimedia data in a distributed system.

This paper presents a multimedia synchronization system for pre-orchestrated data communications, that uses XOCPN-based mechanisms. In this regard it presents a synchronization service and protocol, and an implementation scheme, that includes specialized modules, specific data structures and a graphic application.

This article is organized as follows. Section 2 describes the basic concepts related to the multimedia synchronization. Section 3 presents the Synchronization Layer, including the target application, the OCPN and XOCPN models, the synchronization service and strategies. Section 4 presents the system implementation, including the implementation architecture and environment, automatic generation of XOCPNs and composition of multimedia objects. Finally, conclusions are presented in Section 5.

2. Basic Concepts

This section presents some definitions concerning multimedia synchronization. *Isochronous* data refer to a data generated at a constant rate, from a continuous event. These data have an implicit temporal relation in its generation, requiring rate-controlled transmission (e.g., audio and video). *Anisochronous* data have no relation with generation, transmission or presentation rates. In this case, the spatial composition of the presentation is the most important data requirement (e.g., text and graphics).

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

Pre-orchestrated data concern to synthetic temporal relationships among data. In this case, the time dependencies are explicitly formulated (e.g., graphical presentation with explaining texts). *Multimedia object* is the basic multimedia element. An object refers to a multimedia information unit, complex or not, comprising data elements of a specific media. *Flow* is a temporal sequence of multimedia objects. *Intramedia synchronization* refers to the temporal dependencies between multimedia objects related to only one media. *Intermedia synchronization* refers to the temporal dependencies among multimedia objects related to more than one media.

3. The Synchronization Layer

In the proposed system, the synchronization functions are performed in a layer named *Synchronization*. These functions are responsible for the intra- and intermedia synchronization, based on the XOCPN model. This layer provides services to applications that demand synchronized communication and presentation of pre-orchestrated multimedia data, stored in remote databases.

The Target Application

The system is divided in two parts: the *Local System*, that is accessed by the application user, and the *Remote System*, that accesses the database which stores the information concerning the multimedia presentation.

The target application aims to provide synchronized presentation of multimedia data, that is stored in remote databases. Application users, at the Local System, can select among different multimedia presentations and can set the QoS parameters of the presentation (e.g., frame rate, image dimensions, encoding type, etc.). The complete specification of each presentation is also stored in the database and it can be 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.

Figure 1 shows the application and synchronization layers in the hierarchical system architecture.

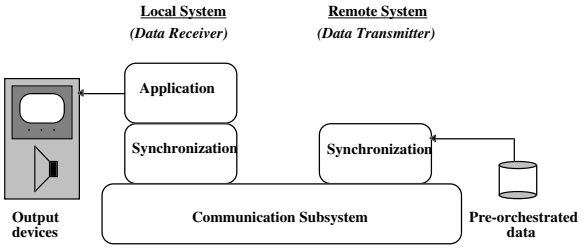

Figure 1: The hierarchical layers of the system.

The synchronization techniques employed in this system are based on the XOCPN model, from the fact that it is well-suitable for the propose of this work, and take into account the temporal dependencies among media.

The OCPN and XOCPN Models

The Object Composition Petri-net (OCPN) model is a special type of timed Petri-net that includes enhanced characteristics [9]. Each place in OCPN is related to resource (audio, video, text, etc.) utilization and has a time duration. Firing of a transition in OCPN is an instantaneous event. Through OCPN it is possible to specify all temporal relationships like 'before', 'during', 'after', etc. The OCPNs are serial-parallel nets (without feedback) represented by graphs and specified by the tuple {T, P, A, D, M} (explained further). Figure 2 shows a simple OCPN with 40s of total presentation time.

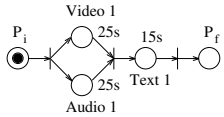

Figure 2: Representation of an OCPN.

XOCPN is an extension of the OCPN model including communication features for distributed environments [5]. Through XOCPN it is possible to prevent and compensate network transmission anomalies. It is suitable for continuous temporal relationships among objects (isochronous data) which demands rate-controlled communication and presentation. The XOCPN model divides objects into sub-objects called Synchronization Interval Units (SIU) in order to provide a fine-grain synchronization. XOCPNs are also represented by graphs and specified by the tuple {T, P, A, D, M, Y, Z}, where [8]:

$T = \{t_1, t_2, ..., t_n\}$ is a set of transitions;

$P = \{p_1, p_2, ..., p_m\}$ is a set of m places;

$A = \{TxP\} \cup \{PxT\} \rightarrow I$, $I = \{1, 2, ...\}$ is a set of directed arcs;

$D: P \rightarrow \{d_1, d_2\}$ where d_1, d_2 represent the delay before an action and duration of an action, respectively;

R: P → {r_1, r_2,..., r_k} is a mapping from the set of places to a set of object types;

M: P → I, I = {0, 1, 2,...} is a mapping from the set of places to the number of tokens in a place;

Y: P → {Resource-setup, Resource-release, SIU-playout, SIU-transmit, Interstream-synchronize} is a mapping from the set of places to a set of action types to be performed during communication;

Z: P → {Addresses} is a mapping from the set of places to the addresses for QoS parameters, synchronization requirements and a SIU.

For the purpose of multimedia communications, a receiver and a transmitter XOCPN are generated from an OCPN specification. Each place in OCPN is replaced by a string of SIUs. Figure 3 shows the receiver and transmitter XOCPN corresponding to the OCPN of Figure 2. According to a pre-specified schedule, the receiver XOCPN playouts the multimedia objects at the user-site. To maintain the pace among related isochronous data flows, the Interstream Pacing Points (IPPs) are inserted in the receiver XOCPN. For example, it is possible to perform synchronization strategies to keep audio and video flows synchronized within 150 ms. Otherwise, the presentation quality degrades [8]. According to a pre-specified schedule, the transmitter XOCPN at the source-site delivers the multimedia objects to the network.

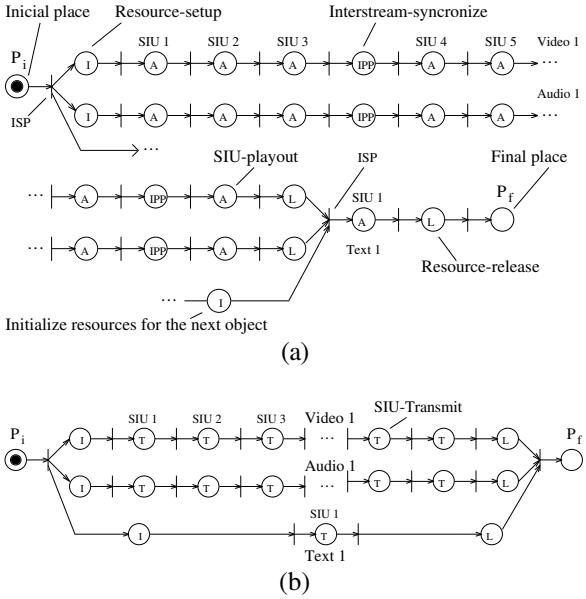

Figure 3: Representation of (a) receiver and (b) transmitter XOCPN.

The Synchronization Service and Protocol

The XOCPN-based synchronization service provides support to real-time distributed applications that communicate pre-orchestrated multimedia information stored in a remote database. The synchronization service is defined by its service primitives [10]. Since the user-site is located at the Local System, there is not an application entity at the Remote System. Thus, the service primitives are defined for the receiver synchronization entity, as shown in Table 1.

Table 1: Synchronization service primitives.

Primitive	Description
SINC-INITIALIZE.Request	Request to initialize the presentation
SINC-PLAYOUT.Indication	Indication of SIU's presentation
SINC-STOP.Request	Request to stop the presentation
SINC-STOP.Indication	Indication of fail or end of SIU

The specified synchronization protocol includes five Protocol Data Units (PDUs). Table 2 shows the PDUs and their brief descriptions.

Table 2: Protocol Data Units.

PDU	Description
IREQ	Initialize request for the presentation procedures
INETDT	Initialize request for XOCPN Generation
IREADY	Confirmation of XOCPN generation
IFAIL	Indication of initialization fail
SIUDT	Sub-object (SIU) data

Synchronization Strategies

The non-deterministic network behavior may introduce errors and jitters during data communication. It is essential for the user-site to compensate problems related to slow flows, maintaining the temporal relations among data, specified by XOCPN, and providing the desired quality of presentation. These actions include different synchronization strategies [5], like blocking, non-blocking and restricted blocking. The choice for a specific strategy depends on the type of synchronization (intramedia or intermedia).

In the blocking strategy, the presentation process suspends until the expected SIU arrives. In the non-blocking strategy, the presentation process continues

even though the expected SIU has not arrived. In the restricted blocking strategy, the presentation process suspends for a pre-specified time period. After this period, the process skips the late SIU and may, for instance, repeat the presentation of the last SIU.

In this work, we adopted the restricted blocking strategy for intramedia synchronization and for intermedia synchronization related to the transitions among different flows. On the other hand, we adopted the non-blocking strategy for intermedia synchronization related to IPP places. The blocking strategy was not used in order to avoid interrupting during playout of multimedia objects.

4. System Implementation

The system implementation has been divided in two parts. The first part is related to the synchronization service and protocol, including the implementation of State Machines, interfaces between layers, data communication management (handling of data structures and interfaces, connections, memory, timers and tasks). The second part is related to the XOCPN-based synchronization mechanisms, involving the representation of Petri-net's places and transitions. Besides, several modules concerning the execution of the transmitter and receiver XOCPN were developed. These modules include procedures to generate the XOCPN from an OCPN specification, execute places and fire of transitions, handle and compensate SIU's delay, apply intermedia synchronization strategies and control intramedia synchronization, etc. Since the target application involves multimedia data, like audio and video, it was also developed procedures to capture, compress, store, retrieve and transmit this type of data.

Implementation Architecture

An efficient implementation architecture [11] was defined to support multimedia data communication and it has been used for the purpose of system implementation. This architecture (Figure 4) defines three specialized modules:

- A priority-based task scheduling (TS), that avoids unnecessary queue tests and takes advantage of required atomicity of some functions;
- An optimized memory management (MM) with fixed and variable size buffer allocation/deallocation and separated transmitter and receiver memory regions, keeping fragmentation as a minimum;
- A timer management (TM) efficiently implemented to avoid unnecessary tests in each timer clock interrupt, allowing the creation of a large number of timers.

The interface between the *Synchronization Layer* and adjacent layers consists of a pair of queues per connection and a set of data structure access procedures.

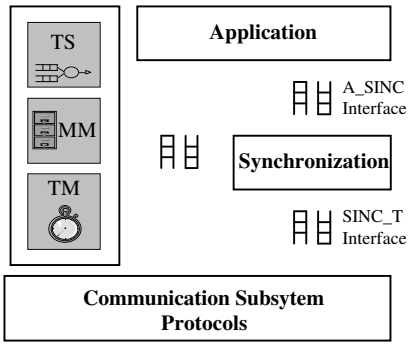

Figure 4: Implementation architecture for synchronized communications.

XOCPN Pointer-based Data Structures

Each XOCPN place and each transition are represented by a fixed size structure. The *Place_Structure* contains fields related to the XOCPN tuple. The *Transition_Structure* contains one field to point to all output places and another field corresponding to the number of all the input places to be executed before firing of a transition. Figure 5 shows the linked structures, based on Figure 3a.

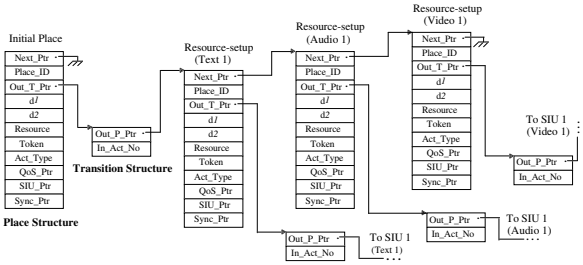

Figure 5. XOCPN place and transition structures.

Automatic Generation of XOCPNs

Each multimedia presentation is specified through an OCPN by using ASCII format files stored in the database. The transmitter and receiver XOCPN are generated from the information contained in these files. For the purpose to automatize this process, it was developed a specific module to generate the XOCPNs automatically, including the schedule of multimedia objects retrieval and transmission [12].

The State Machines and the proposed modules related to XOCPN execution and automatic XOCPN

generation for the receiver-site are shown in the diagram of Figure 6.

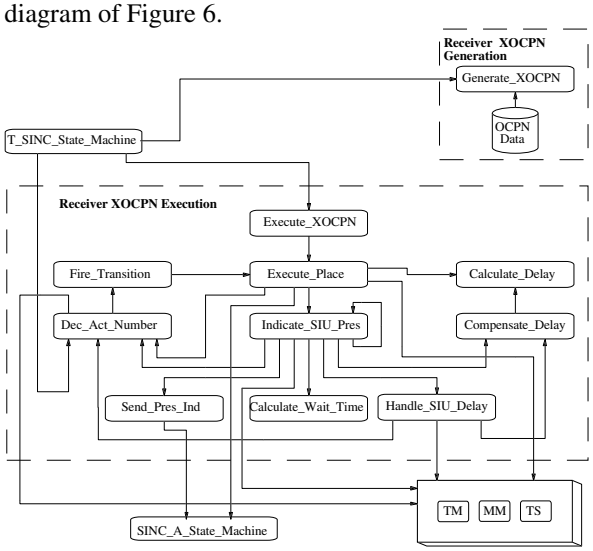

Figure 6: Procedure diagram for the receiver XOCPN.

Interactive Composition of Multimedia Objects

For the purpose to make easy the composition of multimedia objects, it has been developed a graphic and interactive application that works as a specialized authoring tool and helps users that want to compose multimedia objects. This application allows users to compose OCPNs graphically, generating the OCPN specification as well as the essential files to generate the XOCPNs afterwards, storing them in databases. The application was developed using the *Tool Command Language* Tcl/Tk.

The application's user-interface has basic resources for file handling, resources to handle objects of the OCPN graph, and resource to generate the OCPN specification. Authors of multimedia presentation can compose an OCPN graphically and should assign, for each object, its name, its duration, and its specification file, which contains information related to characteristics of the multimedia object (e.g., for an audio object: encoding type, sample rate, bits per sample, data file size and name, etc.). Figure 7 shows the application's user-interface, presenting the window for composition and edition of multimedia objects.

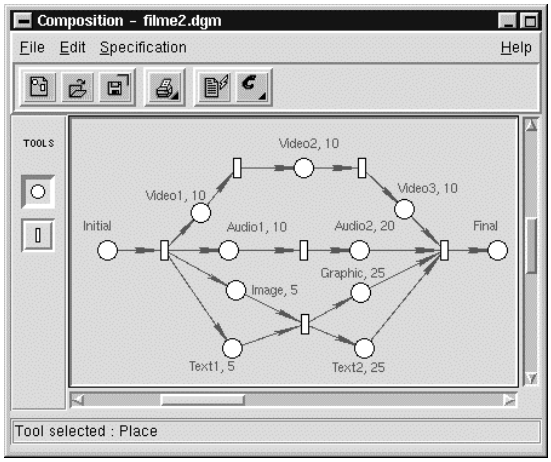

Figure 7: User-interface for composition and edition of multimedia objects.

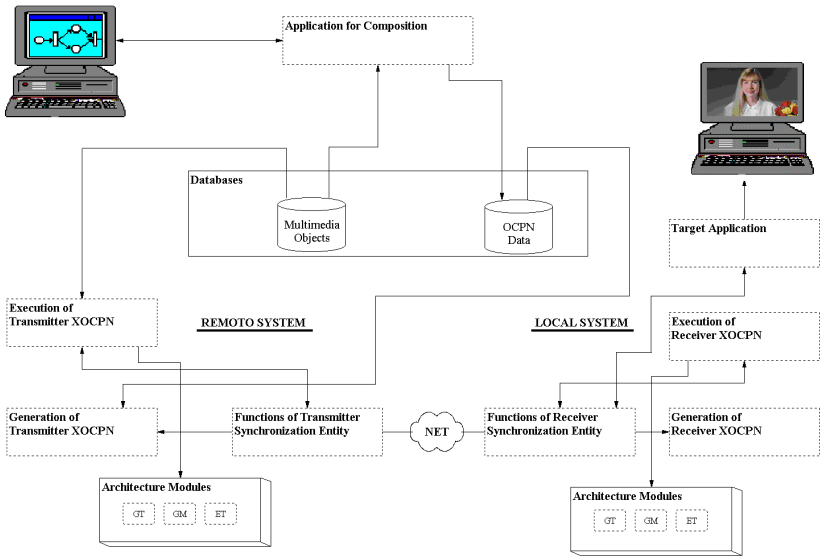

Figure 8: General diagram of the main system modules.

An important characteristic of this application is the validation of the composed OCPN. It is essential to guarantee the composition of a coherent OCPN and to guarantee the definition of the duration and the specification file for all OCPN objects. If these conditions are not fulfilled, all incorrect OCPN components are tagged and an error message is sent to the application user.

Implementation Environment

It has been developed a system prototype using Sparc Stations, Solaris operating system and a LAN environment, including the modules described in this article. The main modules of the system was coded in C language.

Figure 8 shows a general diagram including the main system modules and its links in the Remote and Local Systems.

5. Conclusions

This work has presented a multimedia synchronization system, dedicated to communication and presentation of pre-orchestrated multimedia data, stored in a remote database. The synchronization mechanisms are based on the XOCPN model and take into account synthetic temporal relationships among multimedia objects. The definition of the service primitives and the protocol data units was presented as well as the implementation aspects of the system.

The proposed pointer-based data structures make easy the access to a place and transition in the XOCPN, as well as the implementation of intermedia synchronization strategies. High performance is one of the most important features of the implementation architecture due to the reduction of memory copies. The proposed implementation, including synchronization service, the specialized modules and the graphic application, results in a complete system that supports the composition of multimedia objects and the distributed multimedia presentations.

Acknowledgment

The authors thank Flávio Paiva Junqueira and Paulo André da Silva Gonçalves for their valuable contributions to this work.

References

[1] Blakowski, G. and Steinmetz, R., "A Media Synchronization Survey: Reference Model, Specification, and Case Studies", *IEEE J. Select. Areas Commun.*, vol. 14, nº 1, pp. 5-35, January 1996.

[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 Commun.*, vol. 14, nº 1, pp. 185-95, January 1996.

[3] Nicolaou, C., "An Architecture for Real-Time Multimedia Communication Systems", *IEEE J. Select. Areas Commun.*, vol. 8, nº 3, pp. 391-400, April 1990.

[4] Ramanathan, S. and Rangan, P.V., "Feedback Techniques for Intra-Media Continuity and Inter-Media Synchronization in Distributed Multimedia Systems", *Computer Journal*, vol. 36, nº 1, pp. 19-31, 1993.

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

[6] Escobar, J.; Partridge, C., and Deutsch, D., "Flow Synchronization Protocol", *IEEE/ACM Trans. on Networking*, vol. 2, nº 2, pp. 111-21, April 1994.

[7] Ehley, L.; Furht, B., and Ilyas, M., "Evaluation of Multimedia Synchronization Techniques", *Proc. of the International Conference on Multimedia Computing and Systems*, pp. 514-19, May 14-19, 1994.

[8] Woo, M.; Qazi, N.U., and Ghafoor, A., "A Synchronization Framework for Communication of Pre-orchestrated Multimedia Information", *IEEE Network*, vol. 8, nº 1, pp. 52-61, January 1994.

[9] Little, T.D.C. and Ghafoor, A., "Synchronization and Storage Models for Multimedia Objects", *IEEE J. Select. Areas Commun.*, vol. 8, nº 3, pp. 413-27, April 1990.

[10] De Lima, R.M. and Duarte, O.C.M.B., "Implementação de um Serviço de Sincronização Multimídia para Comunicação de Dados Pré-orquestrados", *XV Simpósio Brasileiro de Redes de Computadores (SBRC'97)*, São Carlos, SP, Brazil, pp. 19-35, May 1997.

[11] Albuquerque, C.V.N.; Nunes, M.D., and Duarte, O.C.M.B., "Performance Measurements in a Manufacturing Communication System", *Proc. of the International Symposium on Circuits and Systems (ISCAS)*, London, UK, pp. 69-72, May 30-June 2, 1994.

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