# Formal Design of Interactive Multimedia Documents

Jean-Pierre Courtiat

LAAS – CNRS
7 Av. du Colonel Roche
31077 Toulouse Cedex 4 – France
courtiat@laas.fr

**Abstract.** The expressiveness power and flexibility of high-level authoring models (such as W3C SMIL 2.0 language) used for editing complex interactive multimedia documents can lead authors to specify inter-media synchronization relations that cannot be met at document presentation time, leading often the document presentation to a temporal deadlock, called a temporal inconsistency. To avoid these undesirable behaviors, a document formal design method is presented in this paper for identifying and then possibly correcting temporal inconsistencies. It is shown how the results of the verification phase can be used by a scheduling automaton for avoiding undesirable temporal inconsistencies at presentation time, as well as for generating a new document temporal model free from temporal inconsistency. The proposed method is instantiated for the SMIL 2.0 authoring model. The automatic translation of a SMIL 2.0 document into a RT-LOTOS specification is addressed. The formal verification of the RT-LOTOS specification is performed using RTL (the RT-LOTOS Laboratory) developed at LAAS-CNRS. The scheduling of the document presentation, taking into account all the document temporal non-determinism, is performed through the use of a new type of time automaton, called a TLSA (Time Labeled Scheduling Automaton). Potential state space explosion problems are addressed at the level of the translation between a SMIL 2.0 document and a RT-LOTOS specification. Simple examples illustrate the main concepts of the paper and the results achieved.

## 1 Introduction

Designing interactive multimedia documents addresses the issue of managing the coordinated presentation of different types of information (text, images, audio, video, *etc.*) possibly associated with user interactions. The expressiveness power and flexibility of high-level authoring models (such as W3C SMIL 2.0 language) used for editing complex interactive multimedia documents can easily lead authors to specify inter-media synchronization relations that cannot be met at document presentation time, leading often the document presentation to a temporal deadlock, called a temporal inconsistency. To avoid these undesirable behaviors, this paper presents a document formal design method for identifying and possibly correcting temporal inconsistencies. It is shown how the results of the verification phase can be used by a scheduling automaton for avoiding undesirable temporal inconsistencies at presentation time, as well as for generating a new document temporal model free from temporal inconsistency. The proposed method is instantiated for the SMIL 2.0 authoring model.

The paper is organized as follows. Section 2 introduces the document formal design method. Section 3 explains how the temporal consistency analysis of interactive multimedia documents can be achieved relying on standard reachability analysis of RT-LOTOS specifications. Section 4 presents the techniques used for scheduling an interactive multimedia document based on a specific scheduling automaton synthesized from the reachability graph. Section 5 addresses the application of the formal design method to SMIL 2.0 interactive multimedia documents. Section 6 presents some conclusions.

## 2 The Formal Design Method

The proposed method provides a framework for the design (specification, verification and presentation) of complex Interactive Multimedia Documents (IMD). Relying on the formal description technique RT-LOTOS, and its associated simulation/ verification environment RTL, it yields three main contributions: it permits to define a formal semantics of the high-level document authoring model, describing without any ambiguity the behavior of the document presentation; it permits to check consistency properties on the RT-LOTOS formal specification derived from the authoring model, using standard verification techniques (reachability analysis and model checking); it permits to generate automatically a scheduling automaton from the underlying reachability graph, providing an easy way to automatically correct the document temporal structure by removing undesirable temporal inconsistencies.

### 2.1 Main Steps of the Formal Design Method

Main steps of the formal design method are presented in Figure 1.

First step: an interactive multimedia document is edited using a high-level authoring model. The proposed design method is not specific to one modeling language. It applies to any authoring technique, which enables translation of document models into RT-LOTOS specifications. So far RT-LOTOS translation algorithms have been defined for an ad-hoc IMD authoring language easily translatable into RT-LOTOS [1], general-purpose authoring models like NCM [2,3], SMIL 1.0, and SMIL 2.0 [4].

Second step: the document's temporal structure expressed in the selected authoring model is translated into a RT-LOTOS specification. The purpose is to define a general and "automatic" translation procedure between the high-level authoring model and RT-LOTOS, in such a way a document author does not need to learn the RT-LOTOS language.

Third step: a minimal reachability graph is derived from the RT-LOTOS specification obtained at Step 2. Using reachability analysis, it is then possible to perform a consistency analysis of the document. If all the temporal constraints of the document

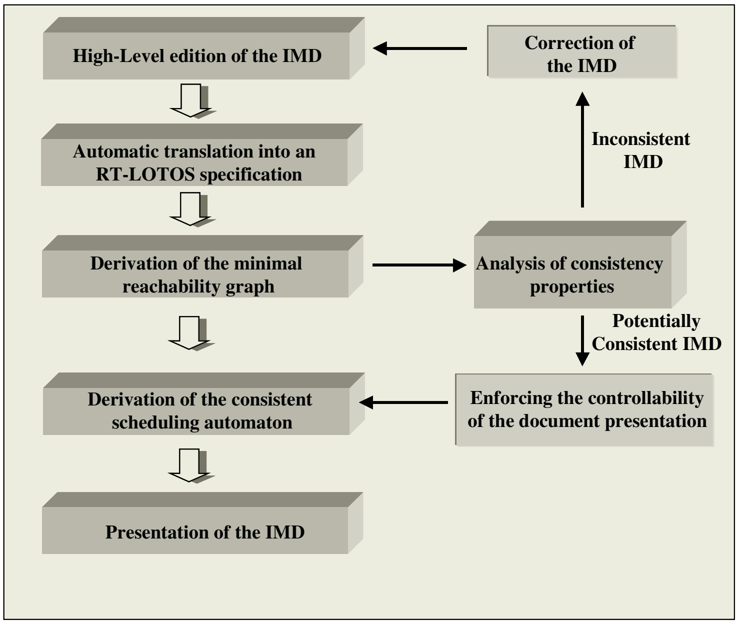

High-Level edition of the IMD
Automatic translation into an RT-LOTOS specification
Derivation of the minimal reachability graph
Derivation of the consistent scheduling automaton
Presentation of the IMD
Correction of the IMD
Inconsistent IMD
Analysis of consistency properties
Potentially Consistent IMD
Enforcing the controllability of the document presentation

**Fig. 1.** Main steps of the formal design method

can be met at presentation time (*i.e.* when the document is consistent), then the presentation scheduling may be performed based on a scheduling automaton directly derived from the reachability graph. This scheduling automaton is simple and provides means to ensure the controllability of the document presentation (controllability is required when the document is only *potentially* consistent, as defined in paragraph 3.2). That means that the document presentation is scheduled while enforcing some time constraints at document presentation time to avoid undesirable inconsistencies. On the opposite, if the document is inconsistent, its high-level description must be revisited and there is a feedback to the initial authoring step.

## 2.2 RT-LOTOS

RT-LOTOS [5] is a temporal extension of the LOTOS (Language of Temporal Ordering Specification) Formal Description Technique. It is very similar to the ET-LOTOS [16] temporal extension of LOTOS, with the exception of the way time non-determinism is handled. RT-LOTOS defines an operator for expressing time non-determinism: the latency operator. More detailed information on RT-LOTOS and its relationships with other temporal extensions of LOTOS can be found in [5].

The RT-LOTOS FDT is supported by a toolset developed at LAAS-CNRS and called RTL (RT-LOTOS Laboratory) [5]. Main functionalities of RTL are presented in Figure 2. Detailed information on how to use RTL is available at http://www.laas.fr/RT-LOTOS.

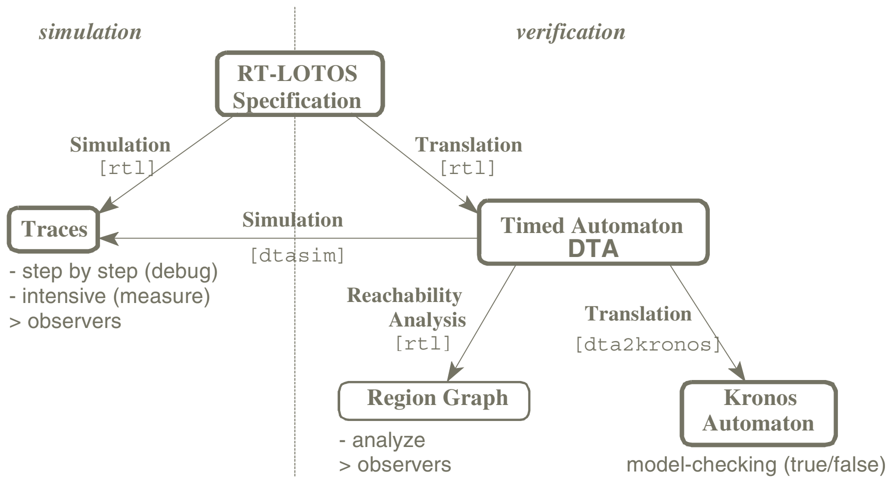


**Fig. 2.** The RTL toolset

Using RTL, a timed automaton, called a DTA [5], is derived from a RT-LOTOS specification, on which a reachability analysis is performed leading to a region graph, called a minimal reachability graph.

Each node of the reachability graph represents a reachable region, which can include an infinite number of configurations according to the current value of time. These configurations are gathered in a same region when they meet the same reachability properties in terms of future and past actions [6]. Each arc of the graph corresponds either to the occurrence of an RT-LOTOS action, or a global progression of time (an arc labeled by the action time or simply t). Reachability analysis can also be carried out on the fly while the DTA is generated. Different tools are used for the derivation and visualization of the minimal reachability graph, such as ALDEBARAN (which outputs an equivalent reachability graph according to various equivalence relations such as the observational equivalence), DOTTY, ATG and BCG [7].

## 2.3 Illustration

To illustrate some characteristics of the design method we will adopt the simple interactive multimedia document presented above. This scenario has been edited with the SMIL 2.0 authoring language and we assume for the moment that a RT-LOTOS specification has been generated for this scenario.

The scenario describes the parallel presentation of a composition of an interactive image (*interactiveButton*) with the sequential composition of an *exclusive* presentation of two audio objects (*audio1* and *audio2*) followed by a video object (*video*). The presentation duration of objects *audio1*, *audio2*, *video* and *interactiveButton* are respectively [0,5] seconds, [0,5] seconds, indefinite[1] and [0,20] seconds, and the following synchronization constraints are considered: (i) the presentations of *audio1*, *audio2* and *interactiveButton* must start simultaneously, (ii) the presentations of *video* and *interactiveButton* must end simultaneously, (iii) if an user interaction occurs during the presentation, the scenario is interrupted and then immediately restarted.

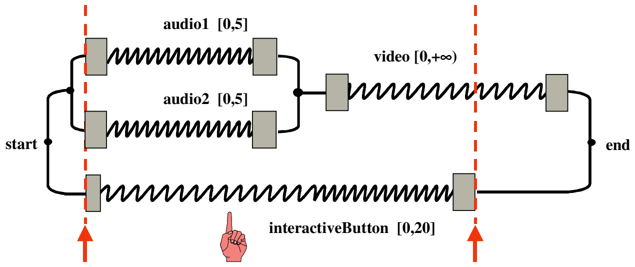


**Fig. 3.** Illustration of an interactive multimedia scenario

## 3 Verification of the Temporal Consistency Properties of an Interactive Multimedia Document

Depending on how document synchronization constraints are defined at authoring time, there exists a risk to create *inconsistent* scenarios, *i.e.* situations where different contradictory synchronization requirements cannot be satisfied together at presentation time, leading to undesirable deadlocks during the document presentation.

Consistency analysis of an interactive multimedia document consists in performing a reachability analysis of the underlying RT-LOTOS specification, consistency properties being expressed through reachability properties defined on this graph. A reachability graph is a *labeled transition system* where actions correspond to the *begin* or *end* of the presentation of media objects, to *user* interactions, to an action that represents the triggering of a hypermedia link (*triggerLink*) and to the progression of time.

### 3.1 Illustration of Temporal Inconsistencies

Many reasons may lead to temporal inconsistencies when an author uses a high-level and flexible authoring model. Examples are for instance: inconsistencies between the expected duration of the nodes and the logic of the synchronization links enforcing their termination; conflicting synchronization links; bad timing of the links, etc.

Let us consider for instance the simple scenario of Figure 4. It describes the presentation of three pictures *P1*, *P2* and *P3* (each one with a fixed duration of 5s) followed by the presentation of a text (*T*) (no timing constraint is associated with the

[1] An indefinite duration characterizes the presentation of an object inside the interval [0,+∞[

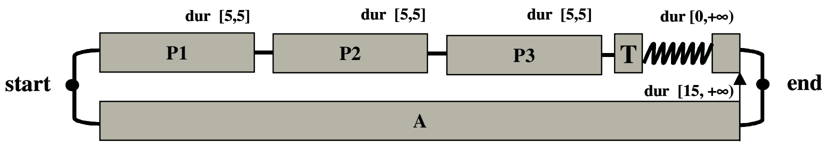


**Fig. 4.** Illustration of a temporally consistent scenario

presentation of T). The start of the presentation of P1 activates an audio object (*A*). Since A has a duration greater than 15s, the scenario is consistent and the end of A determines the end of *T*.

Let us assume now that the author changes his scenario by providing some user interactivity, permitting the user to stop the audio presentation at any time (Figure 5). It is then obvious that the scenario may present a temporal inconsistency depending on the time at which the user interaction occurs. If the presentation of A terminates before the start of the presentation of *T*, the latter cannot terminate, leading to a deadlock.

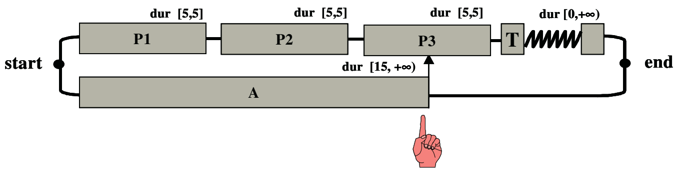


**Fig. 5.** Illustration of a temporally inconsistent scenario

## 3.2 Document Consistency Definitions

The basic idea about verifying the temporal consistency of a document is to analyze in the underlying minimal reachability graph the reachability of the action characterizing the end of the document presentation (action *end*).

*Definition 1*: a document is *potentially consistent* if there exists *at least one* path starting from the reachability graph initial node leading to an action characterizing the end of the document presentation (action *end*). Such a path is called a *consistent path.* A path, which is not consistent, is called an *inconsistent path.*

*Definition 2*: a document is *consistent* if the action characterizing the start of the document presentation (action *start* is by construction the only action enabled in the reachability graph initial node) is necessarily followed, some time later, by an action characterizing the end of the document presentation (action *end*). This means that *all the paths* starting from the reachability graph initial node are consistent.

Note that each consistent path of the reachability graph corresponds to a potential scheduling solution for the document presentation. The purpose of the document presentation scheduling, addressed in Section 4, is to ensure that the document presentation follows a consistent path, and that, in the presence of several consistent paths, it follows the best consistent path with respect to some criteria.

## 3.3 Characterization of Temporal Inconsistencies

The reachability graph expresses all the possible behaviors for the presentation of the document. Temporal inconsistencies are mostly the consequence of the occurrence of *internal* and/or *external non-deterministic events* [8].

*Internal non-deterministic events* are associated with the controlled adjustment of the presentation duration of a media; inconsistencies generated by internal non-deterministic events, *i.e.* the associated inconsistent paths in the reachability graph, may easily be removed at scheduling time since these events are controllable by the scheduler. The scheduler will thus be able to decide, without any interaction with the environment, at which time to start and end the presentation of each particular media; the document will be successfully presented as soon as it is potentially consistent.

*External non-deterministic events* are associated with the occurrence of external actions which are not directly controllable by the scheduler, such as user interactions; in this case the document could only be successfully presented if all the paths depending on external non-deterministic events are consistent; however, it is still possible to automatically correct the document presentation temporal structure, by removing from the reachability graph the associated inconsistent paths, as soon as it is possible to express refined time constraints for the occurrence of the external non-deterministic events. If such a correction may be performed, the document will be successfully presented as soon as it is potentially consistent.

As an illustration, let us consider the multimedia scenario presented in Figure 6. The associated reachability graph is illustrated in Figure 7, where the inconsistent paths are marked in bold. The occurrence of the internal non-deterministic events (see the time progression (t) between nodes 2-(30 30)->2-(30.5 30.5), 12-(0 0)->12-(30 30), 12-(19.5 30)->12-(20 30) and 7-(10 0)->7-(0.5 0.5)) and the external non-deterministic events (see the user interactions (*user*) between nodes 2-(30 30)->4-(30 0), 3-(0 30)->10-(0 0), 12-(9.5 30)->11-(0 0) and 12-(0 30)->11-(30 0)) can easily be checked. Thus the document is only potentially consistent and the scenario presentation will deadlock if the duration of objects *A*, *B* and *C* does not respect the synchronization constraints or if a user interaction occurs during the presentation of *A* or during the first 10 seconds of the presentation of *B*.

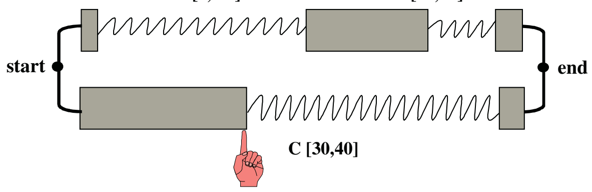


**Fig. 6.** A scenario with external and internal non-deterministic events

Enforcing the controllability of the scenario consists in removing all the inconsistent paths, generated either by internal or external non-deterministic events. Removing inconsistent paths generated by internal non-deterministic events is performed by the

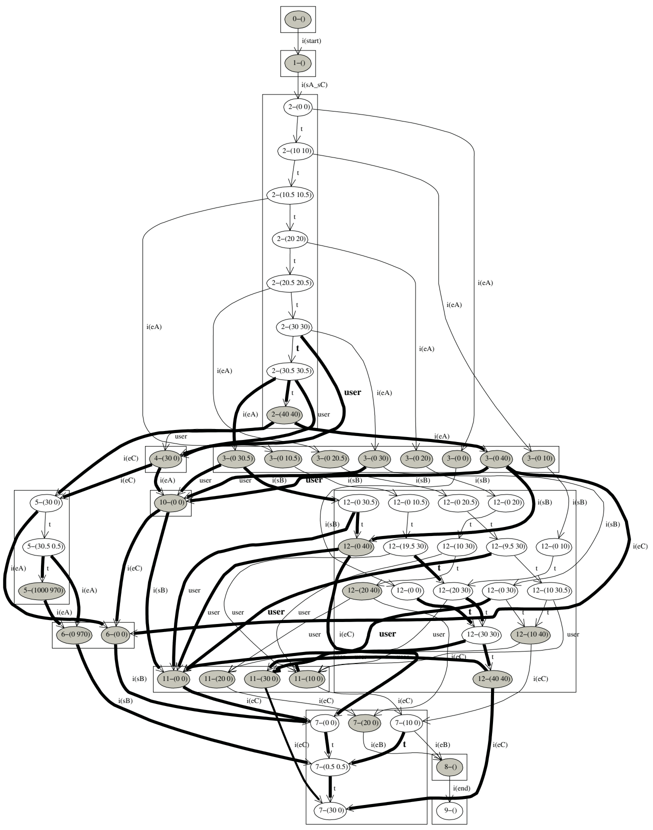


**Fig. 7.** Reachability graph with external and internal non-deterministic events

scheduler at presentation time without any modification to the document temporal structure; however, removing inconsistent paths generated by external non-deterministic events leads to a correction of the document temporal structure: new temporal windows, *i.e.* new time constraints are associated with the occurrence of the external events (for instance the occurrences of an user interaction); this correction mechanism has been completely automated and will be detailed in the next section.

## 4 Scheduling of an Interactive Multimedia Document

As previously discussed, a document may be either potentially consistent, or consistent or even inconsistent. If the document is *inconsistent*, that means that there is no consistent path in the reachability graph leading the document presentation to its normal end; as a consequence, the document cannot be presented and both its logical and temporal structures have to be revisited at the design level.

If the document is *consistent*, that means that all the paths in the reachability graph are consistent and each path represents a valid presentation solution; two points have then to be addressed:

- *Point 1*: how to represent in a more compact and operational way all the consistent paths; this point deals with the synthesis of a scheduling automaton (called a TLSA [9]) from the reachability graph; it will be addressed in paragraph 4.1
- *Point 2*: how to select among the consistent paths, the one which is the best with respect to some criteria; this point deals with the scheduling of an IMD based on a TLSA; it will be addressed in paragraph 4.2

If the document is *potentially consistent*, that means that there is at least one consistent path in the reachability graph. All the inconsistent paths have then to be removed from the reachability graph; this is automatically performed by removing the undesirable blocking nodes, as well as the paths starting from the reachability graph initial node and leading to these blocking nodes (see [11] for details). Once inconsistent paths have been removed, a new reachability graph is obtained which is by definition consistent (it was potentially consistent before removing the inconsistent paths). Points 1 and 2 have then to be addressed for this new reachability graph.

## 4.1 Generating a Scheduling Automaton

Timed automata were proposed to model real-time systems with finite states [10]. A timed automaton is characterized by a set of finite control states and a finite number of clocks. The evolution of all the clocks follows the same rate and measures the amount of time elapsed since they were initialized. Each transition of the system can reset certain clocks, and is described by an enabling condition expressed in the form of a constraint depending on the values of the clocks.

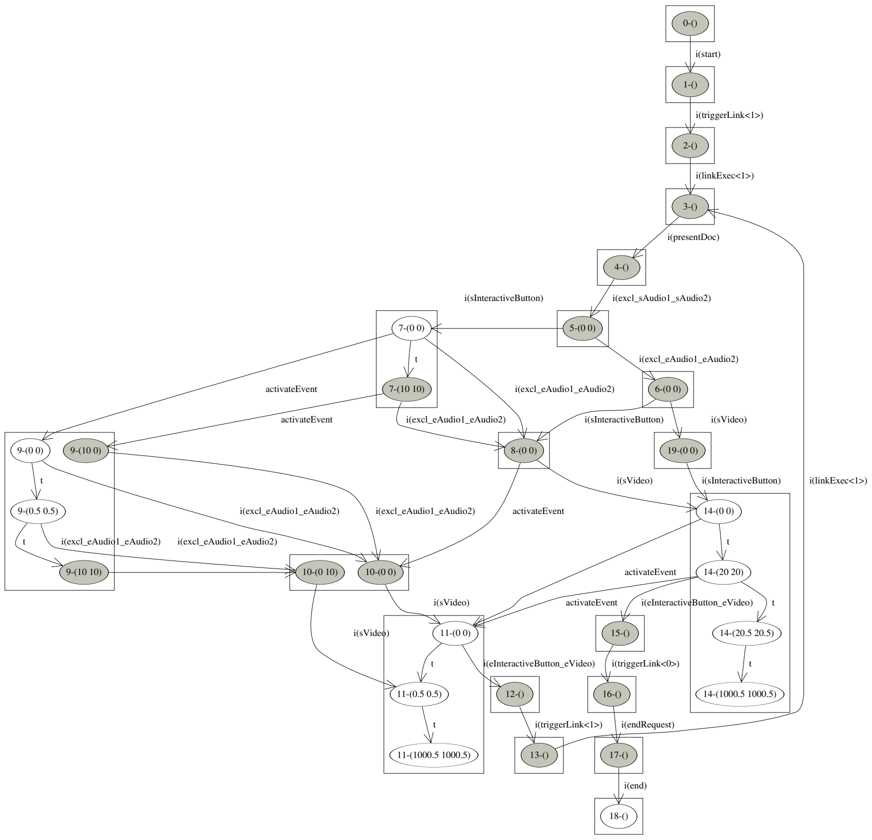
A TLSA features some specific characteristics, which differentiate it from traditional timed automata. A TLSA has as many clock as the number of states defined in the automaton. These clocks are called *timers* in order to distinguish them from the traditional clocks. Each timer measures the time spent by the automaton in the associated control state. No explicit function defines when a timer must be initialized. Indeed, the timer associated with a control state is reset when the automaton enters this state, and its current value is frozen when the automaton leaves it (for this reason, we say that a TLSA is a single clock model, since, for any control state, there is only one active timer). Two temporal conditions are associated with each TLSA transition:

- A *firing window* (indicated as *W*), which defines the temporal interval inside which the transition can be triggered. It is represented as an inequality to be satisfied by the timer associated with the current control state;
- An *enabling condition* (indicated as *K*) which defines the temporal constraints to be satisfied in order to trigger the transition. It is represented as a conjunction of inequalities to be satisfied by a subset of timers, with exception of the timer associated with the current control state.

Intuitively, *W* characterizes the time during which the system can remain in the current control state. The *enabling condition* is related only to previous timers and thus characterizes the enabled transitions according to the behavior of the system. The formal semantics of the TLSA was defined in [9], and the algorithms for deriving automatically a TLSA from a reachability graph were developed in [11].

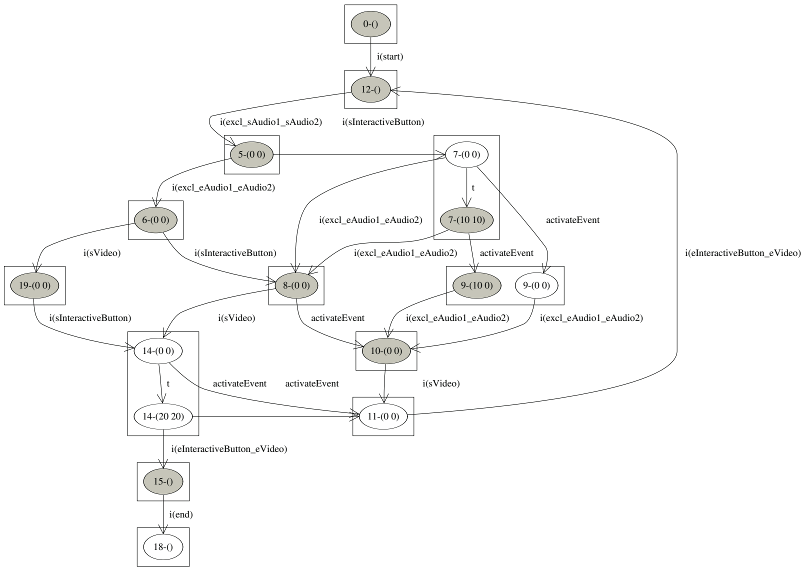
In summary, a TLSA is a single clock automaton (there is always only one running clock), which can be used for scheduling a document. However, some rules must still be defined to allow the orchestration of the document based on the occurrence of the actions described by the transitions of the TLSA. These rules are introduced in paragraph 4.2.

In order to illustrate the generation of a TLSA, let us consider our initial example of Figure 3. The reachability graph associated with this scenario is presented in Figure 8(a). This reachability graph describes all the consistent and inconsistent presentation behaviors of the document. The consistent reachability graph (Figure 8(b)) is obtained by eliminating all the undesirable blocking nodes (nodes 11-(1000.5 1000.5) and 14-(1000.5 1000.5)) as well as the paths leading to these blocking nodes. The TLSA derived from the consistent reachability graph is presented in Figure 9.



(a)

**Fig. 8.** Reachability graphs associated with the scenario of Figure 3



(b)

**Fig. 8.** (Cont.)

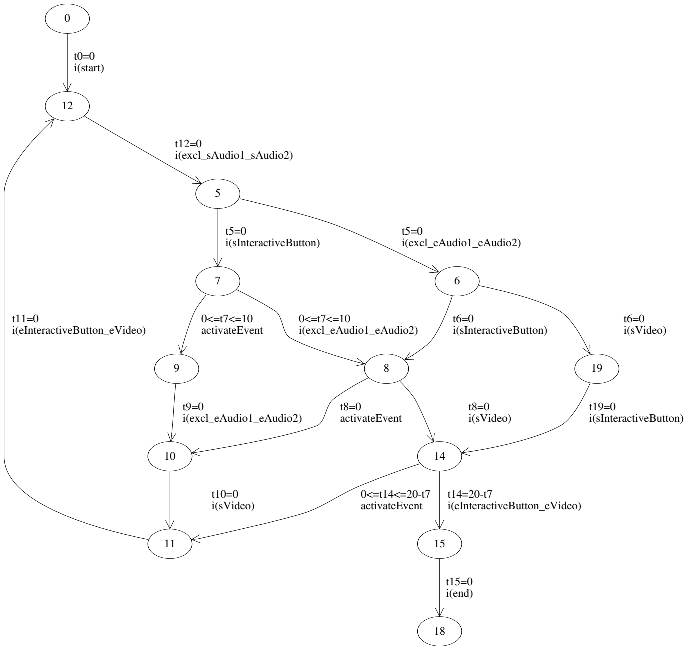


**Fig. 9.** TLSA derived from the consistent reachability graph (Figure 8(b))

In the TLSA of Figure 9, transitions 12-5 and 5-7 represent the start of the mutual exclusive presentation of *audio1* and *audio2*, followed by the start of presentation of *interactiveButton*. If there is no user interaction on *interactiveButton* within 10 seconds (an user interaction is represented here by action *activateEvent*), the end of the mutual exclusive presentation may occur followed immediately by the start of the presentation of *video* (transitions 7-8 and 8-14). Later, the presentation of *interactiveButton* terminates as well as the presentations of *video* and of the scenario (transitions 14-15 and 15-18). If however, a user interaction takes place during the presentation of the scenario (transitions 7-9, 8-10 and 14-11), the presentation is interrupted and then restarted (transition 11-12). Note that the presentation of *video* is synchronized to the end of *interactiveButton* (which occurs within [0,20] seconds - transition 14-15).

## 4.2 Scheduling an Interactive Multimedia Document Based on the TLSA

The TLSA (Time Labeled Scheduling Automaton) was proposed in [9] as a scheduling graph, which permits the temporal formatting (the definition of the valid temporal intervals for the presentation) of a document. The scheduling of the document can be carried out based on timers which measure the time the system "spent" in each state of the automaton and which is also used to determine the triggering condition of a transition. Moreover, each transition of the TLSA is associated with the actions that can be managed by the presentation system such as starting or ending a presentation, announcing the occurrence of a user interaction, etc.

The scheduling of an IMD using a TLSA can be accomplished based on the definition of some important issues, such as handling *active* and *passive* actions.

An action is said *active* if its occurrence does not depend on its environment (e.g., the end of presentation of an image decided by the system). Thus, *active* actions may be related to *start* and *end* actions of all media objects (e.g., video, audio, text, image, etc.). These actions can be generated as soon as the scheduler decides when to trigger them based on their associated temporal firing window (*W*) specified in the associated TLSA

An action is said *passive* if its occurrence does depend on the environment (e.g., an user interaction, or the end of the presentation of a continuous media). The occurrence of *passive* actions cannot be predicted, but can be controlled within their valid temporal firing window in the TLSA. Thus, the scheduler waits until the environment triggers these actions and ensures that they are triggered inside their temporal firing window.

The notion of *active* and *passive* actions enables to determine scheduling policies for the execution of all the actions for the presentation of a document; let thus *W* = *[min,max]* be the firing window associated with transition t, $\tau$ the firing time of transition t, and a the action labeling transition t.

- If *a* is active, then $\tau$ = min; if *a is* passive, then $\tau \in$ [min, max]

Note finally that the scheduler may force at $\tau$ = max the firing of a *passive* action or the firing of an exception action.

## 5 Application of the Design Method to SMIL 2.0 Documents

The *Synchronized Multimedia Integrated Language (SMIL)* [4] has been applied to the design and presentation of interactive multimedia documents distributed over the Web. One important feature of SMIL 2.0 is its flexible temporal model, which eventually can lead authors to describe synchronization constraints that cannot be resolved consistently at document presentation time (tools like *Grins*, *RealOne* behave differently in such circumstances). Efforts have been reported for defining a formal semantics of SMIL [13] but few work [15] has been reported on analyzing consistency properties of interactive multimedia documents [1,2,3,8,14] authored in SMIL.

The purpose of this section is to show how the formal design method proposed in the paper may be applied to SMIL 2.0 interactive multimedia documents.

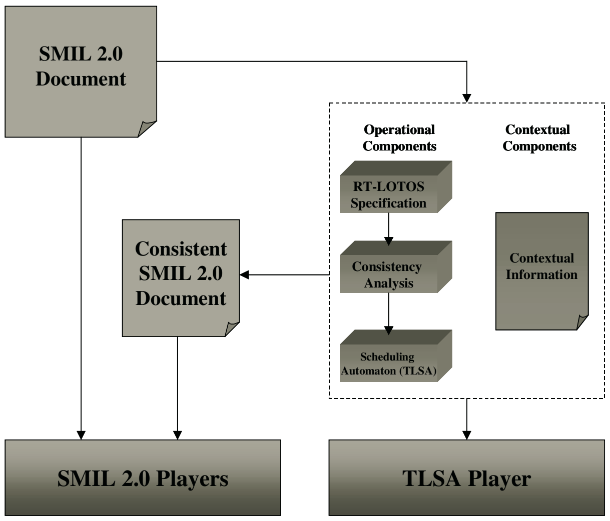


**Fig. 10.** Application of the formal design method to SMIL 2.0 documents

A SMIL 2.0 document may be decomposed into *operational* and *contextual* components. The operational components describe the logical and temporal structure of the document; they are translated into a RT-LOTOS specification. The contextual components describe essentially the spatial synchronization requirements as well as a description of the document content (type and location of the media); they are translated into a data structure called the *Contextual Information*. Consistency analysis is applied as previously on the RT-LOTOS specification derived from the operational components. Assuming that the document is potentially consistent, the document temporal structure may be corrected by generating a consistent scheduling automaton (TLSA).

Two approaches are possible for presenting the resulting consistent document. Either a new consistent SMIL 2.0 document is generated from the TLSA and the Contextual Information, and the presentation may then be performed using commercial tools like *Grins* or *RealOne*; or, the TLSA and the associated Contextual Information are directly presented using a dedicated tool, the *TLSA Player*, developed at LAAS-CNRS.

Two main issues related to the application of the design method to SMIL 2.0 documents are addressed in the sequel: (i) how to translate the operational components into a RT-LOTOS specification, (ii) how to minimize the potential state space explosion.

## 5.1 Translating the Operational Components into a RT-LOTOS Specification

The translation procedure is performed in two phases. In the first phase, two data structures, the *temporal tree* and the *synchronization arcs* are derived from the SMIL document. The temporal tree describes hierarchically the components of the document (media objects, containers like *par* (parallel composition) or *seq* (sequential composition) or *excl* (exclusive composition), anchors, …) and their temporal characteristics. The synchronization arcs describe the causal relations among the components; they are expressed by condition-action relations; a condition is related to the occurrence (possibly associated with a time constraint) of either the *begin* or the *end* of a component, or an *user* interaction: an action is related to the execution of the *begin* or the *end* of a component.

In the second phase, two additional data structures, called *ProcRTL* and the *synchronization points*, are derived; from these data structures, the RT-LOTOS processes are automatically generated. ProcRTL describes the structure of each RT-LOTOS process (process Id, behavior type referring to a library of predefined processes, observable gates, hidden gates, sub-processes, …) and the synchronization points define the synchronization relations to be applied when composing the RT-LOTOS processes. Details of the translation procedure can be found in [15].

## 5.2 Overcoming the State Space Explosion

If each operational component of a SMIL 2.0 document is described by an elementary RT-LOTOS process, the resulting specification corresponds to the parallel composition with synchronization of all these processes, leading potentially to a state space explosion of the associated reachability graph. To overcome this problem, three main techniques have been implemented in the RT-LOTOS translation procedure:

1. To replace single actions that are known to occur simultaneously by *composed actions*,
2. To transform interleaved actions into *serialized actions* by taking into account their associated temporal constraints, and
3. To define so-called *semantics actions* (such as latest_, earliest_, excl_) whose semantics is to be interpreted at presentation time, instead of being detailed in RT-LOTOS.

These techniques have permitted to reduce drastically the size of the derived reachability graph (for instance, from 63 nodes and 116 transitions to 30 nodes and 24 transitions for the scenario in Figure 3).

## 6 Conclusion

The formal design method presented in this paper is not dependent on a particular high-level authoring model, and different applications of this design method have been carried out. The underlying formal description technique, RT-LOTOS, remains hidden to the document author, which makes the approach more accessible and intuitive. One main advantage of this design method is that the author has not to worry about the temporal consistency of his document presentation. He can precisely specify the temporal constraints he wants to enforce leaving all the others temporal constraints unspecified (*i.e.* defined as $[0, \infty[$). This will normally lead to numerous temporal inconsistencies which can automatically be removed at the level of the reachability graph, leading then to a consistent scheduling automaton (TLSA), characterizing all the valid time constraints for both internal and external events. A potential shortcoming of the approach is related to the state space explosion, which is taken into account at the level of the RT-LOTOS translation procedure.

**Acknowledgments**

I would like to thank Roberto Cruz de Oliveira, Celso Alberto Saibel Santos, Paulo Nazareno Maia Sampaio and Christophe Lohr who have been actively involved in this project. Thanks are also due to Pierre de Saqui-Sannes for proofreading this paper.

## References

1. Courtiat, J.-P.; Oliveira, R.C. Proving temporal consistency in a new multimedia synchronization model. In Proc of ACM Multimedia'96, pp.141-152, Boston, USA, Nov. 1996
2. Santos, C.A.S.; Soares, L.F.G.; Souza, G.L.; Courtiat, J.-P. Design methodology and formal validation of hypermedia documents. In: Proc. of ACM Multimedia'98, Bristol, UK, Sep. 1998. pp.39-48
3. Santos, C.A.S.; Courtiat, J.-P.; Saqui-Sannes, P. A design methodology for the formal specification and verification of hypermedia documents. In Proc. of FORTE/PSTV'98, Paris, France, November, 1998. Chapman & Hall
4. http://www.w3.org/AudioVideo/
5. Courtiat, J.P.; Santos, C.A.S.; Lohr, C.; Outtaj, B. Experience with RT-LOTOS, a temporal extension of the LOTOS formal description technique. In Computer Communications 23, July 2000. http://www.laas.fr/RT-LOTOS
6. Yannakakis, M; Lee, D. An efficient algorithm for minimizing real-time transition systems. In Proc of CAV'93, Lecture Notes in Computer Science, Vol 697, Springer Verlag, 1993

7. http://www.inrialpes.fr/vasy/cadp.html
8. Santos, C.A.S.; Sampaio, P.N.M.; Courtiat, J.-P. Revisiting the concept of hypermedia document consistency. In Proc of ACM Multimedia'99, Orlando, USA, November 1999
9. Lohr, C.; Courtiat, J.P. From the specification to the scheduling of time-dependent systems. In Proc of 7th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems, Oldenburg, Germany, September 2002, Lecture Notes in Computer Science, Vol 2469, Springer Verlag, pp 129-145
10. Alur, R.; Dill, D.L. The theory of timed automata. In Proc of REX Workshop "Real-Time: Theory in Practice", Lecture Notes in Computer Science, Vol 600, Springer Verlag, 1991
11. Lohr, C. Contribution à la conception de systèmes temps-réel s'appuyant sur la technique de description formelle RT-LOTOS, PhD Dissertation (in french), Université Paul Sabatier, Toulouse, December 2002
12. Rutledge, L. SMIL 2.0 – XML for Web Multimedia. In IEEE Internet Computing, pp. 78-84, September-October 2001
13. Jourdan, M. A formal semantics of SMIL: a Web standard to describe multimedia documents. In Computer Standards and Interfaces, 23 (2001), pp. 439-455, 2001
14. Layaida, N.; Keramane, C. Mantaining Temporal Consistency of Multimedia Documents. In Proc of ACM Workshop on Effective Abstractions in Multimedia, San Francisco, 1995
15. Sampaio, P. Conception formelle de documents multimédia interactifs : une approche s'appuyant sur RT-LOTOS, PhD Dissertation (in french), Université Paul Sabatier, Toulouse, April 2003
16. Léonard, L ; Leduc, G. An introduction to ET-LOTOS for the description of time-sensitive systems. In Computer Networks and ISDN Systems, Vol 29, 1997, pp. 271-292