Guest Editorial
Synchronization Issues in Multimedia Communications

I. Introduction

IN GENERAL, multimedia synchronization denotes a temporal, spatial or even logical relationship between objects, data entities, or media streams. In the context of multimedia computing and communications, it has become customary to address by synchronization the temporal relationship only [1], [2]. This notion is still very broad and it captures a large variety of issues, including inter-process communication mechanisms and concurrent programming languages. Hence, in this editorial and in all contributions to this special issue, multimedia synchronization will refer to the temporal relationship among media.

As described in [3], a multimedia system is characterized by the integrated computer-controlled generation, manipulation, presentation, storage, and communication of independent discrete (i.e., time independent, like text and graphics) and continuous (i.e., time dependent, like audio and video) media data. The digital representation of any data, and the synchronization between various kinds of media data, are the key issues for this integration. Hence, multimedia synchronization is needed to ensure a temporal ordering of events in a multimedia system.

In several publications IEEE reflects the member demands for information related to multimedia: The *IEEE Computer Magazine* devoted a first issue on Multimedia Communications in October 1985. The most recent issue to focus on multimedia appeared in May 1995. In addition, the *IEEE Multimedia Magazine*, published since 1994, is attracting a large readership. JSAC had a first issue on Multimedia Communication published in April 1990. Papers from that issue are cited throughout the multimedia community frequently, e.g., see this JSAC issue. Almost all ACM and IEEE Communication and Computer Society supported workshops and conferences related to multimedia include at least one paper related to multimedia synchronization. In conjunction with the 1995 IEEE International Conference on Multimedia Computing and Systems, T. D. C. Little and the first two guest editors of this JSAC issue organized the first and very successful workshop on multimedia synchronization.

Considerable work has been performed in the synchronization research and development area. However, there exists no common terminology and there has been a lack of comparative work. This was the motivation for the two overviews and reference models. Fundamental work for understanding the user demands is also still in its infancy. A contribution to this issue is given in the second section. With this notion of user demands, the next section of papers addresses the topic of how to model multimedia synchronization (and therefore how to define the temporal relationships). Mechanisms to perform this comprise the next section, followed by a discussion on fault tolerance issues only (i.e., what shall happen in case of failures and how to avoid them). Finally, different implementation issues are covered, concluding with fully implemented systems and the related experiences.

This issue is therefore organized to include the following six sections:

Manuscript received September 1995.

N. D. Georganas is with the University of Ottawa, Ontario, Canada, K1N 6NS.

R. Steinmetz is with the IBM European Networking Center, Heidelberg, Germany.

T. Nakagawa is with the NTT Data Communications Systems Co., Kawasaki, Japan.

Publisher Item Identifier S 0733-8716(96)00223-5.

II. Overviews and Reference Models

The first section is comprised of papers providing the reader with an overview of various multimedia synchronization issues. For researchers and developers active in the multimedia synchronization domain, it has been difficult until now to obtain a comprehensive overview on the various requirements, models, and mechanisms already in place. In the multimedia synchronization field there were few terms established and new upcoming papers tend to introduce again a new terminology to describe their approach. The reference models as described in the first two papers provide the reader with a framework for categorizing and comparing different approaches.

Blakowski and Steinmetz, introduce a common terminology and start with a description of the requirements from the user perspective. Subsequently, they survey the most prominent synchronization models to describe the application needs. Hence, using these models, the specification of an interface can be derived. The layered reference model presented in this paper distinguishes various layers, among which are: 1) an intra-stream layer which applies to the temporal constraints within one data stream, 2) a stream layer which allows the application to operate on continuous media streams, as well as groups of media streams (including inter-stream synchronization), 3) the object layer which hides the differences between discrete and continuous media, and 4) the fuzzy layer which allows for the specification and use of playout templates. These templates incorporate the definition of fuzzy time constraints. The paper concludes with a survey of existing work as case studies.

With their temporal reference framework for multimedia synchronization Perez-Luque and Little present a uniform theoretical basis for synchronization and temporal specifications. They characterize temporal scenarios, elaborate on various models of time, and develop a new synchronization framework. This reference model is used to compare various synchronization models, namely the timeline approach, OCPN, Firefly, path expressions, clock hierarchies, and others.

Rangan *et al.* describe continuity and synchronization issues in MPEG. This international standard addresses synchronization in various layers: The encoding layer handles compression and decompression, interleaves the packets of different media, and assembles a stream of packs. Via the system reference clock, stamps are inserted at the header of the packs and these packs are the unit for maintaining continuity at playback. The packets are the basic units for maintaining inter-media synchronization using decoding and presentation time stamps. This paper provides a very interesting overview on various system capabilities and the implied restrictions from the multimedia synchronization point of view.

III. User Perception

The requirement to keep temporal relationships within certain time borders must be derived from the (subjective) user perception of multimedia synchronization. If data are "out of synch," the human perception tends to identify this error as data being artificial and annoying. Therefore, the goal of any multimedia system is to present all data without any synchronization error. The achievement of this goal requires a detailed knowledge of the notion of synchronization errors at the user interface.

Steinmetz has worked out a detailed table on the temporal inter-media synchronization constraints to be understood as an initial guide to system implementors. He undertook a large set of experiments to retrieve the appropriate value for lip synchronization demands: a skew of 80 ms is allowed between audio and video. Subsequently, this paper presents the results of pointer synchronization experiments. Finally, the temporal aggregation of more than two media is discussed as well as the perception of jitter to the user is outlined.

IV. Modeling

The users' and applications' demands in terms of synchronization, must be modeled and specified in order to make it available to the computing system. A variety of models have been worked out, proposed, and implemented. This section of papers deals with a set of very different approaches, all unique in their sense. In particular, they are based on an algebraic background, on Petri Nets, on a probabilistic attribute context, free grammar, on events, and on an extended finite state machine.

Haindl elaborates a model based on a description of the start and end data unit presentation points including incomplete timing. This hierarchical model relies on algebraic, stochastic, and event-based fundamentals. It allows one to model any type of parallel synchronization. Haindl concludes with a discussion of synchronization failure treatment.

Senac *et al.* extend the Time Stream Petri Nets model toward a hierarchical approach. Hierarchical extensions, based on an analysis of Time Stream Petri Nets in the context of modeling hypermedia nodes and links, are proposed. Additionally they show how logical and temporal synchronization can be combined into hypermedia synchronization schemes. This paper concludes with an example, and some considerations on the modeling methodology and tools.

Raghavan *et al.* present their probabilistic attribute context free grammar approach to model any kind of temporal multimedia synchronization issue. This is compared to Petri Net approaches. A traffic source model of an orchestrated presentation is elaborated using this new approach. The authors expect that this traffic model will be very useful in performance engineering studies related to multimedia system design issues.

Schnepf *et al.* describe their FLexible Interactive Presentation Synchronization (FLIPS) model, which addresses the specification of coarse synchronization for flexible presentations. In particular, it addresses variable duration media such as animations and simulations, as well as exhaustive user interaction capabilities. With an example comprising a narrative, slides and background music, the various features of this system are shown.

Ates *et al.* makes use of timed CSP (Communicating Sequential Processes) for the specification, verification and simulation of multimedia synchronization. A model based on timed CSP with temporal intervals is presented. Finally, simulation models based on this work are presented. As an example, lip synchronization is discussed.

Huang and Lo apply an extended finite state machine model to formally specify the temporal relationships between various media streams. They address intra-stream synchronization using an Actor and inter-stream synchronization having a Synchronizer. This allows modeling both centralized, as well as distributed, synchronization issues. This model was implemented together with an authoring system.

V. Mechanisms

Whereas models of synchronization allow for a coherent view and specification of the required demands, mechanisms actually perform this synchronization. These mechanisms cover synchronization algorithms and protocols. They operate on a local basis, as well as in a networked environment. However, the question of which system component shall incorporate these mechanisms is not a trivial one. Most approaches tend to incorporate them into the communication system, some into the operating system, others into application, authoring tools, or even database systems.

Li and Ofek elaborate on a distributed source-destination synchronization mechanism which does not rely on a common reference clock for the whole distributed set-up. Here a distributed in-band signaling for the generation of a global clock is presented. These results can be used to implement synchronous protocols such as time stamping.

Akyildiz and Yen present a mechanism to achieve synchronization for groups of sending and receiving applications. The only prerequisite is knowledge on the maximal delay which might occur, no common reference clock is required. The mechanism is addressed as a protocol for group communication. Akyildiz also provides a set of simulation results.

Rothermel and Helbig describe their abstraction for grouping and controlling media streams in terms of clock hierarchies. The used model is based on clocks which are related to the components of the multimedia application described. The related mechanism allows the user to control, i.e., to start, to scale and to pause, streams in a networked environment. The mechanism works with clocks attached to sinks as well as to sources. The mechanism and the related model is part of the CINEMA prototype which is in progress.

Courtiat *et al.* develop a conditional delivery mechanism for intra-media and inter-media synchronization. This mechanism is specified in Real-Time LOTOS, which is also validated in the context of an application scenario. The mechanism resides in the communication system between the transport layer and the application.

VI. Fault Recovery

A synchronization failure is said to occur whenever a media unit is not available for presentation at the expected time specified by the scheduler. In such a case the demanded quality of service with respect to synchronization requirements can not be met; the synchronization system must provide some method of exception handling.

Naik addresses this specific issue in his paper. He introduces the notion of detected virtual faults and introduces schemes to make faults nonpersistent. Fault recovery is to our point of view an issue handled by the mechanisms, whereas a possible choice in terms of behavior can be specified as part of the model.

VII. Implementation

In this last section, papers with implementation-related issues are presented. In general, these papers give overall approaches which include models, mechanisms and a discussion of their performance or related experience. The described systems grow from enhancements of communication systems toward complete approaches, incorporating application experiences.

Yang and Huang start with a description of their Petri Net based model addressed as a real-time synchronization model. This model is applied to the design of a multimedia synchronization protocol which comprises the mechanisms for performing synchronization. The respective architecture and protocol data units are described. The paper concludes with a set of performance measurements of this protocol.

Hui *et al.* propose to make use of large buffers and feedback as a mechanism for achieving synchronization. This flow control mechanism is based on three buffer thresholds. A performance analysis is provided to show the adaptiveness and efficiency of this approach known as the GRAMS server and client.

Chen and Wu develop a mechanism for continuous media playback. It is based on multiple threads, which concurrently perform the required synchronization. The implementation is performed in a local UNIX environment. Finally, some performance measurements are provided.

Blair *et al.* present their object-oriented model for programming distributed multimedia applications. It is based on the synchronous imperative parallel language Esterel with reactive objects and quality of service controlled bindings. The authors show how this model is applied to achieve intra-stream synchronization, lip synchronization, and synchronization in a multimedia presentation. The respective mechanisms are implemented as extensions to the Chorus micro-kernel.

Lamont *et al.* make use of their Time Flow Graph and Stream Synchronization Protocol for synchronization of multimedia data. With absence of a global clock, the presented stream synchronization protocol allows for synchronization recovery due to the presence of random network delay and jitter. This approach also takes into account MPEG-2 coded streams. The whole system is implemented as part of a multimedia news-on-demand application. Finally, results of performance modeling of the synchronization protocol, using deterministic and stochastic Petri Nets, are discussed in detail.

VIII. Conclusion

In conclusion, the editors believe that, given that until today much work on multimedia synchronization has been carried out and presented at many different publications and events, this comprehensive JSAC issue on the topic will help the overall community to more easily make use of widely spread results and will also stimulate further research.

Acknowledgment

The editors would like to thank all authors for their contributions and patience with the rounds of changes required as suggested by reviewers and editors. Unfortunately, they also had to reject a large amount of interesting contributions due to space limitations. They are also grateful to the reviewers, who were very astute and provided them and the authors with many detailed comments.

References

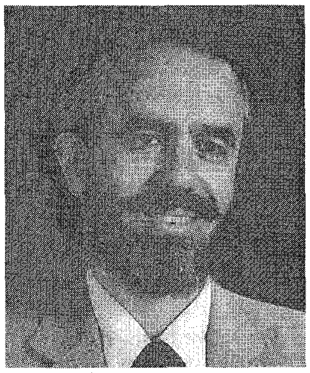
[1] L. Li, A. Karmouch, and N. D. Georganas, "Multimedia teleorchestra with independent sources: Part 1—Temporal modeling of collaborative multimedia scenarios," *ACM/Springer Multimedia Syst.*, vol. 1, no. 4, pp. 143–153, Feb. 1994.

[2] ——, "Multimedia teleorchestra with independent sources: Part 2—Synchronization algorithms," *ACM/Springer Multimedia Syst.*, vol. 1, no. 4, pp. 154–165, Feb. 1994.

[3] R. Steinmetz and K. Nahrstedt, *Multimedia—Computing, Communications and Applications.* New York: Prentice Hall International, 1995, p. 853.

Nicolas D. Georganas, *Guest Editor*
Ralf Steinmetz, *Guest Editor*
Toru Nakagawa, *Guest Editor*

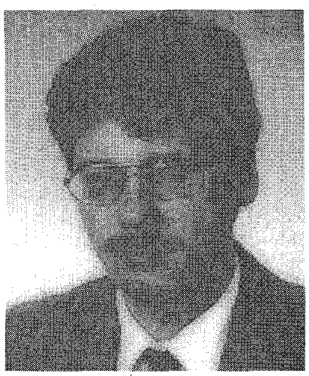
J. F. Hayes, *J-SAC Board Representative*



Nicolas D. Georganas (S'68–M'70–SM'78–F'90) received the Dipl.Ing. degree in electrical and mechanical engineering from the National Technical University of Athens, Greece, in 1966, and the Ph.D. degree in electrical engineering (summa cum laude) from the University of Ottawa, in 1970.

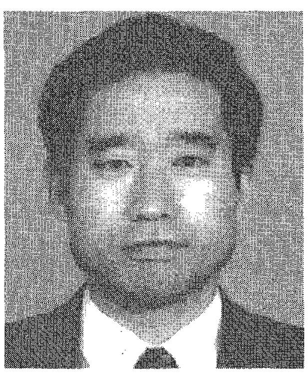
He is Professor of Electrical Engineering, University of Ottawa. He has been a faculty member in the Department of Electrical Engineering, University of Ottawa, and served as Chairman from 1981 to 1984. In 1986, he was appointed Founding Dean of the separate Faculty of Engineering and served a seven year term. He has published over 200 technical papers and is co-author of the book *Queueing Networks—Exact Computational Algorithms: A Unified Theory by Decomposition and Aggregation* (Cambridge, MA: MIT Press, 1989). He has served as Guest Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, issue on "Multimedia Communications" (April 1990), and as Technical Program Chair of IEEE MULTIMEDIA '89, the 2nd IEEE COMSOC International Multimedia Communications Workshop (Montebello, Canada, April 1989), and of the ICCC Multimedia Communications '93 Conference in Banff, Alberta, Canada. He is on the Editorial Boards of the Journals *Performance Evaluation, Computer Networks and ISDN Systems, Computer Communications, Multimedia Tools and Applications and the IEEE Multimedia Magazine.* His current research interests are multimedia communication systems and performance evaluation of multimedia broadband networks.

Dr. Georganas was elected Fellow of IEEE, for "leadership in university-industry research in, and performance evaluation of, multimedia communication networks and systems." He is also a Fellow of the Engineering Institute of Canada and a Governor of ICCC.



Ralf Steinmetz (S'83–M'86–SM'93) received the M.Sc. (Dipl.-Ing.) degree and the Ph.D. (Dr.-Ing.) degree, working in the area of Petri nets and concurrent programming languages with focus on communications, from the Technical University of Darmstadt, Germany, both in electrical engineering, in 1982 and 1986, respectively.

He led the successful IEEE Multimedia Task Force Working Group for magazine publication and is now the associate Editor-in-Chief of *IEEE Multimedia.* He has served as editor and member of the editorial advisory board of several journals and as chair, vice chair, and member of numerous program and steering committees for workshops and conferences. He manages the multimedia department at the IBM European Networking Center, Heidelberg, Germany. There, in 1988, he initiated the first activities on networked multimedia issues. Since then he has been in charge of several multimedia projects. He has worked in academia, supervising numerous theses and lecturing. He wrote the first in-depth technical book on multimedia systems published in German in 1993; a co-authored and restructured edition in English published in 1995. Before joining IBM, he worked in ISDN development activities for Philips.



Toru Nakagawa was born in 1950 in Tokyo. He received the B.E. and M.E. degrees in electrical engineering from Waseda University, Tokyo, in 1973 and 1975, respectively.

He manages the multimedia research area in the Laboratory for Information Technology, since joining NTT Data Communications Systems Co. in 1994. Since joining NTT Laboratories in 1975, he has been active in the development of a facsimile intelligent communication software system which is now well known under the service name "F-Net" in NTT. After that, he was engaged in developmental research on visual information response and communication processing systems. That system is sold by the name of "C&DS," which is the G4 and MHS protocol-based database systems. Then, he was Research Group Leader for multimedia systems in Human and Multimedia Laboratories in NTT Human Interface Laboratories. He has interest in multimedia information systems constructed from represent information processing and natural language processing.

Mr. Nakagawa is a member of the Institute of Electronics, Information and Communication Engineers of Japan.