TLSA PLAYER: A TOOL FOR PRESENTING CONSISTENT SMIL 2.0 DOCUMENTS

Paulo N. M. Sampaio
Distributed Systems and Networks Lab. (Lab-SDR), University of Madeira (UMa)
Campus da Penteada 9000-390, Funchal, Madeira, Portugal

Jean-Pierre Courtiat
LAAS – CNRS, 7, Av. du Colonel Roche
31077 Toulouse – France

Keywords: Formal specification, RT-LOTOS, Temporal consistency, Interactive Multimedia Documents, SMIL 2.0, Multimedia Player.

Abstract: Describing synchronization constraints in complex Interactive Multimedia Documents at authoring time can be an error-prone task, especially if the increasing number of media objects participating in these relations is considered. As a consequence, some synchronization constraints specified by the author may not be satisfied, leading the presentation of the document to undesirable deadlocks or to unexpected misbehaviours, characterizing the occurrence of an inconsistency. In particular, the flexibility of high level authoring models (such as SMIL 2.0) for the edition of complex IMDs can lead authors, in certain cases, to specify inconsistent documents. For this reason, it is important to apply multimedia players that can ensure the presentation of consistent documents. This paper presents the main aspects of the development of a multimedia player which ensures the presentation of consistent multimedia documents, the TLSA Player. The TLSA Player is part of a formal methodology for the design of multimedia documents which provides the formal semantics for the dynamic behaviour of the document, consistency checking, and the scheduling of the presentation taking into account the temporal non-determinism of these documents.

1 INTRODUCTION

The definition of Interactive Multimedia Documents (IMDs) is related to the coordinated presentation of different types of information (text, images, audio, video, etc.) possibly associated with user interactions. An IMD is considered as consistent when all of its synchronization constraints specified by the author can be respected during its presentation. Unfortunately, little has been done concerning the verification of consistency properties and scheduling of IMDs (Courtiat & Oliveira, 1996; Layaida & Keramane, 1995; Mirbel et al, 2000; Jourdan, 2001).

This paper presents the main characteristics of a multimedia player, the TLSA Player which was developed as a part of a formal methodology for the design, consistency analysis, scheduling and presentation of IMDs. In order to illustrate the utilization of this methodology, SMIL 2.0 was applied to the edition of IMDs (Sampaio, 2003). This methodology presents a solid formal basis, the Formal Description Technique Real Time LOTOS (and its simulation/verification environment RTL – RT-LOTOS Laboratory which was developed at LAAS-CNRS (Courtiat, 2000)). This methodology is illustrated in Figure 1.

This methodology covers all the life cycle of multimedia documents: (1) edition of the IMD using SMIL 2.0 as the high-level authoring model; (2) automatic translation of logical and temporal structure of the document into an RT-LOTOS formal specification (the non-temporal components of the document are translated automatically into a

N. M. Sampaio P. and Courtiat J. (2007).
TLSA PLAYER: A TOOL FOR PRESENTING CONSISTENT SMIL 2.0 DOCUMENTS.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - HCI*, pages 268-271
DOI: 10.5220/0002357502680271

Contextual Information File which is used as a support for its presentation); (3) derivation of the minimal reachability graph from the RT-LOTOS specification using the RTL tool; (4) verification of consistency properties by means of reachability analysis; generation of a scheduling graph based on the results of verification (5) and (6), and; further presentation of the document using an appropriate multimedia player (7).

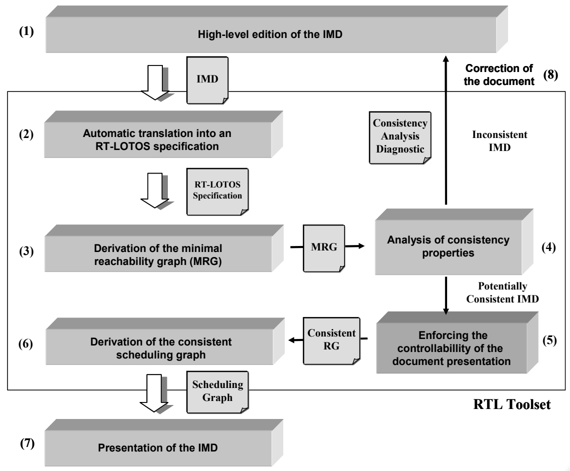

Figure 1: Formal design methodology of SMIL 2.0 documents.

This paper is structured as follows: Section 2 presents a global view of the verification and scheduling of IMDs. Section 3 presents the main aspects of the development of the TLSA Player. Finally, section 4 presents some conclusions of this work.

2 VERIFICATION AND SCHEDULING OF IMDS

The verification technique based on RT-LOTOS was initially applied to support the design of IMDs in (Courtiat & Oliveira, 1996). Relying on the RT-LOTOS specification, different graphs can be generated automatically by the RTL toolset, such as the reachability graph. The reachability graph is a labelled transition system that describes all the possible behaviours of the document where the actions associated with each transition of this graph correspond either to the beginning or to the end of the presentation of media objects, to user interactions, to navigation control actions (hyperlink navigation) and to the progression of time (t).

The basic idea of verifying the temporal consistency in a minimal reachability graph is to analyze the reachability of the action characterizing the end of presentation of the document (action *end*). Thus, considering that the reachability graph describes all the possible behaviours of the document, there are some paths that starting from the initial node of the reachability graph lead to the occurrence of the action *end*. These paths are called *consistent paths*. Contrarily, there can be some other paths that never will lead to the occurrence of the action *end*. These paths are called *inconsistent paths*. If there is at least one consistent path on the reachability graph, the associated document can be presented.

The scheduling of an IMD is carried out based on the results of the verification, and a scheduling graph is derived from the reachability graph. The scheduling graph is so-called *Time Labelled Scheduling Automaton (TLSA)* (Lohr, 2002). The scheduling and presentation of the IMD is further carried out by a particular multimedia player, the TLSA Player.

3 TLSA PLAYER

The TLSA Player relies on the TLSA and on the *Contextual Information* for supporting the presentation of complex and consistent IMDs. The TLSA Player was implemented using JAVA (jdk 1.2) and JMF 2.0. Figure 2 illustrates a snapshot of the TLSA Player.

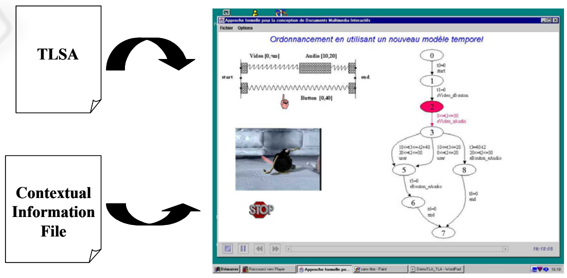

Figure 2: The TLSA Player.

In particular, the *Contextual Information File (CIF)* is automatically derived from a SMIL 2.0 document in order to be used as support for the presentation of the document by the TLSA Player. The CIF describes all the information of an IMD related to the content to be presented (e.g., URL), presentation characteristics (e.g., layout positioning, volume, etc.), transition effects, animation, etc.

The implementation of the TLSA Player was carried out within two different phases which represent the development of the functionalities supported by the player. The first phase represents the implementation of the player based on the characteristics of SMIL 1.0. This version supports: (1) scheduling of single (e.g. sVideo) and composed actions (e.g. eInteractiveButton_eVideo, denoting that both actions occur simultaneously), (2) the control of presentation through the buttons play, pause and stop, (3) the presentation of a list of timers associated with each state of the TLSA, (4) the implementation of an optimized algorithm for prefetching of media objects during the presentation, and (5) the presentation of remote media based on the http, ftp and rtp protocols. Several tests with satisfactory results were carried out based on the presentation of local and remote media objects.

The second phase represents the implementation of the player based on the characteristics of SMIL 2.0. The functionalities implemented and enhanced on the player are: (1) the scheduling of semantic actions on the TLSA, such as lastest_<>, earliest_<> and excl_<>, which are applied during the translation from a SMIL 2.0 document into an RT-LOTOS specification in order to avoid the state space explosion on the respective reachability graph; (2) management and presentation according to the user profile, (3) the management of events during presentation, and (4) the management of exclusive presentation.

3.1 The Architecture of the TLSA Player

The components of the architecture of the TLSA Player were defined in order to implement an operational scheduling procedure based on the TLSA. Figure 3 depicts a global view of the architecture for the implementation of this player.

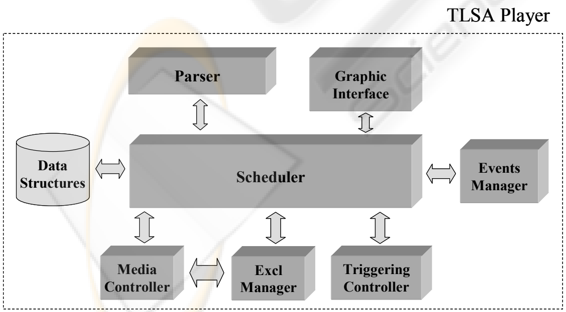

Figure 3: The architecture of the TLSA Player.

The *Scheduler* is considered to play a main role in this architecture since it is responsible to start the reading and the analysis of the TLSA and the *Contextual Information*, to start the presentation of the document, and to manage this presentation until it is finished. Consider Figure 4 which details the *Scheduler* and its relations with all the other components. Basically, the operation of the TLSA Player can be described according to the relations between the components of its architecture, as follows:

(1) When the TLSA Player is executed, the component *MainReader* starts to read and to analyze the *Contextual Information File* and the TLSA;

(2) During this analysis, the information concerning the TLSA and the *Contextual Information File* are recorded in the associated *Data Structures*;

(3) In order to begin the presentation, the *Graphic Interface* is updated;

(4) Also, the player prepares the preloading (*Prefetching*) of the media objects, which can be carried out completely under a static mode or gradually under a dynamic mode;

(5) When the *MainReader* component handles a state of the TLSA, it triggers a timer (*Active Timer*) associated with this state;

(6) The control of all the transitions of a state is managed by the component *Thread Controller*;

(7) *Thread Controller* creates a thread for each valid transition leaving the current state;

(8) The triggering condition associated with each transition from the current state (each thread created) is evaluated by *Triggering Controller*. As soon as a triggering condition associated with a thread is validated, the associated transition is triggered. It is interesting to note that when an action related to a user interaction or to the end of presentation of continuous media takes place (since these events are non-deterministic), its occurrence is announced to the component *ArcFiring* so that the triggering condition of the associated transition can be evaluated;

(9) At this time, *Thread Controller* determines an action (associated with the beginning or the end of presentation) to carry out in order to manage the presentation of the document. This action corresponds either to the presentation of a single media object (*Media Controller*), or to the exclusive presentation of some media objects (*Excl Manager*);

(10) If the action to be executed corresponds to the beginning or the end of an exclusive presentation, the component *Excl Manager* will create a thread associated with an exclusive presentation in particular. *Excl Manager* determines the instants of beginning and end of presentation of

the components of the exclusive presentation and announces them directly to *Media Controller*;

(11) When an event (such as the beginning or the end of presentation of a media object, a user interaction, or an elapsed timer) occurs within the TLSA Player, this event is also handled by *MainReader*. Consequently, an action can be carried out corresponding to the beginning of presentation of a media object, to the end of presentation of a media object, or to a timer triggering;

(12) The execution cycle from (5) to (11) is repeated until the transition associated with the action describing the end of the document presentation (*end*) is carried out.

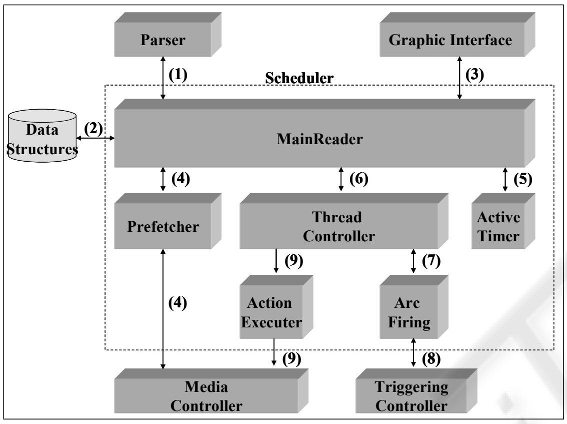

Figure 4: Architecture of the TLSA Player.

Several SMIL players have been developed such as GriNS (GriNS), RealPlayer (RealPlayer), etc. Unfortunately, these players are not based on a formal model that allows the description of the correct semantics of SMIL documents, and the verification of consistency properties.

4 CONCLUSIONS

The TLSA Player was developed as a component of a formal methodology for the design of Interactive Multimedia Documents (IMDs) in order to provide consistent presentations. The main advantages of this methodology are (i) that it is not dependent upon a particular high level authoring model and (ii) that the formal description technique RT-LOTOS is used implicitly for the author of the document (what makes the approach more accessible and intuitive). The contributions of the TLSA Player for the presentation of consistent SMIL 2.0 documents are relevant since it is based on the verification and scheduling techniques using the reachability graph and TLSA. This tool represents a straightforward solution in order to ensure the presentation of consistent complex IMDs and to validate the RT-LOTOS based formal methodology for the design of IMDs.

REFERENCES

Courtiat, J.-P., Oliveira, R.C. 1996. *Proving Temporal Consistency in a New Multimedia Synchronization Model*. In: 4th ACM Multimedia'96, pp.141-152, Boston, USA.

Courtiat, J.P., Santos, C.A.S., Lohr, C., Outtaj, B. 2000. *Experience with RT-LOTOS, a temporal extension of the LOTOS formal description technique*. In: Computer Communications 23.

GriNS Home Page. URL: http://www.oratrix.com/GRiNS/SMIL2.0/.

Jourdan, M. 2001. *A formal semantics of SMIL: a Web standard to describe multimedia documents*. In: Computer Standards and Interfaces, 23 (2001), pp. 439-455.

Layaida, N., Keramane, C. 1995. *Mantaining Temporal Consistency of Multimedia Documents*. In: ACM Workshop on Effective Abstractions in Multimedia, San Francisco.

Lohr, C. 2002. *Contribution à la conception de systèmes temps-réel s'appuyant sur la technique de description formelle RT-LOTOS*. In : PhD Thesis (in french), Université Paul Sabatier, Toulouse.

Mirbel, I., Pernici, B., Sellis, T., Tserkezoglou, S., Vazirgiannis, M. 2000. *Checking Temporal Integrity of Interactive Multimedia Documents*. In: VLDB journal.

RealPlayer G2 Home Page. URL: http://www.real.com/.

Sampaio, P.N.M. 2003. *Conception formelle de documents multimédia interactifs: Une approche s'appuyant sur RT-LOTOS*. In: Ph.D. Thesis (in french). Université Paul Sabatier – Toulouse III, Toulouse, France.

Synchronized Multimedia Integrated Language (SMIL). URL: http://www.w3.org/AudioVideo/