A DESIGN METHODOLOGY FOR THE FORMAL SPECIFICATION AND VERIFICATION OF HYPERMEDIA DOCUMENTS

C. A. S. Santos[£], J.-P. Courtiat[£]
P. de Saqui-Sannes[£,¥]

[£] LAAS-CNRS, 7 avenue du Colonel Roche,
31400 Toulouse - Cedex - France

[¥] ENSICA, 1 Place Emile Blouin,
31056 Toulouse - Cedex - France

Abstract

Hypermedia authoring tools usually suffer from a lack of validation capabilities that would make them capable of checking a document against temporal inconsistencies. The document design method proposed in the paper is meant to overcome this problem. The starting point is a document description provided in a high-level modeling technique featuring hypermedia basic concepts such as nodes (including composite nodes), anchors and links. The high-level document is then automatically translated into a RT-LOTOS formal specification on which classical reachability analysis techniques are applied, making it possible to check the temporal consistency of the document structure. A simple example is used along the text to illustrate the proposed approach.

1. Introduction

A fundamental problem in hypermedia document design is to characterize the temporal structure of such type of document. The temporal structure can be expressed by a conjunction of synchronization constraints that partly depend on user interactions (hence, the hypermedia character of the document). The quality of the document

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35394-4_29

S. Budkowski et al. (eds.), *Formal Description Techniques and Protocol Specification, Testing and Verification*

presentation heavily relies on the global consistency of these synchronization constraints.

So far, few work has been done on the identification and analysis of consistency properties for hypermedia documents (Buchanan, 1993a) (Courtiat, 1996) (Lakayda, 1995). Main reasons are twofold:

- Many models do not have any formal background to support automated proofs of consistency properties.
- The importance of these properties has often been minimized, particularly because they are sometimes obvious on models with a limited expression power (like the timeline model (Blakowski, 1996) largely used in first generation commercial authoring tools).

Hypermedia documents have been a reality for many years, but their current and already important deployment (like WEB pages and CD-ROMs) masks several limitations to be dealt with in the products of the next generation. Examples of limitations include the structure essentially static of the documents, and poor semantic models for expressing temporal synchronization, etc.

Increasing dynamism in documents together with the use of more expressive models may result in a non-consistent temporal structure (Blakowski, 1996). Hence the need for improving the design process with a technique able to check the document temporal consistency. Early results on temporal consistency for hypermedia documents have been reported in (Courtiat, 1996), based on a new temporal synchronization model. That approach was however a not general proposal since the synchronization model was directly re-engineered from the underlying formalism, namely RT-LOTOS (Courtiat, 1995), a temporal extension of LOTOS. The high level model used in (Courtiat, 1996), and consequently the hypermedia document design approach, featured three major weaknesses:

- The model was specific.
- It did not handle completely the link concept, which is classical in hypermedia systems.
- It did not distinguish a node type from a node instance; this was considered as the major limitation, since it was not easy to re-use part of a document (a node) in the design of another document.

An enhanced approach is discussed in this paper. It overcomes the previous limitations and generalizes the concept of temporal consistency. The main contribution is the definition of an hypermedia design method which includes an automatic translation from a high-level hypermedia authoring model into a RT-LOTOS specification; this authoring model relies on those objects that are usually found in hypermedia document specifications, namely nodes (including composite nodes), anchors and links. This design method is applied here to the NCM authoring model (Casanova, 1991), and it is currently being implemented within the HyperProp Authoring System (Soares, 1995).

The paper is organized as follows. Section 2 introduces the high level authoring model adopted in this work. Section 3 describes the translation procedure from the

authoring model to the RT-LOTOS FDT. Section 4 surveys RT-LOTOS verification techniques, with their RTL tool support, and shows how to apply them to prove hypermedia document consistency properties by reachability analysis. This section also illustrates some verification results that have been obtained on the running example used throughout the paper. Some conclusions are finally given in Section 5.

2. Modeling hypermedia documents with temporal constraints

2.1. Hypermedia authoring systems

Hypermedia authoring systems make it possible to integrate different media types, and to define temporal and spatial[1] constraints among the document components. They offer well suited capabilities for structured document design, including navigation control and presentation monitoring. An authoring tool must be easy to use and sufficiently expressive in order to not impose a cognitive overhead to the authors. It is now admitted that a better expressiveness increases the probability to generate inconsistent documents.

Hypermedia document presentations are real-time applications. Hypermedia presentations also depend on user interactions and are therefore dynamic. For instance, a media content may, depending on some user interaction, be presented either as a text or as an audio; presentation durations are accordingly different, and synchronization constraints are to be defined at presentation time. Consistency verification becomes much more difficult.

Authoring systems usually postpone design validation to the run-time stage. This often happens too late, and the lack of well-established procedures makes the validation process uncertain. On the other hand, a complete formal approach is too complex for end-users, which prefer a high-level graphical language ranging from basic timelines to object-oriented models (Casanova, 1991) (Fraissé, 1996). The approach proposed in this paper is meant to allow the designer to use his or her favorite authoring tool. The latter's output is then automatically translated into a RT-LOTOS formal specification that is checked against design errors, using traditional verification techniques.

2.2. Hypermedia authoring models

Various informal and formal hypermedia modeling approaches have been proposed in order to describe the temporal structures of hypermedia documents. Main approaches have been classified into four categories (Blakowski, 1996):

- Interval-based specifications offer several operators on temporal intervals that characterize elementary media with minimal synchronization constraints to be satisfied in a compound document (Wahl, 1994).
- Axes-based specifications such as temporal lines are widely used in commercial tools but their lack of expressive power is notorious.

- Control Flow-based specifications include hierarchical trees, reference points as well as Petri net based models (Willrich, 1996).
- Event-based specifications are the most expressive, but the price to pay is in the complexity of edition and update functions (Buchanan, 1993b) (Casanova, 1991) (Faissé, 1996).

The hypermedia design approach presented in this paper is currently being implemented for hypermedia documents expressed in the *Nested Context Model* (NCM) the conceptual model of the HyperProp authoring system (Soares, 1995). NCM is a high-level authoring model based on the basic concepts of the hypermedia/hypertext community (such as *nodes*, *anchors* and *links*). It permits also to structure a complex hypermedia document by means of the composite node concept and belongs to the category of event-based specifications as far as the document temporal structure description is concerned. Basic features of the NCM authoring model will be introduced intuitively in the next paragraph. Details on NCM and the HyperProp system are available in (Casanova, 1991) (Rodrigues, 1997). A detailed comparison between NCM and hypermedia models used in Firefly (Buchanan, 1993b), CMIF (van Rossun, 1993), I-HTSPN (Willrich, 1996) and others, with respect to temporal and spatial synchronization aspects as well as to the semantic power of their compositions, can be found in (Rodrigues, 1997).

It is important to stress that the scope of the proposed design approach goes far beyond a particular authoring model or system, and could therefore be adapted to another one.

2.3. Hypermedia documents based on compositions

Hypermedia systems are characterized by the existence of complex relationships among the document components. Such systems also include mechanisms intended to structure the document in order to reduce the so-called *lost in the hyperspace* problem. Such a structured definition is desirable as it carries built-in concepts of modularity, encapsulation and abstraction. Conceptual models with composite nodes, like models that allow nested compositions, support such mechanisms.

Nodes are fragments of information and *links* interconnect them into networks of related nodes. *Anchors* represent an area within the content of a node that may be associated either to the source or destination of a link. The NCM model goes one step further by distinguishing two basic classes of nodes, called *content* and *composite* nodes, the latter being the central concept of the model (Soares, 1995).

Intuitively, the *content node* contains data whose internal structure, if any, is application dependent (they are the usual hypermedia nodes). The class of content nodes can be specialized into other classes (*text, video, audio, image*, etc.), as required by the applications.

A *composite node C* is a node whose content is a collection *S* of nodes and links. Note that a component may be included more than once in *S*. However, in this paper, we will assume links and nodes collection as a set, without loss of generality. An important restriction however must be done: a node cannot be recursively contained in itself. It is worth to note that composite node contains nodes and links, generalizing some models that group only nodes in compositions.

Composite nodes have several desired properties, such as:

- Compositions nesting, that is, compositions that contain other compositions.
- Grouping of the components of a document and the relationships among them independently of their types (synchronization relationships for presentation, selection relationships for usual hyperlink navigation, etc.).
- Use of the composite nodes as a new type of node, in all senses, that is:
  - They can be presented - since in a presentation, it is important to exhibit not only the data content of a document, but also its structure as it is specified in the composite node (for example, when one accesses a book chapter modeled as a composite node, besides seeing its content, one may want to visualize its section structuring).
  - Different entry points in a composition can be defined, i.e., in a composition, components may have different presentations, depending on the entry point. For instance, the duration of a composition (duration of its component exhibition) will depend not only on the duration of its components, but also on the associated entry point. In HyperProp system, descriptor objects are used to specify how and with which tool the associated node will be presented. Using distinct descriptors, one can define different presentations for the same data object. For example, a media segment can be synthesized as an audio, using descriptor D_1, or presented as a text using descriptor D_2.
  - Relations among compositions can be defined.
- Inheritance in the composition nesting, in the sense that relations can be defined in a composition *C*, referencing components recursively contained in *C*. This mechanism is extremely important in object reuse.
- Composite node presentation to help user navigation through a document - what can require the use of some filtering mechanism to present the document structure in order to lessen the user disorientation.

2.4. Example

Figure 1 depicts the composite node1 that includes 4 content nodes (node2, node3, node4 and node8), and another composite node (composite node5), which includes two content nodes (node6 and node7). The name of each node is followed by a time interval, which represents respectively the minimal and the maximal duration defined for that node. The duration of the content node may be estimated by the document author depending on the quantity of information to be presented. For a composite node, the situation is a little bit more complex since the termination of the node presentation depends on the internal logic of the node. If one does not want to put any duration constraint on the node, one specifies only $[0,\infty]$ avoiding consequently the risk to induce undesirable temporal consistencies in the node specification. The global behavior of the document is characterized by composite node1, which may informally be described as follows:

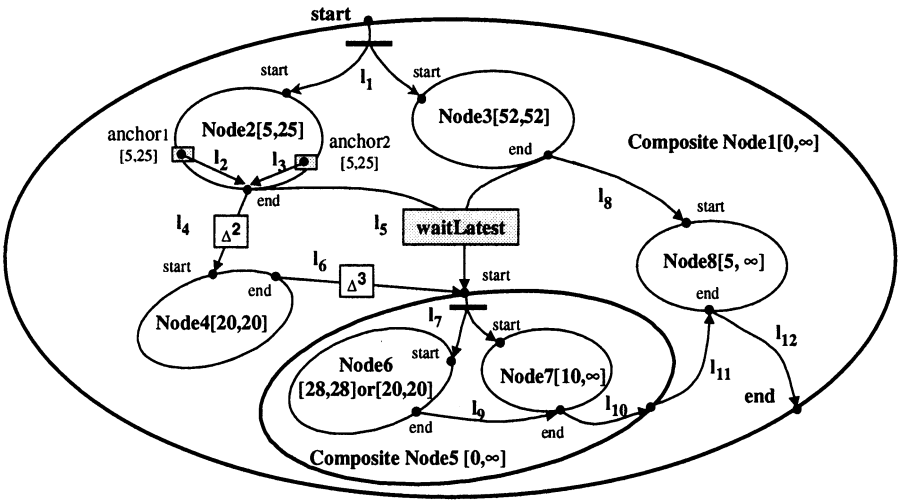

Figure 1 – Structure of the composite node1.

- The *start* of node1 leads to the immediate and concurrent *start* of node2 and node3 (see the link l_1).
- Within node2, two anchors have been defined. Node2 should terminate as soon as one of the anchors has been triggered. The selection of *anchor1* yields to *start* of node 4 starts 2s after the *end* of node2 (link l_4 with delay 2s noted Δ^2). The selection of *anchor2* yields to the immediate *start* of node5, as soon as node 3 terminated (link l_5 with a **waitLatest** logic). It is further assumed that both anchors are timed, i.e. that their activation depend on additional time constraints: they can only be triggered after *dmin* (set here to 5s) and until *dmax* (set here to 25s). Furthermore, one considers that *anchor1* is the default anchor, i.e. it triggers automatically when there is no user interaction during the time interval where both anchors are offered.
- Node3 is a content node representing an audio; its duration is initially set to 52s. The *end* of this node triggers the *start* of node8 (link l_8).
- Node4 is a content node representing a video; its duration is set to 20s.
- Node8 is a content node representing an image; its minimal duration is set to 5s, and its maximal duration is undetermined (set to infinity). In fact, the presentation of node8 will end as soon as composite node5 ends (link l_{11}); finally, the *end* of node8, causes the *end* of node1 presentation (link l_{12}).
- Composite node5 may be started 3s after the *end* of node4 or as soon as the link l_5 is triggered. As a consequence, node5 may be started with two different descriptors depending on the event which triggered its start; these descriptors have an impact on the internal structure of composite node5, i.e. the type of presentation for content node6 (either an audio or a text) and their associated duration. Node 7 contains a video, with a minimal duration of 10s.

- The *end* of node6 leads to the *end* of node7, which itself leads to the *end* of composite node5, which causes the *end* of node8.

3. Translation from the high-level model into a RT-LOTOS specification

The purpose is not to define a new formalism for the description of temporal structures in hypermedia documents. Instead, our goal is to formalize hypermedia objects defined in some high-level authoring model like NCM via a mapping of these objects onto a general-purpose formal description technique (FDT in short). The FDT is therefore completely hidden to the document authors. Having an FDT document specification, verification techniques, such as reachability analysis and model-checking, can be applied for analyzing application oriented properties of the design and consistency properties in particular.

How to perform such mapping is the prime question. The selected FDT should have the following features in order to be useful as well as efficient:

- It must have a formal semantics, and not only an intuitive one.
- It should be based on compositions, in the sense that complex document structures should be expressed by general constructs.
- It should be able to express non-deterministic behaviors, particularly interactions with the external environment (i.e., the users).
- It should have the ability to express complex time-constrained behaviors.
- It should be mature enough, and supported by software tools for automating the verification procedure.

Process algebra meet the first three requirements and LOTOS becomes a strong candidate because it is an international standard. Since standard LOTOS is not able to express time-constrained behaviors, different extension proposals have been made within the international LOTOS community, and are currently being standardized. Among these proposals, we selected RT-LOTOS (Real-Time LOTOS) proposed at LAAS-CNRS, in particular because the RTL software tool is available and operational (Courtiat, 1995). The same approach may easily be adapted to the new emerging E-LOTOS standard once stabilized with an adequate tool support.

3.1. The specification methodology

The approach developed in the paper addresses the formal semantics of the basic objects of a hypermedia document by providing a mapping between these objects and RT-LOTOS processes, and another mapping between object composition rules and RT-LOTOS process parallel compositions. The approach relies on general mapping rules, as well as on the definition of RT-LOTOS process libraries specifying the behavior of reusable basic hypermedia objects, like content nodes, links and anchors. The approach is therefore general and may possibly be fully automated within an authoring system. As previously mentioned, an experimental implementation is

currently being developed for the HyperProp authoring system. Even without any full automated translation procedure currently available, various RT-LOTOS specifications of documents authored in NCM have been produced using this general translation procedure which is currently being put in practice by means of templates.

3.2. Hypermedia objects specification

The overall architecture of the specification is presented in Figure 2, where the specification of a hypermedia document (process *Hypermedia*) is composed in parallel with a process representing the environment (process *Users*).

Process *Hypermedia* is itself specified as the instantiation of a process specifying the document top level (composite) node, synchronized with a parallel composition of the processes describing the links defined for the nodes. These process models timed synchronization relationships among all components.

```
...
hide user, start, end, trigger, endRequest in
   Hypermedia[user, start, end, trigger, endRequest] |[user]| Users[user]
where
process Hypermedia[user, start, end, trigger, endRequest] : exit :=
   compositeNode[user, start, end, trigger, endRequest] (...)
      |[start, end, trigger, endRequest]|
         ( ... parallel composition of the different synchronization links ...)
endproc
process Users[user] : exit :=
         ( ... specification of the user interactions ... )
endproc
```

Figure 2 – Specification overall architecture.

The basic structure in a hierarchical composition model of documents is the Composite Node. This general structure is described by the compositeNode RT-LOTOS process definition as presented in Figure 3.

Process *compositeNode* has three formal parameters characterizing respectively the node ID, and its minimal and maximal presentation duration. It features five external gates: gates *start* and *end* associated with the start and end of the node presentation; gates *user* and *trigger* related to the capture of user interactions and to the trigger of an anchor, respectively; gate *endRequest* related to the termination of the node. All these gates are parameterized in order to identify the node ID and the media type (for gates *start*, *end* and *endRequest*), the anchor ID (for gates *user* and *trigger*).

When process *compositeNode* is instantiated with some *ID* value and other parameters, the process first synchronizes on gate *start*. Using the recursive definition of compositeNode, one may note that the composite node may be re-instantiated again. Once the synchronization on *start* is performed, the behavior of the process is essentially described by process *bodyNode* together with the specification of the termination conditions of the composite node (see process *endNode* in Figure 3).

Process *bodyNode* permits to select the body of the current composite node with respect to the node ID; assuming some generic node identifier *i*, the associated bodyNode process is *bodyNodei*; it describes the internal logical structure of composite node *i*; it is made up by the parallel composition of the RT-LOTOS

processes characterizing content nodes and/or other composite nodes nested within composite node *i*.

```
process compositeNode[start, end, user, trigger, endRequest, grant]  (ID, dmin, dmax : nat) : exit :=
let composite : nat = 0 in
start !ID!composite;
 ( ( bodyNode[start, end, user, trigger, endRequest, grant](ID)
    [> endNode[endRequest,end] (ID,dmin,dmax) )
            ||| compositeNode[start, end, user, trigger, endRequest, grant] (ID, dmin, dmax) )
where
process bodyNode[start, end, user, trigger, endRequest, grant] (ID:nat) : exit :=
            ...
            [ID == i] -> bodyNodei[start, end, user, trigger, endRequest, grant]
            ...
endproc
process endNode[endRequest,end] (ID,dmin,dmax : nat) : exit :=
let composite : nat = 0 in
            delay(dmin) (endRequest!ID!composite; end!ID!composite; exit)
                        |[end]| end{dmax-dmin}!ID!composite; exit)
endproc
...
process bodyNodei[start, end, user, trigger, endRequest, grant] : exit :=
  ... ||| contentNode[start, end, user, trigger, endRequest, grant] (...)
  ||| ... ||| compositeNode[start, end, user, trigger, endRequest, grant] (...) ||| ...
endproc
process contentNode[start, end, user, trigger, grant] (ID, dmin, dmax, media : nat) : exit :=
    start!ID!media;
    grant!media?dstart:nat!true; delay(dstart) (( delay(dmin) end{dmax-dmin}!ID!media; exit )
        ||| contentNode[start, end, user, trigger, grant] (ID, dmin, dmax, media) )
endproc
```

Figure 3 - compositeNode and contentNode generic process definitions.

The general structure of a content node is described by the contentNode RT-LOTOS process presented in the bottom of Figure 3. This process is much simpler than the previous composite node specification, since there are no more nodes nested within a content node. The content node specification forces the termination of the node according to the minimal and maximal duration defined for this node. As previously for the composite node, one can note that process contentNode may be defined recursively.

As suggested by the excerpts of the formal specification described in, the approach did not try to get the most readable formal RT-LOTOS specification, but instead the most general (with respect to the expressive power of the NCM model) and generic, based on reusable components. The goal has been to completely automate the derivation of the formal specification from an NCM authoring. As a consequence, our specification methodology is based on the identification and formalization of several process libraries characterizing, basic content nodes, basic links, basic composition of nodes and links and basic termination conditions respectively. Some components of these libraries have been inherited from previous work on a simpler hypermedia synchronization model (Courtiat, 1996).

4. Consistency verification

Using formal methods within the context of a high-level authoring tool brings two main advantages. First, it provides a formal semantics to the high-level authoring model, describing without any ambiguity the behavior of a document presentation. Second, it permits to check consistency properties on the formal specification derived from the authoring model, using standard verification techniques.

4.1 Reachability analysis of RT-LOTOS specifications

The verification techniques developed and implemented for RT-LOTOS within the RTL software tool (Courtiat, 1995) consist in applying reachability analysis to the timed automaton model into which the RT-LOTOS specification is compiled. The method implemented in RTL is characterized by two important features:

- It permits to minimize the number of clocks in each control state of the timed automaton, thanks to the definition of the DTA (Dynamic Timed Automaton) model.
- Reachability analysis may be performed on the fly during the DTA generation.

The resulting graph is a minimal reachability graph. Each node of the graph represents a reachable class which may include an infinite number of elements depending on the value of the current time. Each arc of the graph corresponds either to an action occurrence (*i.e.* an event) or to a global time progression (in this case, the arc is labeled by action *time*).

RTL tool has further been interfaced with other tools: BCG for labeled transition systems display, and ALDEBARAN (Fernandez, 1996) for deriving a minimal automaton from a labeled reachability graph with respect to an equivalence relation (for instance, the observational equivalence).

4.2 Consistency properties

Hypermedia documents are expected to satisfy temporal consistency properties stating that temporal synchronization constraints to be met during the document presentation are not in conflict with one another. Depending on how these synchronization constraints are defined, there exists indeed a risk to create inconsistent situations, *i.e.* situations where different contradictory synchronization requirements cannot be satisfied together, leading to undesirable deadlocks (global or partial) during the document presentation.

By definition (Courtiat, 1996), we consider that a document presentation is consistent[2] if the action characterizing the *start* of the document presentation is necessary followed, some (finite) time later, by an action characterizing the *end* of the document presentation (Property *P1*). As a consequence, action *end* should be always reachable from the initial state of the RT-LOTOS minimal reachability graph, assuming that, by construction of the RT-LOTOS specification, action *start* is the unique action to be enabled in the initial state. The definition also implies that there is no divergence (i.e. no undesirable loop) within the reachability graph.

The above definition is related to a document presentation. Within the framework of our authoring model, the definition may also be applied to a composite node presentation.

In order to distinguish different consistency properties, let us call *P1* an *intrinsic* consistency property (Santos, 1998). It is said to be intrinsic because it is independent of any particular multimedia resource used for the document presentation. Many reasons may lead to temporal intrinsic inconsistencies, when one uses a high-level and powerful authoring model like NCM. For examples which seemed a priori not too complex, the following situations were encountered: 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 synchronization links, omission of some default cases (for instance, what happens for a node featuring several anchors, when none of them is triggered by the user: should the node terminate or not? What the impact on other nodes?)

Another interesting consistency property (called *extrinsic* consistency property, or consistency property *P2*) to be proved when a document presentation is related to potential undesirable concurrent accesses to some multimedia presentation resources (Santos, 1998). A simple example includes an erroneous attempt to mix different audio signals on the same audio channel (sound card + loudspeakers). A complete specification of these consistency properties would probably require to define a high-level RT-LOTOS specification of the multimedia platform on which the document is supposed to be presented and to compose this specification with the RT-LOTOS specification of the document. Such a study is being carried out at LAAS-CNRS, but it is not reported in this paper. We will see however how some extrinsic consistency properties may be proven without developing any formal model of the underlying multimedia platform.

4.3 Verification approaches

Two complementary approaches have been investigated in order to prove consistency properties: global reachability analysis and node aggregation.

4.3.1. Global reachability analysis. This classical approach consists in performing the reachability analysis on the RT-LOTOS specification derived from the entire document. Its main limitation is related to the state space explosion problem. If one nevertheless succeeds in building the reachability graph, the latter can be transformed into a labeled transition system (LTS) whose observable events are appropriately selected with consistency properties verification in mind (typically, the *start* and *end* of the document or node, the *time* progression action, the *user* interactions). Other events may *de facto* be hidden. The LTS can be minimized using, *e.g.*, observational or safety equivalence relations, which are both supported by the ALDEBARAN tool.

4.3.2. Node aggregation. Validation by node aggregation is bottom up. We first select a composite node and perform reachability analysis on it. The resulting graph together with the minimized automaton output by ALDEBARAN may be used to prove, on this composite node, the required consistency property. Once the consistency of the composite node has been established, one knows, according to the

consistency definition, that the occurrence of the *start* action of the composite node is followed some (finite) time later by the occurrence of the composite node *end* action. The idea is then to replace the composite node specification by a content node specification whose minimal (*dmin*) and maximal (*dmax*) duration are directly derived from the internal timing logic of the composite node. We sketched an algorithm which derives dmin and dmax information from the composite node global reachability graph. This algorithm relies on the reachability graph clock regions and the analysis of the enabled transitions, taking into account the general structure of the composite node specification. However, we did not get any proof of the algorithm correctness, but experimented successfully its "supposed" validity on different examples. Once we get this timing information, we come up with the following conjecture: any composite node may be replaced by an "abstract" content node whose *start* and *end* actions are similar to the composite node's *start* and *end* actions, and whose (possibly non deterministic) global delay is computed from the composite node's internal timing logic. The required congruency has not been formally proved, but seems to be a direct consequence of the rather particular structure of a consistent composite node (one *start* action, followed by several branching actions which are hidden within the composite node and without any divergence, leading finally to the unique *end* action). Further work is required in this respect, but the aggregation method appears to work well in practice, since it permits to analyze complex structured hypermedia documents composite note by composite node instead of globally.

4.4 Illustrations

In this section we survey some verification results achieved for the hypermedia document presented in the section 2.4.

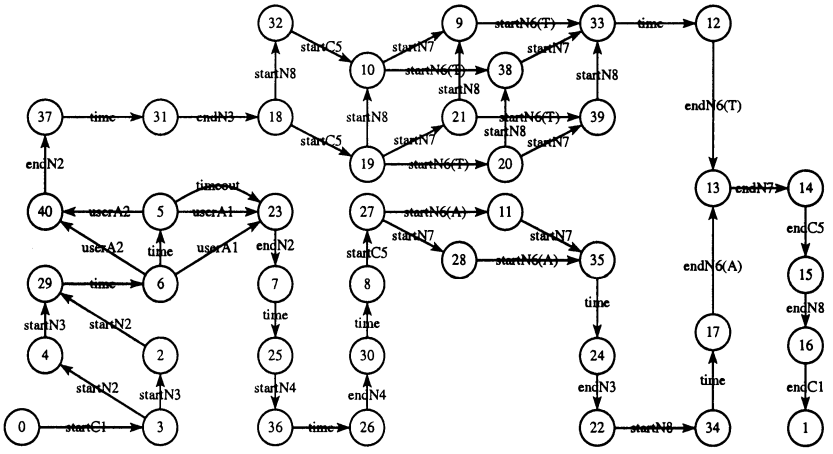

Figure 4 – Document minimal automaton.

Figure 4 presents the minimal automaton[3] of the whole document reachability graph, where only actions *start* and *end* of all nodes (composite or not), *user* (for anchors selection), *timeout* (for the automatic trigger of *anchor1*), *time* (for time

progression) have been selected as visible actions. The graph shows that there is no visible loop, and that action *startC1* (i.e. the start of compositeNode1) leads eventually to *endC1*. As a consequence, the document is intrinsically consistent. The fact that there is no hidden internal loop is a direct consequence of the RT-LOTOS semantics. The invisible internal actions are indeed urgent actions, and a loop among these actions will stop the time progression; now, one can note that any *start* action is always followed by a time progression action before achieving the corresponding *end* action.

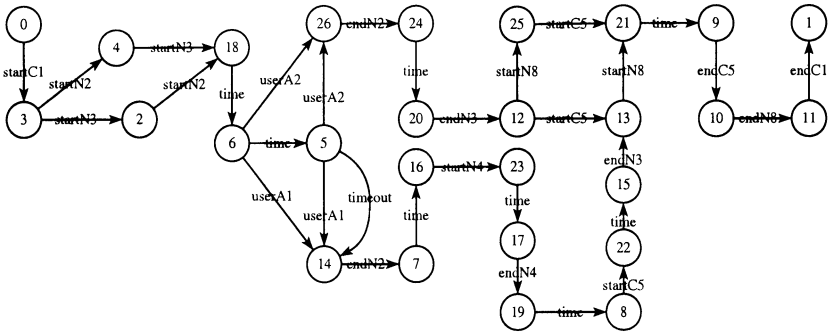

Figure 5 – Document minimal automaton (with node aggregation).

Figure 5 presents the minimal automaton of the global reachability graph which has been derived from the document RT-LOTOS specification, where composite node5 has been replaced by an abstract content node (following the aggregation method discussed previously). This graph is observationally equivalent to the minimal automaton derived from the graph of Figure 4, with actions *startN6(A)*, *startN6(T)*, *startN7*, *endA6(A)*, *endN6(T)*, and *endN7* being invisible.

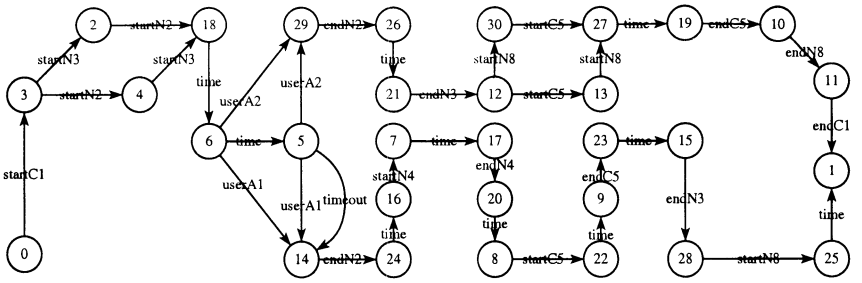

Figure 6 – Inconsistency situation (intrinsic).

Figure 6 presents the same graph as Figure 4 with the difference now that the duration of node3 has been set to 100s instead of 52s. One can see that such a specification leads to an intrinsic inconsistency since the *start* of node8 (action *startN8*) may occur after the *end* of node5 (action *endC5*); and yet, we specified that the *end* of node5 should lead immediately to the *end* of node8, hence the inconsistency since action *endC1* may be not reachable from the initial state.

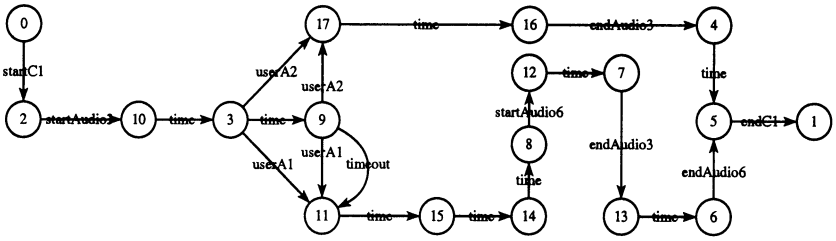

Figure 7 – Inconsistency situation (extrinsic).

Figure 7 considers the same initial case previously analyzed in Figures 4 and 5. The minimal automaton is constructed considering as visible only the actions starting and ending presentations of audio nodes (either audio content nodes or composite nodes including an audio node), plus the *user*, *timeout* and *time* actions. Analyzing this automaton, one can see that the document specification does not satisfy consistency property *P2* (see the concurrent access to an audio channel), since for some behaviors depending on the time at which the user has selected an anchor, the start of an audio presentation is followed by the start of another audio presentation, without the end of the previous presentation; hence an inconsistency, leading to an undesirable mix on the same audio channel of two audio presentations. For the construction of this automaton and for clarity purpose, *startN3* (respectively *endN3*) has been relabeled *startAudio3* (respectively *endAudio3*), and *startN6(A)* (respectively *endN6(A)*) has been relabeled *startAudio6* (respectively *endAudio6*).

This extrinsic inconsistency may be corrected by different ways: either by decreasing node3 duration, by modifying the version of node6 considering a textual presentation instead of an audio, or more probably by adding a new link in the document specification. Thus one can see that error corrections are made at the level of the authoring model, which are then translated into the RT-LOTOS formal specification, following the translation procedure of section 3.

5. Conclusions

As a conclusion, one can stress the following points. First, it is extremely easy and flexible to work with high-level models at the authoring stage. Second, it is easier for the author to handle document logical structuring than only document presentation structuring. Hence, models based on compositions allowing every type of relationships among their components become very important. Third, the use of a hidden formal description technique, like RT-LOTOS, brings two important advantages: (i) it provides a formal semantics for high level hypermedia models, permitting a better understanding of some critical behaviors that may be expressed by the model (ii) it allows applying, when possible, exhaustive verification techniques to check a document against design errors. More work is currently devoted to the study of the extrinsic properties, taking into account additional delays that may be forced by an underlying multimedia platform. The limitation of the proposed method is related to the state space explosion of the reachability graph. The definition of the

aggregation method, which remains to be formalized and proved, appears to be an important step towards the verification of complex structured documents.

Endnotes

1. Only temporal constraints are considered in this paper; note however that some ideas may probably be generalized to spatial synchronization.
2. For simplification purpose, we do not consider in this definition the case where one wants, in the middle of a document presentation, restart the whole presentation; in that case, a start of a document could be followed by another start, and so on.
3. All the minimal automata referred to in this paragraph are related to the observational equivalence.

Acknowledgments

The work reported in this paper is funded by a CNRS research grant (Action Télécoms). The first author is supported by a grant from the Brazilian Government (CAPES).

References

Blakowski G.; Steinmetz R. (1996) A Media Synchronization Survey: Reference Model, Specification and Case Studies, *IEEE Journal on Selected Areas in Communications*, 24(1).

Buchanan MC.; Zellweger PT. (1993a) Automatically generating consistent schedules for multimedia documents, *Multimedia System Journal.*

Buchanan, MC; Zellweger, PT. (1993b) Automatic Temporal Layout Mechanisms. *ACM Multimedia'93.* San Francisco, USA.

Casanova, M.A. et all. (1991) The Nested Context Model for Hyperdocuments. *ACM Conference on Hypertext*, San Antonio, USA.

Courtiat, JP.; De Oliveira, RC. (1995) A Reachability Analysis of RT-LOTOS Specifications. *FORTE'95*, Montreal, Canada.

Courtiat, JP.; Oliveira, RC. (1996) Proving Temporal Consistency in a New Multimedia Synchronization Model. *ACM Multimedia'96.* Boston, USA.

Fernandez JC. et all. (1996) CADP, CAESAR/ALDEBARAN Development Package: a protocol validation and verification toolbox, *8th Conference on Computer Aided Verification*, New Brunswick, USA.

Fraissé, S.; Nanard, J.; Nanard M. (1996) Generating Hypermedia from Specifications by Sketching Multimedia Templates. *ACM Multimedia'96*, Boston, USA.

Lakaida N.; Keramane C. (1995) Maintaining temporal consistency of multimedia documents. *ACM Workshop on Effective Abstractions in Multimedia*, San Francisco, USA.

Rodrigues, R.F.; Soares, L.F.G.; Souza, G.L. (1997) Authoring and Formatting of Documents Based on Event-Driven Hypermedia Models, *IEEE Conference on Protocols for Multimedia Systems, Multimedia Networking — PROMSMmNet'97*, Santiago, Chile.

Santos, C.A.S; Soares, L.F.G; Souza; G.L; Courtiat, J.-P. (1998) Design Methodology and Formal Validation of Hypermedia Documents. *ACM Multimedia'98*, Bristol, UK. (To be published)

Soares, L.F.G.; Casanova, M.A.; Rodriguez, N.L.R. (1995) Nested Composite Nodes and Version Control in an Open Hypermedia System, *International Journal on Information Systems, Special Issue on Multimedia Information Systems.* 501-519.

Van Rossum et all. (1993) CMIFed: A Presentation Environment for Portable Hypermedia Documents, *ACM Multimedia'93*, California, USA.

Wahl T., Rothermel K. (1994) Representing time in multimedia systems. *IEEE Int. Conf. on Multimedia Computing and Systems*, Boston, USA.

Willrich, R.; Sénac, P.; Saqui-Sannes, P.; Diaz, M. (1996) Hypermedia Documents Design Using the HTSPN Model. *3rd Int. Conf. on MultiMedia Modeling - MMM'96*, Toulouse, France.