An Approach for the Automatic Generation of RT-LOTOS Specifications from SMIL 2.0 Documents

Paulo N.M. Sampaio[1], Jean-Pierre Courtiat[2]

Universidade da Madeira (UMA) [1]

Dep. de Matemática e Engenharias

Campus da Penteada 9000-390

Funchal, Madeira, Portugal

Tel: +351 291 705291

psampaio@uma.pt

LAAS – CNRS[2]

7, Av. du Colonel Roche

31077   Toulouse – France

Tel: +33 5.61.33.62.44

courtiat@laas.fr

Abstract — *The flexibility of high level authoring models (such as SMIL 2.0) for the edition of complex Interactive Multimedia Documents can lead authors, in certain cases, to specify synchronization relations which could not be satisfied during the presentation of the document, thus characterizing the occurrence of temporal inconsistencies. For this reason, we need to apply a methodology 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. This paper refers to a methodology for the formal design of Interactive Multimedia Documents based on the formal description technique RT-LOTOS. In particular, this paper presents an approach applied by our methodology for the automatic translation of SMIL 2.0 documents into RT-LOTOS specifications.*

Resumo — *A flexibilidade de modelos de autoria de alto nível (tais como o de SMIL 2.0) para a edição de Documentos Multimídia Interativos complexos pode levar os autores, em certos casos, a especificar relações de sincronização que não podem ser satisfeitas durante a apresentação do documento, caracterizando a ocorrência de inconsistências temporais. Por essa razão, é necessária a utilização de uma metodologia que ofereça uma semântica formal para o comportamento do documento, realize a verificação de consistência, o escalonamento e a apresentação levando em consideração o não-determinismo temporal desses documentos. Este artigo faz referência a uma metodologia para a concepção formal de Documentos Multimídia Interativos baseados na Técnica de Descrição Formal RT-LOTOS. Em particular, este artigo apresenta uma abordagem, utilizada pela nossa metodologia, para a tradução automática de documentos SMIL 2.0 em especificações RT-LOTOS.*

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

1. Introduction

The definition of Interactive Multimedia Documents is related to the coordinated presentation of different types of information (text, images, audio, video, etc.) possibly associated with user interactions. The quality of the presentation of these documents depends upon the global consistency of their temporal synchronization constraints. An Interactive Multimedia Document is considered as consistent when all of its synchronization constraints can be respected during the presentation. Otherwise, if a synchronization constraint can not be respected, the presentation of the document can be lead to a deadlock situation.

1

Unfortunately, few has been done concerning the verification of consistency properties, and scheduling of Interactive Multimedia Documents [Courtiat96, Layaida95, Mirbel2000, Jourdan2001].

Over the last few years, the consortium W3C proposed the language *Synchronized Multimedia Integrated Language (SMIL)* in order to provide the integration of Interactive Multimedia Documents (IMDs) on the web. Thus, SMIL has been largely applied for the design and presentation of these documents [Rutledge2001, Bulterman2001, Tae-Hyun2000]. SMIL 2.0 enhances the first version of this language providing several innovative features. One important characteristic of SMIL 2.0 is its flexible temporal model which eventually can lead the author to describe synchronization constraints that can not be resolved during the presentation of the document (in fact, tools like Grins, RealOne behave differently in such circumstances). For this purpose, it is necessary a formal methodology for the design of SMIL 2.0 documents.

In this paper we refer to a methodology for the design, the consistency analysis and the scheduling of Interactive Multimedia Documents. Our 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 [Courtiat2000]). The utilization of RT-LOTOS enables the complete management of the non-determinism inherent to Interactive Multimedia Documents during their presentation. In order to illustrate this methodology, we apply the SMIL 2.0 language for the design of complex Interactive Multimedia Documents. In particular, this paper presents an approach for the automatic translation of SMIL 2.0 documents into RT-LOTOS specification, which is applied in our methodology.

This paper is structured as follows: Section 2 presents a global view of the applied formal design methodology. Section 3 introduces an approach for the automatic translation of a SMIL 2.0 document into an RT-LOTOS specification. Sections 4 and 5 present, respectively, an overview of the verification and scheduling of IMDs. Section 6 discusses some aspects of the literature. Finally, section 7 presents some conclusions of this work.

2. The Formal Design Methodology

The proposed methodology aims to provide a framework for the design (specification, verification and presentation) of complex Interactive Multimedia Documents which relies on the formal description technique RT-LOTOS (illustrated in Figure 1).
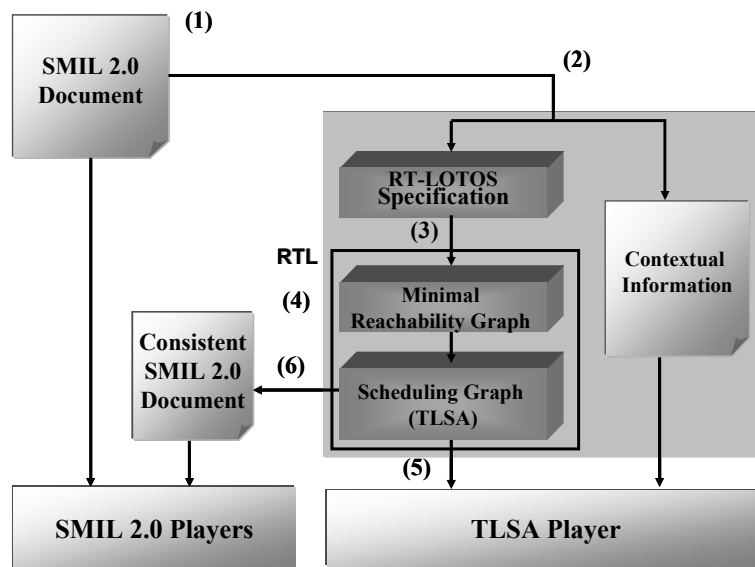

Figure 1. Formal design methodology of SMIL 2.0 documents

This methodology provides a high flexibility for the author during the edition of the IMD since he is able to describe his document using the authoring model of his preference, for instance SMIL 2.0. For this reason, this paper addresses the utilization of the proposed methodology as a means to provide a formal approach for the verification, scheduling and presentation of consistent SMIL 2.0 documents. This methodology consists in the following phases: (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 *contextual information* description which is used as a support for its presentation), (3) derivation of the minimal reachability graph from the RT-LOTOS specification through the RTL tool, (4) verification of consistency properties by means of reachability analysis, and (5) scheduling of the document presentation if there are valid (consistent) solutions for it. This methodology is further discussed in [Sampaio2003].

Following the classical approach, a SMIL 2.0 document can be presented directly by a SMIL player. However, this approach prevents the author from detecting and eventually correcting the potential inconsistencies of his document. In the opposite, in our approach, we provide the detection of inconsistencies and, later on, the presentation of consistent SMIL 2.0 documents. Furthermore, we still provide (6) the re-generation of a consistent SMIL 2.0 document based on the original structure of the document and on a consistent scheduling graph. In particular, we present the approach for the automatic translation of SMIL 2.0 documents into RT-LOTOS specifications in the next section.

3. Automatic translation of SMIL 2.0 Documents into RT-LOTOS specifications

The functionalities of SMIL 2.0 were defined into several modules so that the author is able to describe his document into different levels of complexity through the combination of these modules. In order to apply the proposed methodology for the formal design of SMIL 2.0 documents, it is necessary to distinguish the components of SMIL which describe the temporal dynamics and the presentation features (spatial and content) of a document. For this reason, we distinguish the components of SMIL 2.0 into *Operational* and *Contextual* components.
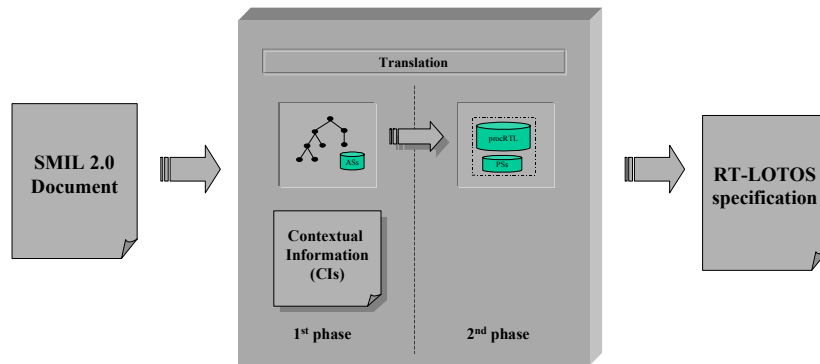

Figure 2.   Translation of the SMIL 2.0 Document

The *Operational* components of a SMIL 2.0 document represent the elements and attributes of this language which describe the logical and temporal behavior of the document (for instance, *par*, *seq*, *excl*, *begin*, *end*, *dur*, etc.). These components are translated into an RT-LOTOS specification. The *Contextual* components of a SMIL 2.0 document represent the elements and attributes which describe the spatial synchronization, content and the attributes describing the exclusive presentation of the components, animation, etc. These components are expressed by the *contextual information* description.

Once the components of a SMIL 2.0 document are identified, we can propose an approach based on the definition of intermediate structures representing all the components of the document and their temporal dynamics in order to preserve the semantics of a document during its translation. This approach is

structured into two main phases. In the first phase, we generate some data structures corresponding to the temporal semantics of the SMIL 2.0 document. In this phase we also generate the description of the contextual information. In the second phase, we generate some data structures corresponding to the structure of the RT-LOTOS specification, later on we generate the RT-LOTOS specification from these structures. A global view of this approach is illustrated in Figure 2.

To illustrate the utilization of this approach, we can consider the multimedia scenario of Figure 3 which describes the parallel presentation of the 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 the objects *audio1*, *audio2*, *video* and *interactiveButton* are, respectively, [0,5] seconds, [0,5] seconds, indefinite[1] and [0,20] seconds. As a synchronization constraint, (1) the presentation of *audio1*, *audio2* and *interactiveButton* must start at the same time, and (2) the presentation of *video* and *interactiveButton* must finish together. Also, if there is a user interaction during the presentation, the presentation of the scenario is interrupted, and it is restarted immediately.
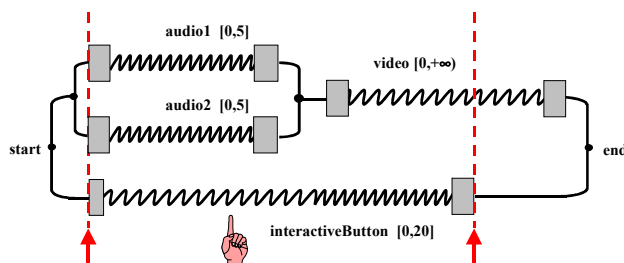

Figure 3. **An interactive multimedia scenario**

The first phase of this translation is presented on section 3.1, and the second phase of this approach is presented on section 3.2.

3.1. The First Phase of the Translation Approach

First of all, we carry out a lexical analysis on the SMIL 2.0 document to identify all the components of the document, their synchronization relationships, and their characteristics of presentation, in order to generate the data structures (*temporal tree* and *synchronization arcs*) necessary for the translation and the description of the *contextual information*. This phase is illustrated in Figure 4.
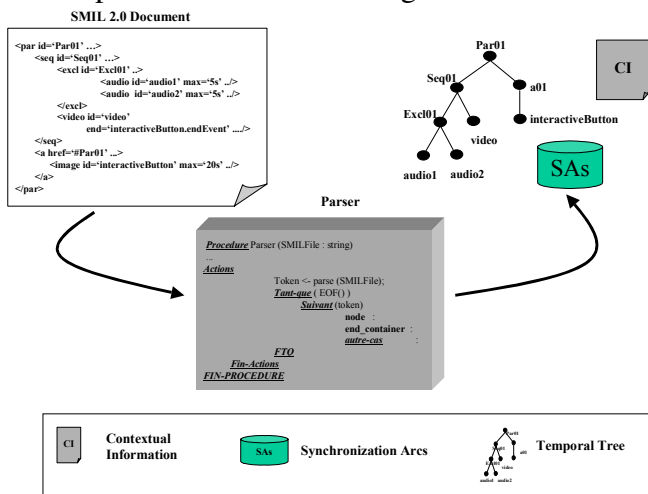

Figure 4. **First phase of translation approach**

---

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

3.1.1. Generation of the Temporal Tree and Synchronization Arcs

Since we want to describe the dynamic behavior (the components and their logical and temporal relationships) of a SMIL 2.0 document, we generate :

- A *temporal tree* which is a declarative representation that describes hierarchically all the components of a document (nodes) and their temporal characteristics. For instance, consider Figure 5 which illustrates the node *interactiveButton* of the previous scenario.
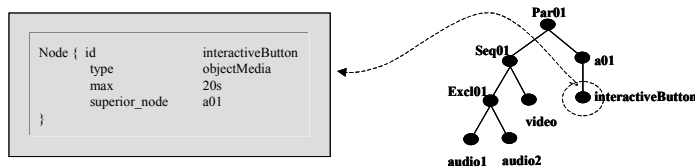

Figure 5.   Node of the temporal tree

The nodes of a *temporal tree* can be a media object, a container *par* (parallel composition), *seq* (sequential composition) or *excl* (exclusive composition), a *switch* element, a *prefetch* element, an *animation*, or an *anchor* a or *area*. For each node of the temporal tree some attributes are defined such as *dur* (presentation duration), *min* (minimal presentation duration), *max* (maximal presentation duration), *endType* (in this case, the semantics of the termination of a parallel composition can be *latest*, *earliest* or *master*), *Link_destination* (if it is an anchor), *superior_node*, *inferior_nodes*, etc.

- *Synchronization arcs* which are a declarative representation that describe the (causal) synchronization relationships defined among the components of a SMIL document. The synchronization relationships are defined by the attributes *begin* or/and *end* of presentation of a component of the document. For instance, consider the relation defined for the component *video* of the previous example, as follows :

```
…
<video id='video' end='interactiveButton.endEvent' ..../>
…
```

The attribute *end* of the element *video* determines that its presentation finishes when the presentation of *interactiveButton* is finished. This relation can be expressed by the *synchronization arc* presented in Figure 6.
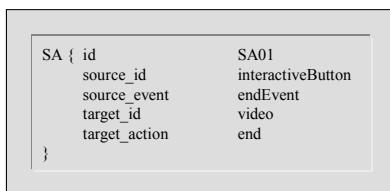
```
SA {  id                  SA01
      source_id           interactiveButton
      source_event        endEvent
      target_id           video
      target_action       end
}
```

Figure 6.   *Synchronization arc* associated with the object video

A *synchronization arc* is described by a *condition-action* where the *condition* is related to the occurrence of an event *begin*, *end* or *user interaction* of an element (container or a media object) probably associated with a temporal delay. An *action* is related to the execution of a *begin* or an *end* action of an element.

The first phase of the methodology is illustrated in Figure 7. The *temporal tree* and the *synchronization arcs* represent all the components of the document, their temporal and logical structures, and their synchronization relationships. Based on these structures, we can proceed with the second phase of the translation approach.
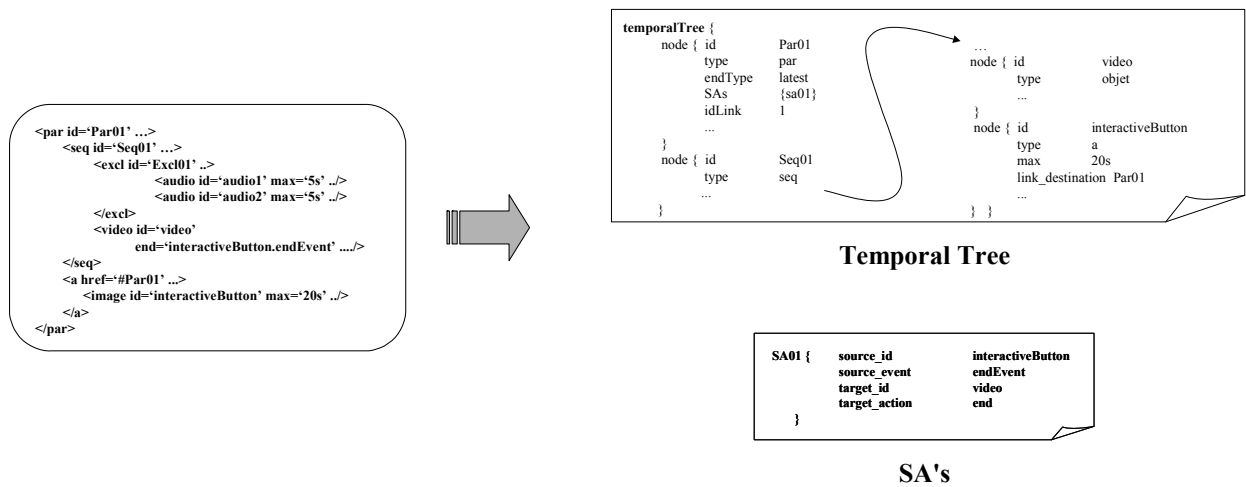
Temporal Tree

SA's

Figure 7.   Derivation of the *temporal tree* and *synchronization arcs*

3.1.2. Generation of the Contextual Information Description

The *contextual information* (CI) is automatically derived from the SMIL 2.0 document in order to be applied further on as a support for the presentation of the document. The CI is derived from the attributes describing the presentation characteristics of the components of the document, and it is composed basically of three sections, an optional *Main_Window*, which describes the features of the presentation window, a mandatory *Layout*, which describes the spatial synchronization of the components of the document, and a mandatory *Components*, which describes all the contextual presentation characteristics of media objects and *Excl* containers. Media objects components (e.g., *video_object*, *audio_object*, *text_object*, *animation_object* and *image_object*) describe all the information concerning the media to be presented in the document. *Excl_Container* describes the semantics for the mutual exclusive presentation of a container in particular.

SMIL 2.0

CI

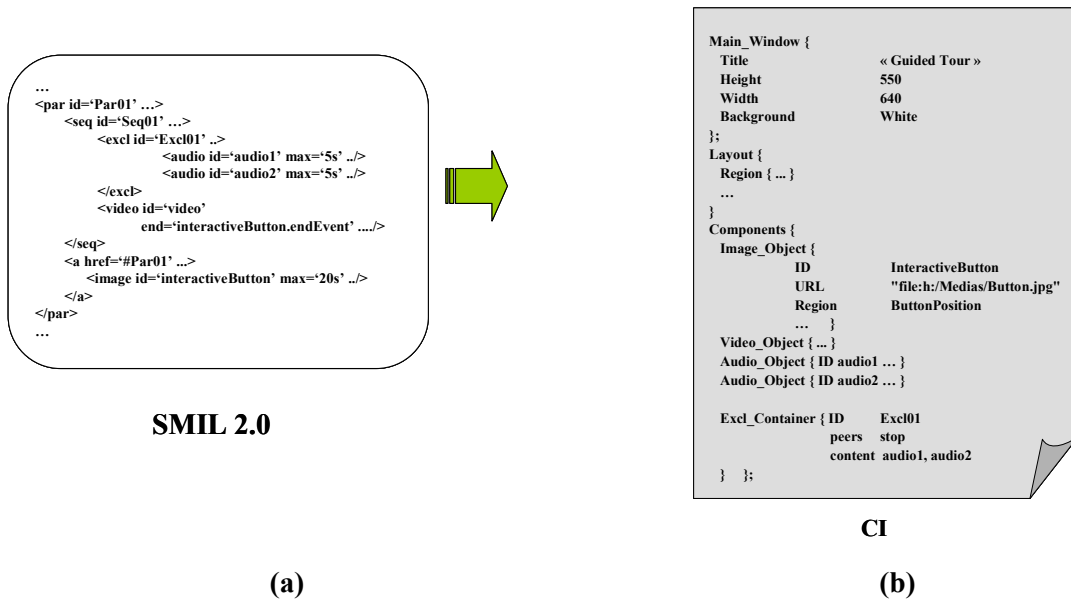**(a)**   **(b)**

Figure 8.   Derivation of the contextual information.

Figure 8 illustrates the derivation of the *contextual information* from the SMIL 2.0 document associated with the scenario of Figure 3.

3.2. The Second Phase of the Translation Approach

On this phase some appropriate structures are derived from the *temporal tree* and *synchronization arcs* before the generation of the RT-LOTOS specification. These structures are known as *ProcRTL* and *synchronization points (SP's)*. This phase is divided into three steps: (1) the generation of *ProcRTL* and *synchronization points*; (2) the transformation of the attributes of *ProcRTL*, and; (3) the generation of the RT-LOTOS specification.

3.2.1. Generation of ProcRTL and Synchronization Points

Figure 9 illustrates the high-level structure of an algorithm for the generation of *ProcRTL* and *synchronization points*.
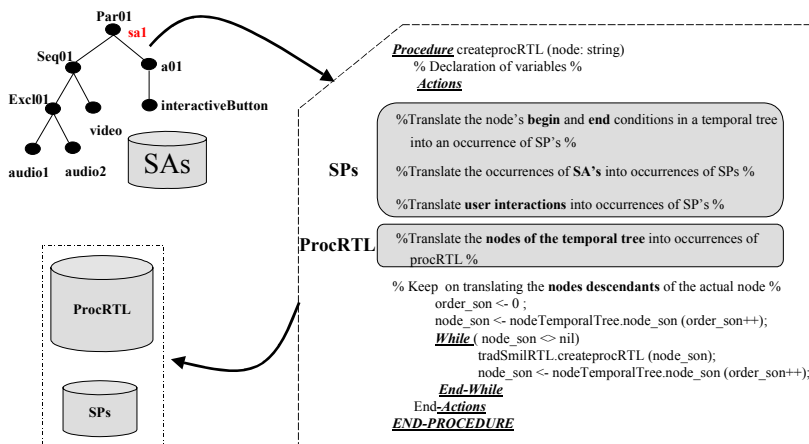

Figure 9.   Second phase of the translation approach

Following this algorithm, we perform a depth first-search on the *temporal tree* derived from the document, and generate one occurrence of *ProcRTL* for each node of the *temporal tree* and all the *synchronization points* associated with the current node of the *temporal tree*.
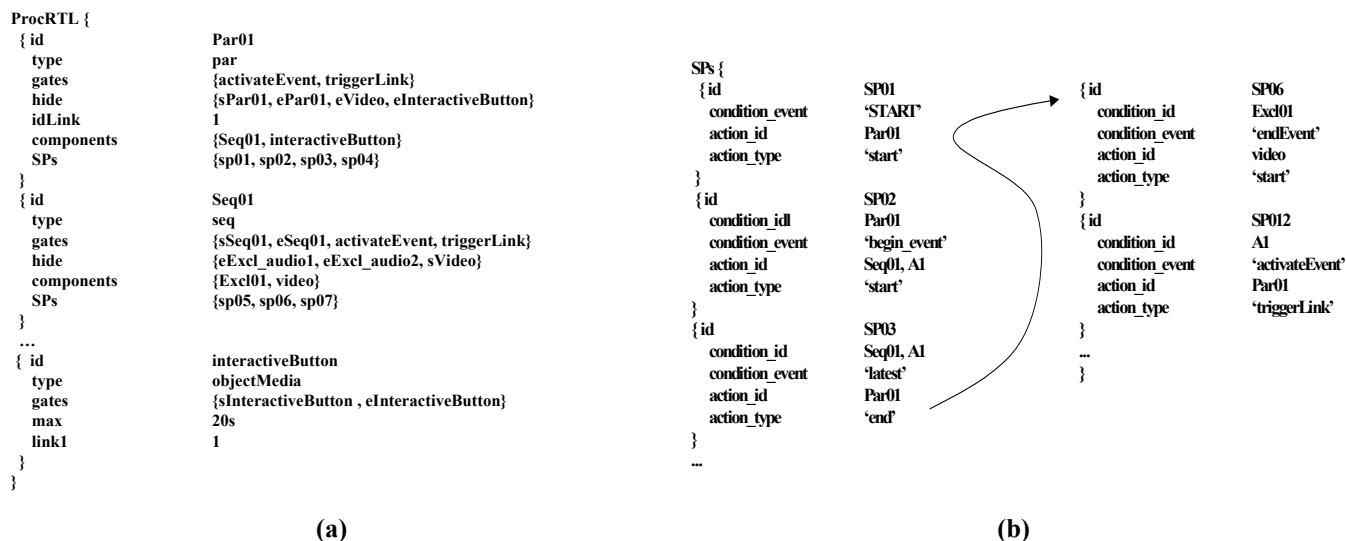

(a)   **(b)**

Figure 10.   *ProcRTL* and *synchronization points* derived for the previous example

ProcRTL is a data structure which represents all the elements of an RT-LOTOS process described by the attributes *id* (process id), *type* (type of node which can be *objectMedia*, *par*, *seq*, *excl*, *switch*, *prefetch*, *animation*, *a* and *area*), gates (which describes the observable gates of the process and represents the *start*

and *end* actions of the current node of the *temporal tree*), hide (which represents the actions from the *synchronization points* that do not represent the *start* and *end* actions of the current node), components (which describes the sub-processes of the current process), SPs (which describes the *synchronization points* associated with the process), *min* and *max* (which describe the minimal and maximal duration of the media object described by the current process), etc.

Synchronization points (SP's) describe all the synchronization relations of a node of the *temporal tree* such as the begin and end relations of a node, the occurrences of *synchronization arcs* for a node, and the user interactions over a node. As an example, consider Figure 10(a) and 10(b) which describe, respectively, the *ProcRTL* description and the *synchronization points (SPs)* for the previous example.

3.2.2. Transformation of the attributes of ProcRTL

Once the *ProcRTL* description and the *synchronization points* are derived, all the actions declared in *ProcRTL* must be transformed so that this description expresses correctly the actions associated with the *begin* and *end* of presentation of all media objects of the document. Furthermore, since a timed automata is generated later on the applied methodology for the verification of the obtained RT-LOTOS specification, we have to deal with a classical state space explosion problem. For this reason, some solutions were presented in [Sampaio2001a] and they are considered in this phase. In particular, these solutions propose the composition of actions according to some simple rules in order to reduce the size of the resultant automata.
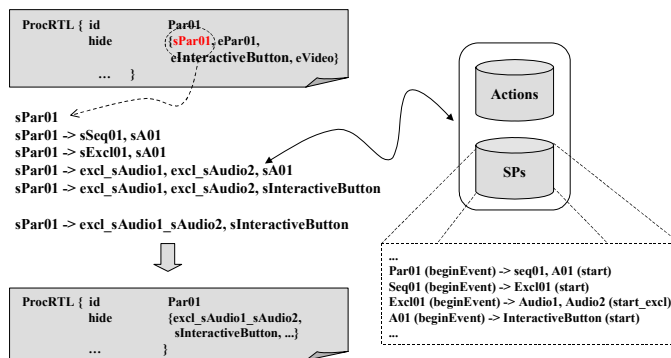

Figure 11.   Transformation and composition of actions

For instance, consider Figure 11 where the value of the attribute *hide* for the occurrence of ProcRTL *Par01* is {*sPar01*, *ePar01*, *eInteractiveButton*, *eVideo*}. In particular, action *sPar01* can be transformed. For this purpose, this transformation is based on *synchronization points (SP's)* and the resultant actions are registered in a list of actions (*Actions*) for further utilization. Firstly, SP's indicate that the value of *sPar01* as *sSeq01* and *sA01*. In a recursive research, the value of *sSeq01* becomes *sExcl01*. Then, the value of *sExcl01* becomes *excl_sAudio1* (exclusive start of audio1) and *excl_sAudio2* (exclusive start of audio2), and so on until the actions can not be transformed anymore. In a second step, the rules presented in [Sampaio2001a] for the composition of actions are taken into account for deriving the resultant actions, in this case, *excl_sAudio1_sAudio2* and *sInteractiveButton*.

3.2.3. Generation of the RT-LOTOS specification

The last step of the second phase is the generation of the RT-LOTOS specification based on *ProcRTL*, *synchronization points* and list of *Actions*, described by the high-level description of the algorithm presented in Figure 12. According to this algorithm, before generating the RT-LOTOS specification the attributes (identification of the target nodes) that describe the navigation on the document through hyper-links are updated in *ProcRTL*.
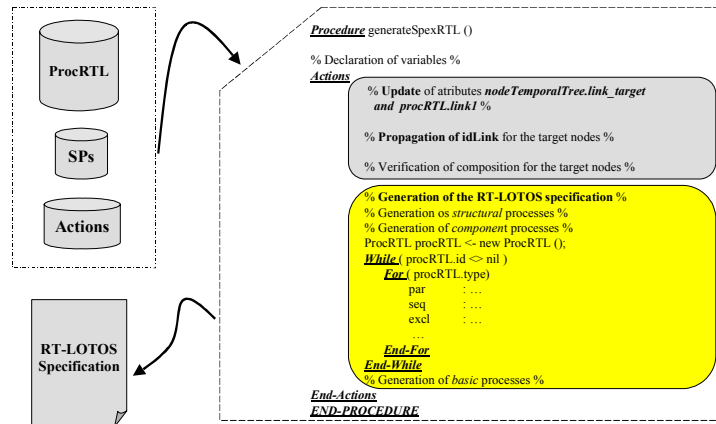
Figure 12. Generation of the RT-LOTOS specification

After updating the navigation mechanisms of a document, we can proceed with the derivation of the RT-LOTOS specification. For this purpose, we derive the main structure of the specification based on a library of processes defined as templates. Figure 13 illustrates the main structure of an RT-LOTOS specification and its components.
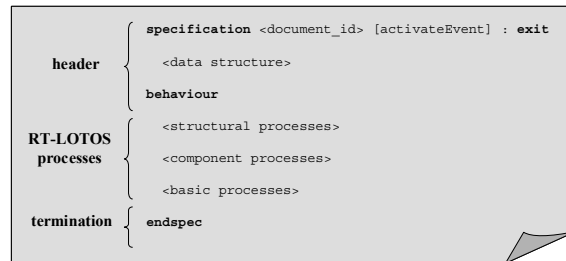
Figure 13. Structure of an RT-LOTOS specification

This structure is composed of three main sections: the *header* which describes the identification of the specification followed by the declaration of data types used on this specification. The *RT-LOTOS processes* which are derived into three separate steps:

(1) The generation of *structural processes*, which allow the hyper-link navigation on the document. These processes (called *Document*, *Trigger*, *mainBodyDoc* and *bodyDoc*) are illustrated by Figure 14.

Figure 14. Structural processes

The process *Document* encapsulates the global specification of the document. The process *Trigger* manages all the events generated during the presentation which are communicated to this process by the action *triggerLink*. Thus, this process can trigger the *start* of the presentation of the document, the start of presentation of a target node of a hyper-link or the *end* of the presentation of the document. The process *mainBodyDoc* describes the occurrence of a document presentation. This process is responsible for the communication between the temporal behavior of the document presentation and the process *Trigger*. Finally, process *bodyDoc* describes the temporal behavior of a document through the hierarchy of processes associated with its components;

(2) The generation of *component processes*, which describe the hierarchy of processes (*ProcRTL*) associated with the structure of a document. The types of these processes can be assigned as *par*, *seq*, *excl*, *switch*, *objectMedia*, *prefetch*, *animation*, *a* and *area*. The specification of these processes is presented into three different styles: (a) The specification of the processes *par*, *seq*, *excl* and *switch* (illustrated in Figure 15a); (b) The specification of the processes *objectMedia*, *prefetch* and *animation* (illustrated in Figure 15b), and; (c) The specification of the processes *a* and *area* (illustrated in Figure 15c).
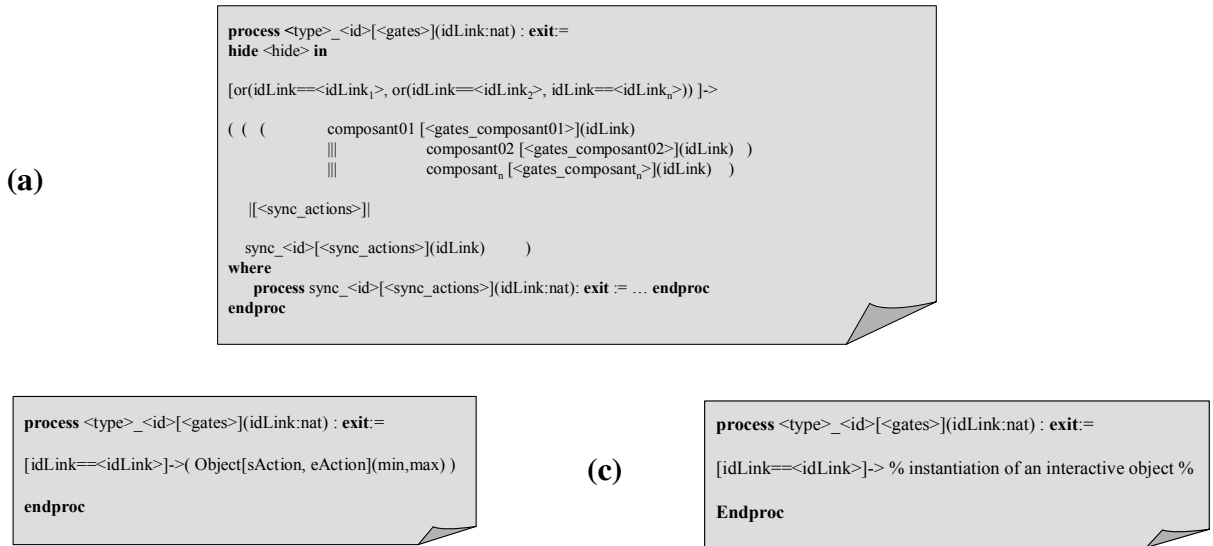
(a)

```
process <type>_<id>[<gates>](idLink:nat) : exit:=
hide <hide> in

[or(idLink==<idLink₁>, or(idLink==<idLink₂>, idLink==<idLink_n>)) ]->

( ( (           composant01 [<gates_composant01>](idLink)
        |||              composant02 [<gates_composant02>](idLink)   )
        |||              composant_n [<gates_composant_n>](idLink)   )

   |[<sync_actions>]|

   sync_<id>[<sync_actions>](idLink)       )
where
     process sync_<id>[<sync_actions>](idLink:nat): exit := … endproc
endproc
```

(b)

```
process <type>_<id>[<gates>](idLink:nat) : exit:=

[idLink==<idLink>]->( Object[sAction, eAction](min,max) )

endproc
```

(c)

```
process <type>_<id>[<gates>](idLink:nat) : exit:=

[idLink==<idLink>]-> % instantiation of an interactive object %

Endproc
```

Figure 15.   Component processes

(3) The generation of *basic processes*, which describe the template representation for the temporal behavior of a non-interactive or an interactive object. The RT-LOTOS process (*Object* process) describing the temporal behavior of a non-interactive object is depicted in Figure 16.

```
process Object [start,endRequest](I1,I2:nat) : exit :=

( start; delay(I1,I2) endRequest{I2-I1}; exit )

endproc
```
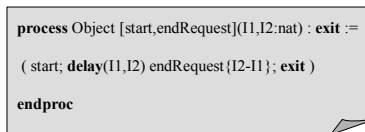
Figure 16.   RT-LOTOS process for a non-interactive object

The temporal behavior of interactive objects is described by the RT-LOTOS processes *InteractiveAncre*, *InteractivePlay* and *InteractiveStop*. The process *InteractiveAncre* describes the temporal behavior of an interactive object without possibility of navigation. The process *InteractivePlay* describes the temporal behavior of an interactive object allowing the simultaneous presentation of the source and target nodes. Contrarily, the process *InteractiveStop* describes the temporal behavior of an interactive object where the presentation of the source node is interrupted

before the start of the presentation of the target node. The processes *InteractiveAncre*, *InteractivePlay* and *InteractiveStop* are illustrated in Figure 17.

```
process InteractiveAncre [start,endRequest,activateEvent](I1,I2,obj:nat) : exit :=
(
  start;
  (  ( delay(I1) activateEvent{I2-I1} !obj; endRequest{0}; exit)
   [] ( delay(I2) endRequest{0}; exit )      )
)
endproc


process InteractivePlay [start,endRequest,activateEvent, triggerLink](I1,I2,obj,nLink:nat) : exit :=
( start;
  (  (delay(I1) activateEvent{I2-I1} !obj; triggerLink!nLink; stop)
                   [> ( delay(I2) endRequest{0}; triggerLink!99; exit)        )
)
endproc


process InteractiveStop [start,endRequest,activateEvent, triggerLink](I1,I2,obj,nLink:nat) : exit :=
( start;
  (              ( delay(I1) activateEvent{I2-I1} !obj; endRequest{0}; triggerLink!nLink; exit)
            [] ( delay(I2) endRequest{0}; triggerLink!99; exit)           )
)
endproc
```

Figure 17.   RT-LOTOS processes for an interactive object

Finally, the *termination* of the specification is described by the declaration *endspec*.

An appropriate RT-LOTOS specification style for modeling Interactive Multimedia Documents must be able to describe all the components of the documents and their synchronization constraints. As we presented previously, all the synchronization constraints of the document are described by the *synchronization points (SPs)*. In order to model the synchronization constraints associated with an occurrence of *ProcRTL* (described by an RT-LOTOS process) we apply an RT-LOTOS process called *sync_<name_of_the_node>*. Consequently, each occurrence of *synchronization points* associated with an occurrence of *ProcRTL* is translated into an RT-LOTOS behavior expression declared in process *sync*.
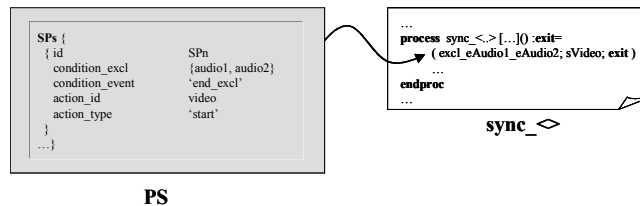


```
SPs {
  { id                SPn
    condition_excl    {audio1, audio2}
    condition_event   'end_excl'
    action_id         video
    action_type       'start'
  }
…}
```

PS

```
…
process sync_<.> […]() :exit=
  ( excl_eAudio1_eAudio2; sVideo; exit )
  …
endproc
…
```

sync_<>

Figure 18.   Translation of a synchronization point into a sync_ behavior

For instance, consider Figure 18 which describes the translation of a *synchronization point* into an expression behavior of RT-LOTOS declared in a sync_ process. Furthermore, some composition rules were also defined in order to express the synchronization among these expressions [Sampaio2003]. A global view of the resultant RT-LOTOS specification for the previous example is illustrated in Figure 19.

```
specification document[activateEvent] : exit
…
behaviour
 hide start, end  in
      Document[start,end,activateEvent]
 where
     process Document[start,end,activateEvent] : exit:=
       let idLink:nat=1 in
       hide endRequest, triggerLink, linkExec, presentDoc  in
        …
       where
          process Trigger[triggerLink,endRequest,linkExec, presentDoc]: exit:= ... endproc
          process mainbodyDoc[activateEvent, triggerLink, linkExec, presentDoc](idLink:nat) : exit:=
             bodyDoc[activateEvent, triggerLink, linkExec, presentDoc](idLink)
          where
             process bodyDoc[activateEvent, triggerLink, linkExec, presentDoc](idLink:nat): exit:= ...
             endproc
          endproc
       process par_Par01[activateEvent, triggerLink](idLink:nat) : exit:=
       hide excl_sAudio1_sAudio2, sInteractiveButton, eInteractiveButton_eVideo  in
         [idLink==1]->
         (  ( seq_Seq01[excl_sAudio1_sAudio2, eInteractiveButton_eVideo, activateEvent, triggerLink](idLink)
                       |[eInteractiveButton_eVideo]|
             InteractiveAncre[sInteractiveButton, eInteractiveButton_eVideo,activateEvent, triggerLink](0,20,1)  )
           |[excl_sAudio1_sAudio2, sInteractiveButton, eInteractiveButton_eVideo, triggerLink]|
             sync_Par01[excl_sAudio1_sAudio2, sInteractiveButton, eInteractiveButton_eVideo, triggerLink](idLink)          )
         where
           process sync_Par01[excl_sAudio1_sAudio2, sInteractiveButton, eInteractiveButton_eVideo, triggerLink](idLink:nat): exit :=
               (        ( excl_sAudio1_sAudio2; sInteractiveButton{0}; exit )
                    ||| ( eInteractiveButton_eVideo; ( triggerLink!0; exit [] triggerLink!1; exit ) )      )
           endproc
         endproc

         process seq_Seq01[excl_sAudio1_sAudio2, eVideo, activateEvent, triggerLink](idLink:nat) : exit:= … endproc
         process excl_Excl01[excl_sAudio1_sAudio2, excl_eAudio1_eAudio2, activateEvent, triggerLink](idLink:nat) : exit:=
               …
         endproc
     endproc

   process InteractiveAncre [start,endRequest,activateEvent, triggerLink](I1,I2,nLink1:nat) : exit := ... Endproc
   process Object [start,endRequest](I1,I2:nat) : exit := … endproc
 endspec
```

Figure 19.   RT-LOTOS specification

4. Verification of Interactive Multimedia Documents

The approach presented for the verification of temporal consistency of an Interactive Multimedia Document (IMD) was initially introduced in [Courtiat96]. Based on the derived RT-LOTOS specification, we can proceed with the formal verification of the consistency properties of the document. The basic idea about verifying the temporal consistency is to analyze the reachability of the action characterizing the end of presentation of the document (action *end*), in a *minimal reachability graph* generated from the RT-LOTOS specification. The notion of consistency can be described by the following definitions:

Definition 1: a path on the *reachability graph* is consistent if the initial node of this graph leads to the occurrence of the action representing the end of the presentation of the document (represented by transition labeled by the action *end*).

Definition 2: a document is *consistent* if there is at least one consistent path leaving the initial node of the reachability graph associated with this document.

In order to avoid any inconsistency during the scheduling of a presentation, we can apply a correction technique by which all the inconsistent paths generated by the occurrence of these events are removed from the *reachability graph*. This approach enables the author of a document to express only the temporal constraints which he considers essential (which are known previously), leaving the others as flexible as possible. The possible inconsistencies induced by these flexible constraints will be detected by the formal verification. The technique of correction allows to replace automatically the loose constraints by precise constraints ensuring the global consistency for the presentation of the document (of course, if there is a valid scheduling solution, which is guaranteed by the existence of least one consistent path on the reachability graph).

For instance, consider the *reachability graph* illustrated in Figure 20(a) which is related to the scenario presented in Figure 3. In this case, the presentation of the document can be considered consistent because some paths of this graph lead to the occurrence of the action end characterizing the end of presentation of

the document. In this case, there are also inconsistent paths which occur since the duration of presentation of *Video* exceeds the global duration of the presentation of the document not respecting its synchronization constraints.

We can carry out the scheduling of this document by using the technique of correction. Consequently, that will enable the consistent presentation of the document, by controlling in particular the presentation duration of *Video* so that its synchronization constraints previously defined by the author of the document can be respected.

5. Scheduling and Presentation of Interactive Multimedia Documents

The scheduling of an IMD can be carried out based on the results of the verification of the temporal consistency during the generation of a scheduling graph, detailed in [Lohr2002]. The scheduling graph describes an operational specification of all the valid (consistent) executions for an Interactive Multimedia Document. The scheduling graph applied by our approach is called *Time Labelled Scheduling Automaton (TLSA)*. The TLSA associated with the scenario of Figure 3 is presented in Figure 20(b).
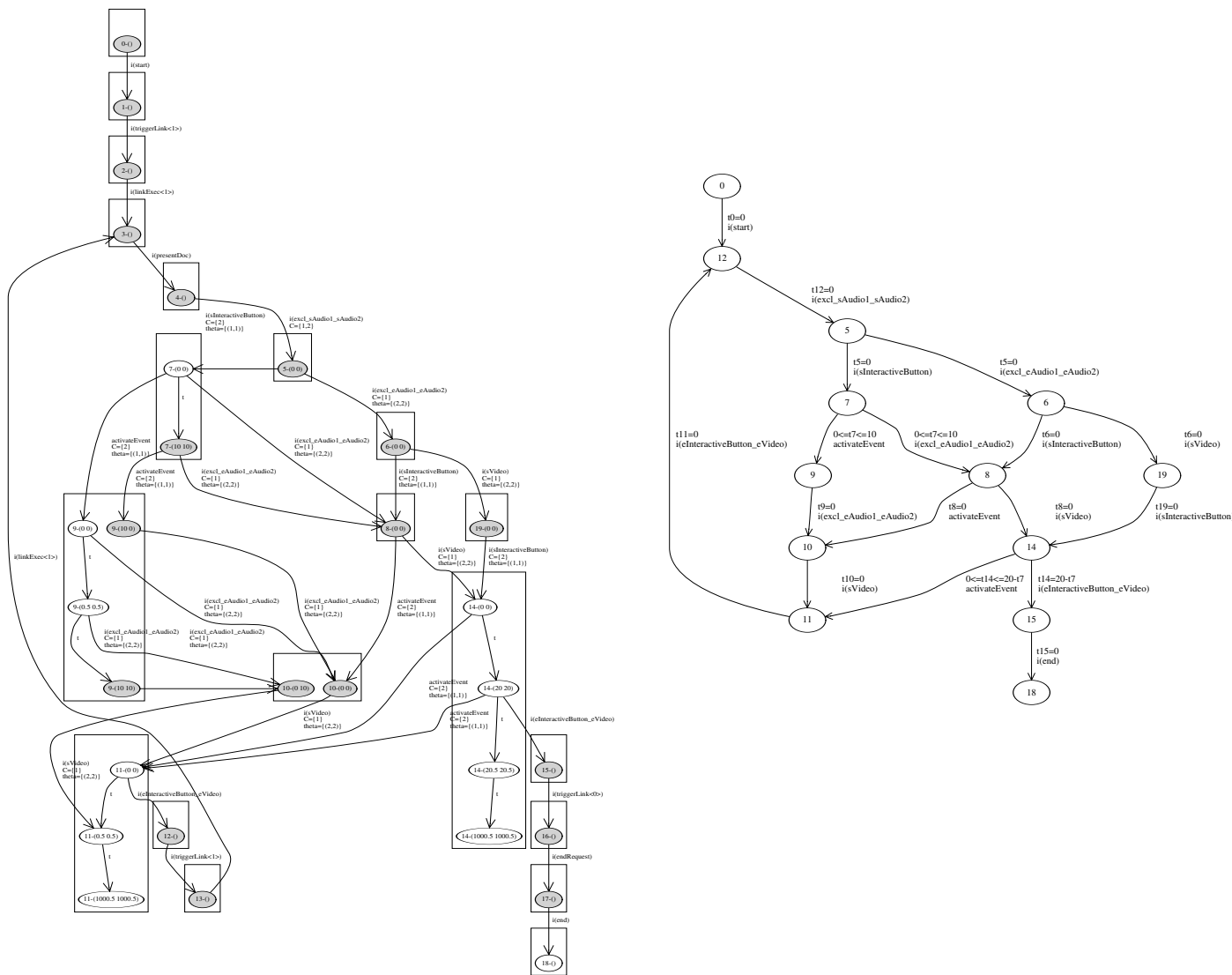
(a)   (b)

Figure 20.   Reachability graph and TLSA associated with the previous scenario

13

The TLSA was proposed as a scheduling graph which allows the temporal formatting (the definition of the valid temporal intervals for the presentation) of the 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 stopping a presentation, to announce an occurrence of a user interaction, etc. Thus, some scheduling policies were defined in order to allow the orchestration of the document based on the TLSA. These scheduling policies are presented in detail in [Santos2000, Sampaio2001b, Sampaio2003].

A prototype for the presentation of interactive multimedia documents was implemented based on these scheduling policies. The TLSA Player is based 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 22 illustrates a snapshot of the TLSA Player.
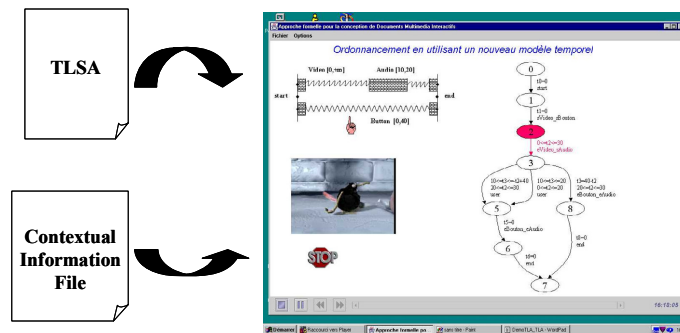

Figure 21.   Snapshot of the TLSA Player

6. Related Works

The design of Interactive Multimedia Documents (IMDs) has been addressed in the literature by different concepts and approaches to deal with issues such as modeling the temporal structure, and further consistency checking and scheduling. A current solution consists in using a temporal graph (such as a directed acyclic graph) and specific algorithms (such as shortest path solution, or based on linear programming) to determine the valid solutions for the presentation of a document. Further approaches can also be based on: timeline (GRiNS [GRiNSComposer]), linear programming (Firefly [Buchanan95]), constraint graphs (CHIMP [Candan96], MADEUS [Jourdan97], ISIS [Song99], Tiempo [Wirag99], IMAP [Vazirgiannis98], Hyperprop [Rodriguez97], MAVA [Hauser99]), transition systems (TOCPN [Yoon98], Dynamique Extended Finit State Machines - DEFSMs [Huang99]), partial order associated with linear programming and constraint graphs [Rivière2002], and state of the components of document (FLIPS [Schnepf96]).

This paper is related to a methodology for the formal design of Interactive Multimedia Documents. This methodology is based on a transition system derived from the RT-LOTOS specification which allows for further formal consistency checking and scheduling of the document. Furthermore, this methodology also provides the presentation of complex and consistent IMDs and the management of the temporal non-determinism of the document during the presentation.

5. Conclusion and future works

The methodology presented in this paper is related to the formal design of Interactive Multimedia Documents (IMDs) based on the formal description technique RT-LOTOS. 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). Thus, the author designs his document using the high level language (or the paradigm) of his preference, and then the temporal and logical behavior of his document is translated into an RT-LOTOS specification. The compositional characteristic of the RT-LOTOS specifications enables to model complex temporal scenarios by the successive composition of simpler behaviors. Since RT-LOTOS has a formal semantics, it is also possible to carry out the verification of properties of temporal consistency, and to carry out the scheduling of the document based on a timed automata model (TLSA) derived from the subjacent RT-LOTOS specification. To illustrate the utilization of this methodology, we adopted the language SMIL 2.0 for the modeling of complex Interactive Multimedia Documents.

An approach for the translation of a high-level language must take into account the semantics of a document concerning its components and its synchronization relations. The approach that we presented in this paper supports the definition of intermediate structures for the representation of the semantics of a document. This approach was defined into two main phases according to the need to describe the structures related to the high level model (in the case of SMIL 2.0), and the structures related to the RT-LOTOS specification. Furthermore, it was necessary to define a suitable RT-LOTOS specification style in order to model the hierarchical composition of the components of the document, its synchronization relations, and the internal mechanisms of navigation inside a document.

As for future works, two major issues can be considered : (a) the implementation of an automatic translator of SMIL 2.0 into RT-LOTOS specifications based on the approach presented in this paper, and (b) the modeling and implementation of solutions for the verification feedback for the author of a document. In particular, this tool allows for the integration of the edition and verification phases enabling an author to automatically detect the potential inconsistencies of his document, and to correct it, if possible.

6. Acknowledgements

7. References

[Buchanan95]   Buchanan, M.C.; Zellweger. Automatically generating consistent schedules for multimedia documents. In ACM-Multimedia Systems Journal 1:2, pp 55-67, 1995.

[Bulterman2001] Bulterman, D.C.A. *SMIL 2.0 – Part 1 : Overview, Concepts and Structure*. IEEE Multimedia, pp. 82-88, October-December 2001.

[Cadan96]   Cadan, K.S.; Prabhakaran, B.; Subrahmanian, V.S. A Framework for supporting multimedia document authoring and presentation. In 4th ACM Multimedia '96, pp 329-339, Boston, USA, November 1996.

[Courtiat96]   Courtiat, J.-P.; Oliveira, R.C. Proving Temporal Consistency in a New Multimedia Synchronization Model. In 4th ACM Multimedia'96, pp.141-152, Boston, USA, Nov. 1996.

[Courtiat2000]   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.

[GRiNSComposer]   Oratrix Development BV. SMIL 2.0 Evaluation KIT. Online. Available: http://www2.oratrix.nl/W3C/pressRelease2.

[Hauser99]   Hauser, J. Realization of an extensible multimedia document model. In Multimedia '99 - Media Convergence: Models, Technologies and Applications. Springer-Verlag. pp 113-122, Wien, 1999.

[Huang99]   Huang, C.-M.; Lin, C.-H.; Wang, C. Specifying and Executing Interactive Multimedia Presentations using the Formal Approach. In : proceedings of National Science Concil, vol. 23, No. 4, pp. 495-510, 1999.

[Jourdan97]   Jourdan, M.; Ladayia, N.; Roisin, C.; Sabry-Ismail, L. An Integrated Authoring and Presentation Environment for Hypermedia Documents. In: Proc. of Int. Conf. on MultiMedia Modeling - MMM'97, Singapore, Nov. 1997. World Scientific.

[Jourdan2001]   Jourdan, M. A formal semantics of SMIL: a Web standard to describe multimedia documents. Computer Standards and Interfaces, 23 (2001), pp. 439-455, 2001.

[Layaida95]   Layaida, N.; Keramane, C. Mantaining Temporal Consistency of Multimedia Documents. In ACM Workshop on Effective Abstractions in Multimedia, San Francisco, 1995.

[Lohr2002]   Lohr, C.; Courtiat, J.-P. From the specification to the scheduling of time-dependent systems. In: 7th International Symposium on Formal Techniques in Real-Time and Fault Tolerant Systems, Oldenburg, Germany, lncs, september, 2002.

[Mirbel2000]   Mirbel, I.; Pernici, B.; Sellis, T.; Tserkezoglou, S.; Vazirgiannis, M. Checking Temporal Integrity of Interactive Multimedia Documents. In VLDB journal, 2000.

[Riviere2002]   Riviere, N.; Bradin-Chezaviel, B.; Valette, R. Propagation de contraintes et ordonnancement de documents multimédias. Rapport LAAS No02044. In : 5ème Journée Doctorales Informatique et Réseaux (JDIR'2002), Toulouse (France), 4-6 Mars 2002, pp.211-218.

[Rodrigues97]   Rodrigues, R.F.; Soares, L.F.G.; Souza, G.L. Authoring and Formatting of Documents Based on Event-Driven Hypermedia Models.   In: Proc. of IEEE Conf. on Protocols for Multimedia Systems, Multimedia Networking - PROMSMmNet'97, Santiago, Chile. Nov. 1997.

[Rutledge2001]   Rutledge, L. SMIL 2.0 – XML for Web Multimedia. IEEE Internet Computing, pp. 78-84,   September-October 2001.

[Santos2000]   Santos, C.A.S.; Sampaio, P.N.M. ; Courtiat, J.-P Da análise de consistência à formatação temporal de um documento hipermídia usando RT-LOTOS. In: VI Brazilian Symposium on Multimedia and Hypermedia Systems (SBMIDIA), Natal (Brazil), June 14-16 2000, 15p.

[Sampaio2001a] Sampaio, P.N.M.; Lohr, C.; Courtiat, J.-P. An Integrated Environment for the Presentation of Consistent SMIL 2.0 Documents. In ACM Symposium on Document Engineering 2001 (DocEng'2001), Atlanta, Georgia, USA, November, 2001.

[Sampaio2001b] Sampaio, P.N.M.; Courtiat, J.-P. Scheduling and Presenting Interactive Multimedia Documents. In IEEE International Conference on Multimedia and Expo (ICME'2001), Tokyo, Japan, August 2001.

[Sampaio2003]   Sampaio, P.N.M. Conception formelle de documents multimédia interactifs: Une approche s'appuyant sur RT-LOTOS. 2003. Ph.D. Thesis. Université Paul Sabatier – Toulouse III, Toulouse, France.

[Schnepf96]   Schnepf, J.; Konstan, J.A.; Du, D. Doing FLIPS : FLexible Interactive Presentation Synchronization. Distributed Multimedia Center, Department of Computer Sciences, University of Minnesota, 1996. ftp://ftp.cs.umn.edu/users/du/papers/flips.ps

[Song99]   Song, J.; Ramalingam, G.; Miller, R.; Yi, B. Interactive authoring of multimedia documents in a constraint-based authoring system. In: Multimedia Systems 7:424-437 (1999), Springer-Verlag, 1999.

[Tae-Hyun2000] Tae-Hyun et all. Simple and Consistent SMIL Authoring: No more editing and no more error. In IEEE Multimedia Computing on the World Wide Web 2000.

[Vazirgiannis98] Vazirgiannis, M.; Trafalis, M.; Stamati, Y.; Hatzopoulos, M. Interactive Multimedia Scenario: Modeling and Rendering.   In: Proc. of the Multimedia track of ACM-SAC'98 Conference. (aussi disponible en : http://www.dbnet.ece.ntua.gr/~michalis/publications.html)

[Wirag99]   Wirag, S. Specification and scheduling of adaptative multimedia documents. In Technical report TR-1999-04, University of Stuttgart, January 1999

[Yoon98]   Yoon, K.; Berra, P.B. TOCPN : Interactive Temporal Model for Interactive Multimedia Documents. In : International Workshop on Multimedia Database Management Systems (IW-MMDBMS'98), Dayton, Ohio, USA, 1998.