Modelling Multimedia Synchronization using a Time Petri Net Based Approach

Abdelghani Ghomari[1] and Chabane Djeraba[2]
[1]Department of Computer Science, University of Es-Senia Oran
[2]University of sciences and Technologies of Lille 1
[1]Algeria
[2]France

1. Introduction

Multimedia refers to the presentation of collections of both static and dynamic data (i.e., data with natural time dependencies e.g., audio or video) in a specified order and time. Therefore, their mutual synchronization must assure a proper temporal order of presentation events. Multimedia synchronization can be defined as a mutual assignment of data items and time instants. These time instants may be known in advance (e.g., standard consumer data players) or they can be also results of some unknown function of time (event driven synchronization) or known with some limited accuracy (e.g., random network delays).

The modelling and the presentation of multimedia scenarios are challenges of multimedia applications. Multimedia scenarios are results of temporal composition and user interactions of multimedia objects in an application domain, and lot of works discussed this notion (Adjeroh & Lee, 1995; Perez-Luque & Little, 1996). Temporal compositions consist in presenting multimedia objects which requires synchronization among different media.

Most of specification models are based on Allen's relations (Allen, 1983). Allen defined seven basic relations between two temporal intervals. For example, a TV program starts at 9:00 pm, and finishes at 11:00 pm. The TV program can be considered as one of multimedia objects. In addition, "interval" is considered as a range from 9:00 pm to 11:00 pm, and "duration" as two hours. Allen's relations require this duration of the interval. Before designing the specification model, interval duration must be known. This means that multimedia database systems must determine duration of multimedia objects, because presentations are almost dependent of duration.

Our approach defines a tool that has two important inter-dependent features:

User temporal specification based on an appropriate temporal specification language, which is itself based on an extension of Allen's temporal relations (Allen, 1983). This extension models both existing temporal arrangement and dependency relations between multimedia objects, and this is an interesting point of our work.

Automatic generation of a time Petri net based on the previous temporal specification. The time Petri net is stored in an object called "scenario object". The user may request the simulation and the interpretation of the scenario object which leads to scenario

presentations with domain expert interactions. The Petri net permits a formal specification and a proof of scenarios. The simulation and the proofs are two advantages of the Petri net.

A first version of our approach (Ghomari & Djeraba, 2003) considers multimedia objects of known or unknown duration and interactive relations, but doesn't consider dependency temporal relations between multimedia objects and the management of multimedia scenario in a database system. This is the main differences between the first version of our approach and the second one that will be described in this chapter. This approach provides the following benefits (Ghomari & Djeraba, 2010):

1. The ability to deal with non-deterministic time intervals, e.g. objects with an unknown duration, objects whose reproduction can fail and objects that represent user interactions.
2. The possibility of automatic detection of inconsistent synchronization conditions such as "A *precedes* B, B *precedes* C, C *precedes* A".
3. A graphical notation to describe and simulate the presentation.
4. An editor which abstracts the internal Petri net representation and allows the user to think in familiar terms such as "*precedence*" or "*overlap*".
5. Automatic generation of a MP-RdPT net based on the previous temporal specification.
6. Automatic analysis of the multimedia scenarios properties, like: safeness, liveness, reversibility and consistency.

In this chapter, we highlight the following points: related works (Section 2), our scenario temporal specification (Section 3), the multimedia p-time Petri net (Section 4), object oriented database modelling of the multimedia specification (Section 5), and the architecture of the system prototype (Section 6).

2. Related work

Existing temporal models for multimedia may be decomposed into two classes: instant-based and interval-based (Hamblin, 1972; Blakowski & Steinmetz, 1996). In instant-based models, the elementary units are instants in a time space. Each event in the model has its associated time instants. The time instants arranged according to some relations such as *precede*, *simultaneous* or *after* form complex multimedia presentations. An example of the instant-based approach is timeline, in which media objects are placed on several time axes called tracks, one per each media type. All events such as the beginning or the end of a segment are totally ordered on the time line.

Several approaches support instant-based models such as Hy-Time (ISO, 1992). The model is well suited for temporal composition of media segments of known durations; however it falls short for unknown durations.

Interval-based models consider elementary media entities as time intervals ordered according to some relations. Existing models are mainly based on the relations defined by Allen for expressing the knowledge about time (Allen, 1983). However, using Allen's relations for multimedia composition faces several problems. First, the relations were designed to express existing relationships between intervals of fixed duration and not for specifying relationships that must be always satisfied even when interval durations are changed. Consider for example an existing relation *before* between intervals a and b (see Fig. 1). When we increase the duration of interval a, the relation changes from *before* to *during* passing through intermediate relations *meets*, *overlaps*, and *finishes*. This drawback makes the Allen relations not suitable for specifying composition of intervals with unknown duration.
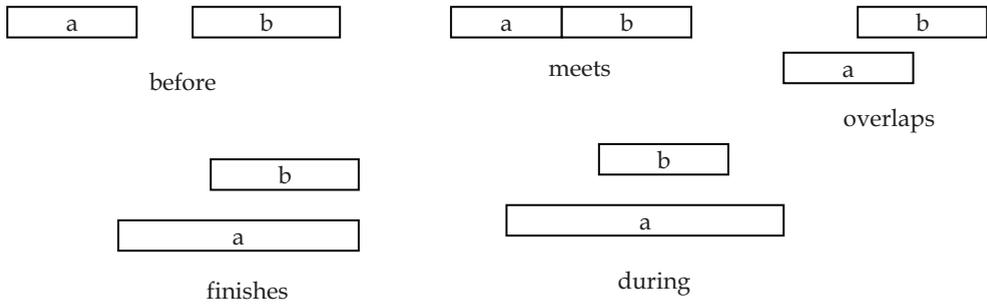
Fig. 1. Relations change when the interval duration is modified

Another problem with the Allen's relations is their descriptive character. They allow expression of an existing, a posteriori arrangement of intervals, but they do not express any causal or functional relation between intervals. For example, relation *meets* only states that the end of the first interval coincides with the end of the second one, but it does not say whether the first interval starts the second one, whether the second interval stops the first one or whether it is a pure coincidence (see Fig. 2). So, the Allen's relations can be useful for characterizing an existing, instantiated presentation (a presentation for which all start and termination instants of media segments are known).
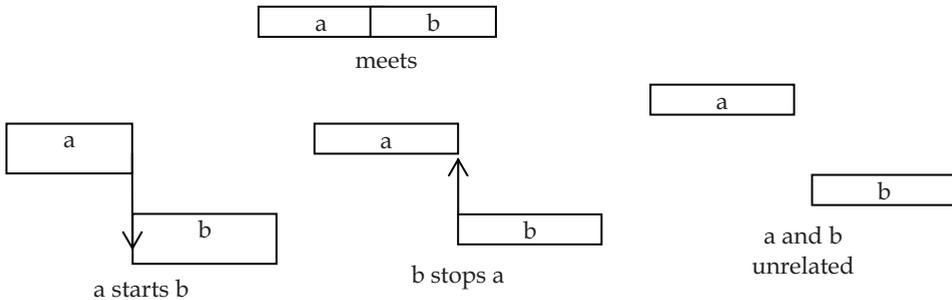

Fig. 2. Meets represents temporal coincidence not a functional relationship.

The third problem with the Allen's relations is related to inconsistent specifications that can be introduced in a multimedia presentation. Detecting inconsistent specification requires algorithms of complexity $[O(N^2)]$, where N is the number of intervals (Allen, 1983).

Many approaches are based on time interval-based model. For example:

- King (King, 1994), proposes a different formalism based on a temporal logic. He/She shows how the Allen's relations can be expressed using temporal logic formulae. Although his formalism has solid mathematical bases, composition of multimedia presentations using declarative formulae is awkward. Logic formulae are difficult to use by any author unaware of formal methods. Moreover, to be useful, the formalism must be supported by a consistency checker and an interpreter to execute a given temporal specification.

- Courtiat and De Oliveira (Courtiat and Oliveira, 1996), presented a synchronization model for the formal description of multimedia documents that consider network

performances. This model automatically translates the user formalization into a real-time LOTOS formal specification and verifies a multimedia document aiming to identify potential temporal inconsistencies.

- Blakowski and Steinmetz (Blakowski & Steinmetz, 1996), recognized an event-based representation of a multimedia scenario as one of the four categories for modelling a multimedia presentation. Events are represented in the Hypermedia/Time-Based Structuring Language (HyTime) and Hypermedia Office Document Architecture (HyperODA). Events are defined in HyTime as presentations of media objects along with the playout specifications and finite coordinate system (FCS) coordinates. HyperODA events happen instantaneously and mainly correspond to start and end of media objects or timers. This approach suffers from poor semantics conveyed by the events and moreover it does not provide any scheme for composition and consumption architectures.

Another group of approaches consider the PN tool for multimedia synchronization. For example:

- Object Composition Petri Net Model (OCPN) (Little and Ghafoor, 1993). However, the OCPN approach does not inherently support modelling of interaction.
- Hierarchical Time Stream Petri Net (HTSPN) Model (Senac et al., 1995). However, in the HTSPN approach, it is unspecified what would happen to tokens representing streams left behind the synchronization. If those "dead tokens" were remained in the state, a semantic inconsistency between the model and the real system would be produced.
- Multimedia Organization Employing a Network Approach (MORENA) (Botafogo and Moss, 1995). However, since MORENA does not fully follow PN theory, it may not be easy to analyse the output model using the existing techniques developed in the Petri net field.
- High-Level Petri Net-Based Hypermedia System (Na Cheo and Furuta, 2001). However, the authoring environment used to specify multimedia scenarios with coloured timed Petri nets is not user-friendly, like in the user community.
- Dynamic Fuzzy Multimedia Petri Net (DFMN) (Chen and Huang, 2005). However, in the approach of Shterev (Shterev, 2005) using the DFMN model for modelling of interaction on multimedia streams and objects, none analysis or simulation tools of multimedia presentation are provided.

Special attention should be paid to the upcoming standard Synchronized Multimedia Integration Language (SMIL) (W3C, 2001). SMIL is a meta-language for authoring and presentation of multimedia documents. However, SMIL is a script language and does not support analysis to the same degree that Petri-net-based systems do.

An interesting survey on authoring models and approaches are presented in (Jourdan et al. 1997; Roisin & Sèdes, 2004; Boronat et al., 2008).

Our approach is a compromise between formal specification useful for checking inconsistency and high level language for user specification.

3. Our scenario temporal specification

We will present a model for temporal composition of multimedia objects. The model is based on time-interval and Weiss relations (Weiss et al., 1995). We consider the seven relations of Allen (Allen, 1983) (*equals, meets, before, finishes, starts, overlaps, during*) with the following features:

Firstly, the temporal relations are designed to specify relations between multimedia objects of both determined and undetermined duration. Secondly, the temporal relations describe both existing arrangement of multimedia objects, and dependency relations between multimedia objects. For example, x *meets* y means that the end of multimedia object x coincides with the start of multimedia object y, but it doesn't describe whether multimedia object x starts multimedia object y, or whether multimedia object y stops multimedia object x. Thirdly, the detection of inconsistent specification is not necessary.

3.1 Interval

Our elementary entities are time intervals. Time interval I is defined by the end points (I.begin ≤ I.end) as I = {t | I.begin ≤ t ≤ I.end }. The duration of interval I is d = I.end – I.begin and can be constant (e.g. 5 seconds), dependent on the intrinsic playing time of the medium (e.g. playing time of a video segment) or unspecified (e.g. user interaction or live feed). In this paper each interval corresponds to the presentation of one object (e.g. an image or a music selection). In that sense, the beginning and the end of an interval are logical times which will really correspond to physical time during the effective presentation to the user.

3.2 Temporal relations

Several relationships have been defined on time intervals: *equals, before, meets, finishes, starts, during, overlaps*, (Allen, 1983). Usually, they are binary relationships but can be easily extended to n-ary ones (Little & Ghafoor, 1993). Sequential relationships combine intervals which share the same timeline (mutual exclusion), occurring one after the other with (before) or without delay (meet) between them. Parallel relationships relate intervals which have their own timeline. In our model these relations are used for composing and synchronizing multimedia objects in presentations.

Temporal aspects are also discussed in standards like HyTime (ISO, 1992) where the concept of finite coordinate system is used to define a set of axes of finite dimensions. The system designer defines both the number of axes and the units of dimension used along each of them. Hence, an x-y-time coordinate system can be used to model spatial as well as temporal aspects for database objects belonging to presentations.

A program of our temporal specification language is divided into four parts: declaration, assignation, temporal and interactive relations.

Declaration: the declaration part contains the declarations of multimedia objects and returns as a result: multimedia-object (min, opt, max): media-type.

Assignation: the assign part contains the assign functions between the objects declared in the first part and the data streams. For example, the data streams may be mpeg or jpeg when storing video or images objects respectively. For example: assign (video1, "file1.mpeg"); or assign (image1, "file2.jpeg").

When using several equal multimedia objects of the same media, we have to declare several multimedia objects with the same duration assigned to the same physical support. For example, if video3 and video4 share the same physical object with the same duration, we will have: video3 (min, opt, max): VIDEO; video4 (min, opt, max): VIDEO; assign (video3, "file3.mpeg"); assign (video4, "file3.mpeg").

When using several multimedia objects of the same media with different duration, we have to declare several multimedia objects with different duration assigned to the same physical support. For example, if video3 and video4 share the same physical object with different

duration, we will have : video3 (min1, opt1, max1) : VIDEO; video4 (min2, opt2, max2) : VIDEO; assign(video3,"file3.mpeg");assign(video4,"file3.mpeg".

The temporal relations part contains a set of temporal relations, each one representing a binary Allen's relation between multimedia objects. These multimedia objects are either declared objects and assigned to physical supports, or objects resulting from temporal relations. A relation takes two multimedia objects as arguments and returns a multimedia object as a result: temporal-relation (multimedia object1, multimedia object2) -> multimedia object3. The resulting multimedia object may be used as an argument of another temporal relation. Example: *equal (start (meets (video1, video2)*, *meets (text1, text2)), image1)*. The Fig. 3 summarizes the Backus-Naur Form (BNF) of the grammar of our temporal specification language.

```
<Scenario> ::= Scenario <ScenarioName> : Idf elements <Declaration_assignation> <Var1> Relations
<Interactive _temporal_relation> End.
<Declaration_assignation > ::= <Declaration > | <Declaration >; <Scenario>
<Declaration> ::= <Declaration> | <Assignation>
<Declaration> ::= idf (min, opt, max) : <Type> : Path ; <Post_decl>
<Post_decl> ::= <Declaration> | λ
<Assignation> ::= Assign (Idf, Path); <Post_assign>
<Post_assign> ::= <Assignation> | λ
<Var1> ::= Variables <Var2> | λ
<Var2> ::= <Mise> <Post_mise>
<Mise> ::= idf ::= <Temporal_interactive_Relation >
<Post_mise> ::= <Var2> |λ
<Type> ::=  Audio | Video | Image | Text | Button
< Interactive _temporal_relation > ::= (<Temporal _Relation >|<Interactive_Relation > ); <Post_ Interactive
_temporal_relation >
<Temporal_Relation> ::= <Relation> <Post_relation> |<NomScenario>
<Relation> ::= <Equal> | <meet> | <before> | <begin> | <during> | <finish> | <overlap>
<Post-relation> ::= <Temporal_Relation> | λ
<equals> ::= equals (<Relation>, <Relation>)
<meets> ::= meets (<Relation>, <Relation>)
<before> ::= before (<Relation>, <Relation>, <Delay>)
<starts> ::= starts (<Relation>, <Relation>, <Delay>)
<during> ::= during (<Relation>, <Relation>, <Delay>)
<finishes> ::= finishes (<Relation>, <Relation>, <Delay>)
<overlaps> ::= overlaps (<Relation>, <Relation>, <Delay>)
<Interactive_Relation> ::= <Relation_int> <Post_link> | <ScenarioName>
<relation_int> ::=par-min | par-max | par-master |
<post_link> ::= <relation_int> | λ
with :
Idf = [a-zA-Z]([a-zA-Z]|[0-9])*
Path = [a-zA-Z0-9]+\. [a-zA-Z0-9]+
min, opt, max, delai = [1-9][0-9]*
λ = empty expression
```

Fig. 3. BNF form of the grammar

3.2.1 Temporal interaction

Our approach synchronizes the scenario with the user (i.e an expert of the application domain). The interaction takes the form of temporal interaction (*start, stop, pause, reverse, forward*) and browsing interactions.

Temporal interactions concern user elementary operations such as *pause/resume*, *reverse* and *forward*. In pause/resume operations, the system records the current state of presentation modelled by a p-time Petri net, and when resume operation is executed, the system loads

the amount of time that the presentation had paused, and starts the presentation again from where it stopped. The reverse operation is specified in terms of temporal skip given by the user. Example "go back 15 minutes".

When the reverse operation is requested, then the Petri net deals with objects associated with places currently being presented. If the reverse operation involves objects that are further behind a place P_i in the presentation graph, the presentation graph is traversed backward until the target object is reached. The forward operation is similar to the reverse operation.

3.2.2 Browsing interaction

In browsing interactions, the user branches out of the current presentations, so he/she effectively modifies the current presentation. Let us consider a multimedia database representing scenes from a visitor while visiting art objects in a gallery. The highlight on a spatial art object is possible through animation. When the database contains images of all possible art objects, visiting may include highlight and corresponding jumps out of the sequential nature of the sequence of images corresponding to art objects. To approach this problem, we use the hierarchical modelling capability of the Petri net representation. A place can be another Petri net. So, there is a global Petri net that is composed of sub-nets of smaller Petri nets corresponding to presentations.

Branching to different presentation graph is then equivalent to following a hypermedia relation, so the user can select a branch to any part of the global presentation graph, or follow the presentation schedule as previously defined in the current presentation graph. The entry point into a branch is represented as a hypermedia node with a link to the desired presentation. Thus for the node where there is a branch, the object represented by the Petri net place at that node is just a hypermedia button indicating a branch to different presentations.

At each branching point, our approach models a hypermedia node, in parallel to the object that the branch presentation is related to. Before the branch presentation is chosen, the hypermedia node appears as a hypermedia button, with an internal duration independent of the multimedia object duration of the button. So, if the branch is not selected by the user, the presentation represented by the sub-net will not be presented, and the Petri net associated with it will not be executed. We declare the branch node as follow:

branch-node $(0,-,+\infty)$: HYPERMEDIA;

video1 (20,30,40): VIDEO;

4. The multimedia p-time Petri net (Mp-RdPT)

4.1 Differences with other approaches

Our Petri net may be considered as a variant of the temporal Petri net developed in several works (Little & Ghafoor, 1990), (Adjeroh & Lee, 1995), (Senac et al., 1995) with these interesting features:

First, the Petri net is generated automatically on the basis of the user's temporal specifications that help him to define temporal relations naturally and simply without any considerations of the Petri net details.

Secondly, the Petri net models relations that consider both existing temporal arrangement and causal temporal relations between multimedia objects.

Thirdly, during the generation of the Petri net on the basis of the temporal specification, it is not necessary to detect temporal inconsistencies like in the current approaches based on time interval.

Fourthly, after the generation of the Petri net, the system returns, when requested by the user, the simulation of the scenario that corresponds to the Petri net generated and may detect two kinds of errors: graph design errors (i.e. a multimedia object that is declared but never used) and allocation resource errors (i.e. allocation of the same resource to several multimedia objects, it is the classical problem of mutual exclusion on a critical resource).

Fifthly, in our approach, all scenarios can be expressed and executed by using our specification language, some authors, such as (Weiss et al. 1995), say that the resulting graph becomes complicated and difficult to manipulate and to modify. In our approach, the modification is very simple, because it does not concern the Petri net, but the temporal specification which is natural and simple to use.

We think that few approaches implement these features together. We can find some features of our Petri net in a powerful Hierarchical Time Stream Petri Net model HTSPN of (Senac et al., 1995).

4.2 Formal definition of Mp-RdPT

Petri nets (PN) (Peterson, 1977) are designed to model systems with interacting concurrent components. The basic PN structure is composed of four parts: a set of places P, a set of transitions T, an input or (backward) function B and an output or (forward) function F. The input and output functions relate transitions and places. A basic PN graph is graphically represented as a bipartite directed graph, in which the circular nodes are called places and the bar nodes are called transitions. A dot in a place represents a token, and a place containing one or more tokens is said to be marked.

Allen's relations have the problem that multimedia objects expressions are dependent of multimedia objects duration. The temporal relations are designed to specify relations between multimedia objects of determined duration. Therefore, they are not appropriate for undetermined duration. If creators produce a video digest from several multimedia objects, they need to modify the temporal relations between multimedia objects after their duration changes. In order to solve this problem, the system should represent relations between multimedia objects with unknown duration. We must study the model of temporal relations independently of duration changes. PN is one of graph representations and considers multimedia objects with known or unknown duration; also, the simulation of the scenario may detect errors, such as: specification errors, graph design errors, graph configuration errors, or allocation resources errors.

The PN tool has been chosen as a tool of synchronization and analysis because PN allows modelling the dynamic behavior of multimedia scenarios that can be characterized by the qualitative properties of PN corresponding. These properties are liveness, boundedness, reversibility and consistency. In the context of a temporal synchronization modelling, a class of enhanced PN model has been developed which assign a firing delay to each place and a type of synchronization to a transition (Ghomari & Djeraba, 2003; Ghomari & Djeraba, 2010). This model is called Multimedia p-Time Petri Nets (Mp-RdPT) and it is defined as follows:

A Mp-RdPT is a tuple (P, T, B, F, M_0, IS, SYN, MP, R), where:

- (P, T, B, F, M_0) defines a PN, where P is a non empty finite set of places, T is a non empty finite set of transitions, with $P \cap T = \varnothing$, B: P x T $\rightarrow$ N is the backward function,

similarly, $F : P \times T \rightarrow N$ is the forward function, $M_0 : P \rightarrow N$ is the initial marking. As usual, we denote by $\bullet t = \{p \in P \setminus B (p, t) \geq 1\}$ the set of ingoing places and $t\bullet = \{ p \in P \setminus F(p, t) \geq 1 \}$ the set of outgoing places of a transition t. Similarly, $\bullet p = \{ t \in T \setminus F(p, t) \geq 1 \}$ and $p\bullet = \{ t \in T \setminus B(p, t) \geq 1 \}$ are the sets of ingoing transitions and outgoing transitions of a place p.

The set of markings a Mp-RdPT can reach from its initial marking M_0 will be denoted as $S (M_0)$.

- $\forall p \in P, \ \forall M \in S (M_0), M(p) \leq 1$ (Mp-RdPT is safe),
- IS: is the static interval function, IS: $P \rightarrow (Q+ \cup 0) \cup (Q+ \cup 0) \cup (Q+ \cup \infty)$,

  The IS function associates with each ingoing place a static validity time interval, where (a, n, b), associated with a place, represents respectively the earliest, the nominal and the latest firing times. The firing time of a place is a timing interval during which the newly created tokens are valid to fire a transition.
- SYN is the synchronization function that defines the firing rule associated to a transition.
- SYN: $T \rightarrow Rules$, with Rules $=_{def} \{strong\text{-}or, weak\text{-}and, master\}$, the set of synchronization rules. This synchronization semantics defines synchronization instants from a place statically or dynamically chosen.

  MP is the function which indicates the master place of each transition from which the rule of transition requires a master, defined by: MP : $T_{master} =_{def} \{t \setminus SYN(t) = master \}$,

  - The strong-or synchronization rule is driven by the earliest media. If either one of the two media objects finishes, the other one has to stop, and $[Min(a_i), Min(b_i)]$ is the sensibilisation interval.
  - The weak-and synchronization rule is driven by the latest media. All the media objects are presented completely and $[Max (a_i), Max (b_i)]$ is the sensibilisation interval.
  - The master synchronization rule is driven by the master media. If two multimedia objects are presented simultaneously, when the higher priority media finishes, the other has to stop.

The multimedia presentation continues after that, and $[a_m, b_m]$ is the sensibilisation interval, with P_m indicating the master place.

We define a_m, b_m by: let MP $(t) = p_m$ and IS $(p_m) = [a_m, b_m]$.

- R: $P \rightarrow \{r_1, r_2 \ldots r_n\}$, a mapping from the set of places to a set of resources (e.g., audio/video card, processor, virtual memory, and others operating system resources).

4.3 Mp-RdPT generation

To create the temporal Petri net, each temporal relation is associated with a Petri net as illustrated by (Hamblin, 1972), and modelled in several approaches, such as in OCPN (Little and Ghafoor, 1990). This mapping is helpful for automatic creation of a time Petri net. In the Fig. 4, T_α, T_β, T_δ model respectively the duration of places P_α, P_β and P_δ.

4.4 Rules of translations

The created Mp-RdPT net is then translated to an equivalent t-time Petri net (Merlin and Farber, 1976) for analyzing by the tool Tina (Berthomieu et al., 2004). For this, we use three rules of translation (see Fig. 5) inspired from (Senac et al., 1995).
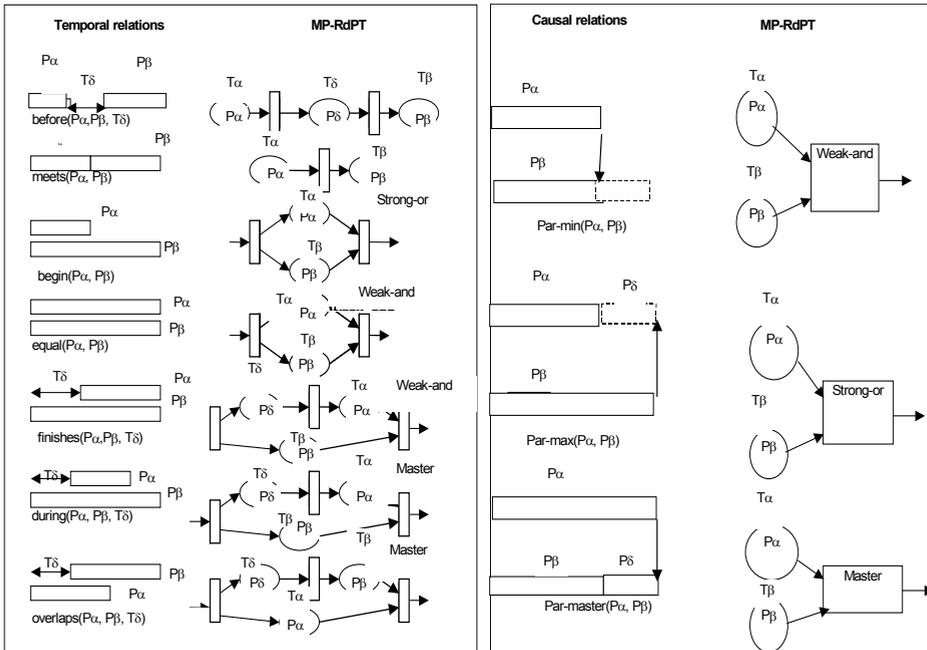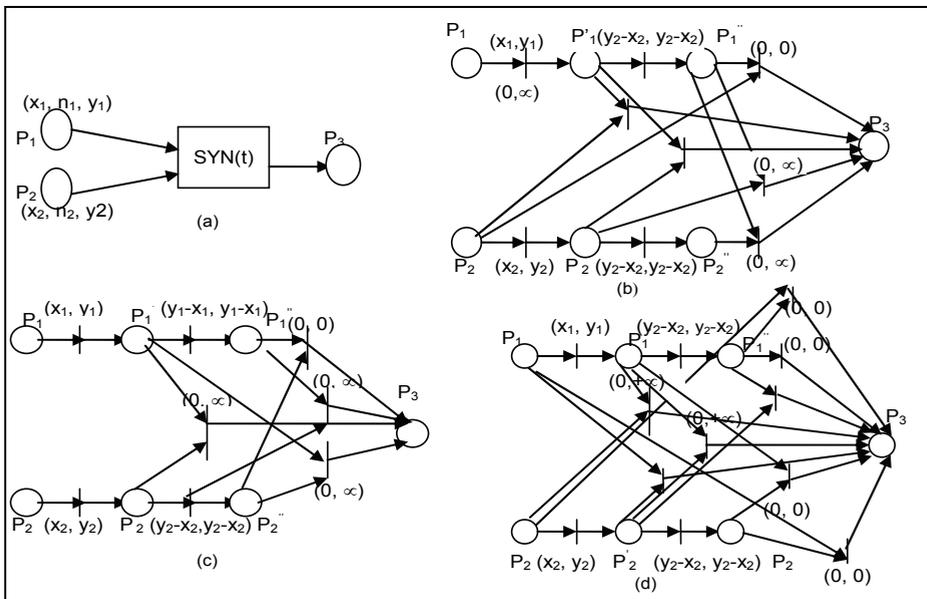
Fig. 4. Diagram of association

Fig. 5. Translation of an inter-stream synchronization schema of a Mp-RdPT net (a) in the form of a t-time Petri net (b, c, d) according to inter-stream synchronization (b) transition of the type «master », (c) transition of the type « weak_and», (d) transition of the type « strong_or ».

5. Analysis of multimedia scenarios using the tool Tina

Tina (Time Petri Net Analyser) (Berthomieu et al., 2004) is a software environment to edit and analyze Petri Net and t-time Petri Net (Merlin and Farber, 1976). In addition to the usual editing and analysis facilities of such environments (computation of marking reachability sets, coverability trees, semi-flows), Tina offers various abstract state spaces constructions that preserve specific classes of properties of the concrete state spaces of the nets. Classes of properties may be general properties (reachability properties, deadlock freeness, liveness), specific properties relying on the linear structure of the concrete space state properties relying on the linear concrete space state (linear time temporal logic properties).

After generating the t-time Petri net, the author investigates the scenario specification before it is delivered to the reader by using the analysis tool Tina. Currently, the following characteristics can be verified by the analysis tool: terminate state existence (i.e., if a state m exists in which non transitions are enabled), safeness (i.e., if every place has only one token), liveness (i.e., if blocking will never occur), reversibility (i.e., if the Petri net come back to its initial state whatever state it reaches), consistency (is a necessary condition for the reversibility that is a difficult property to establish.

6. Object oriented modelling

Our multimedia framework looks to the framework proposed in (Gibbs et al., 1993). It is composed of abstract classes serving to specify interfaces, and suggested procedures for using the classes. The abstract classes are specialized for different multimedia platforms. So, applications using the abstract classes may adapt to variations in platform functionality.

The classes of our framework belong to two distinct groups: media classes and scenario classes. Media classes correspond to audio, video, image, text, and other media types, their basic properties and operations, and scenario classes model temporal composition of media objects. In this paper, we will focus on scenario classes which are a main difference with the framework presented in (Gibbs et al., 1993).

Scenarios are divided into types corresponding to application domains. Each type is represented by a class. These are called scenario classes and form a hierarchy (see Fig. 6).

Nodes depict classes and edges depict superclass/subclass relationships. An edge points from the superclass to the subclass. Instances of scenario classes are called scenario objects. A scenario class models scenario object properties and operations.

7. Architecture of the system prototype

The system prototype is implemented on Compaq Intel Pentium 4 platform using the programming language C++ and Oracle DBMS, and run in a window environment. According to the architecture of the system prototype (depicted in Fig. 7) the following components can be described: (1) Authoring tool which use an editing language of temporal and causal relations; (2) parser which reads and analyses the specification file that contains the temporal specification language; (3) generator which generate the Mp-RdPT model; (4) Translator which translate the Mp-RdPT model to t-time PN; (5) Analysis tool Tina which analyses and verifies properties of the time PN. Manager which manages the multimedia scenarios, such as: the generation, the simulation and the interpretation of Mp-RdPT, object-relational oracle database which stores media objects as large binary objects (BLOBs) in mpeg, gif or jpeg files.
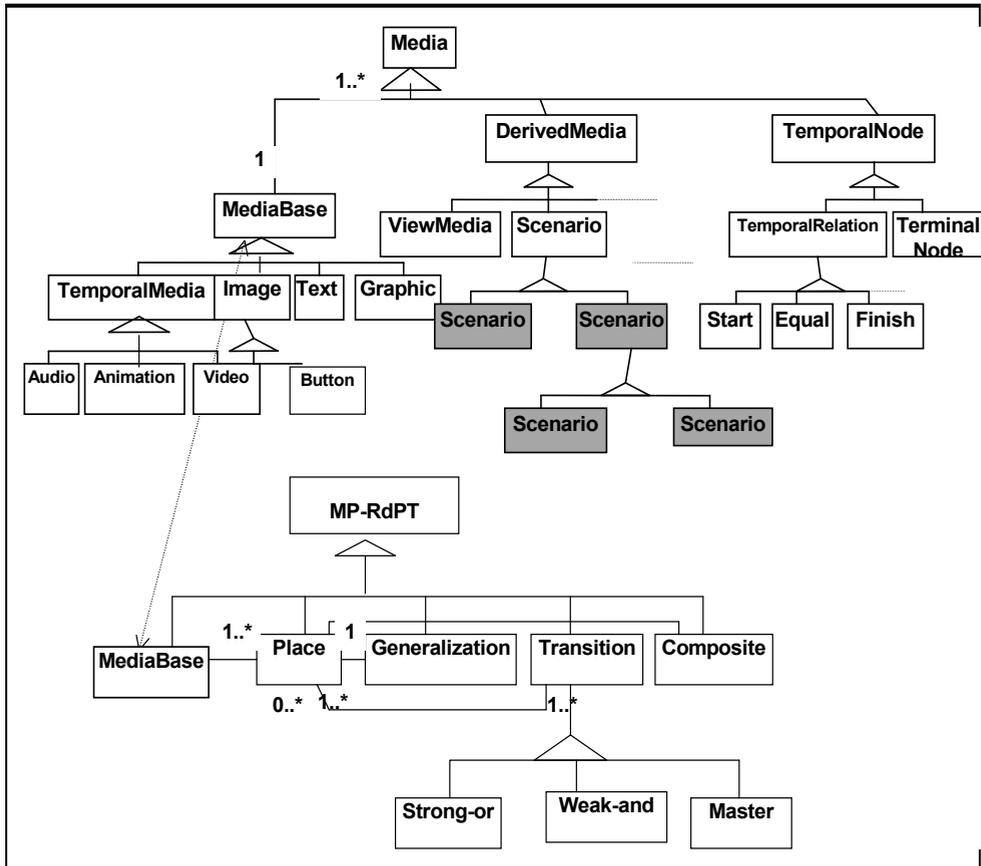
Fig. 6. A class diagram with UML formalism

8. Conclusion and future work

We have presented an approach for multimedia scenario generation in an object oriented database. Our language features consider: - object-oriented concepts for multimedia and scenario modelling; - temporal specification based on temporal and interactive relation; the temporal relations are based on both existing temporal arrangement and causal relations between multimedia objects; - powerful time Petri net automatic generation based on temporal specifications; - and finally user interactions based on composite time Petri net.

In the future, we will perform the algorithms that detect inconsistencies in the Mp-RdPT net generated, and we will provide a support for the programmer to develop distributed multimedia applications using the object-oriented model. It is important to provide communication of various types of data over the high speed ATM network and the synchronization of the multimedia objects at the target system. The target system is the location in which the final synchronization is executed respecting the network delay with insignificant modifications of the earlier synchronization.
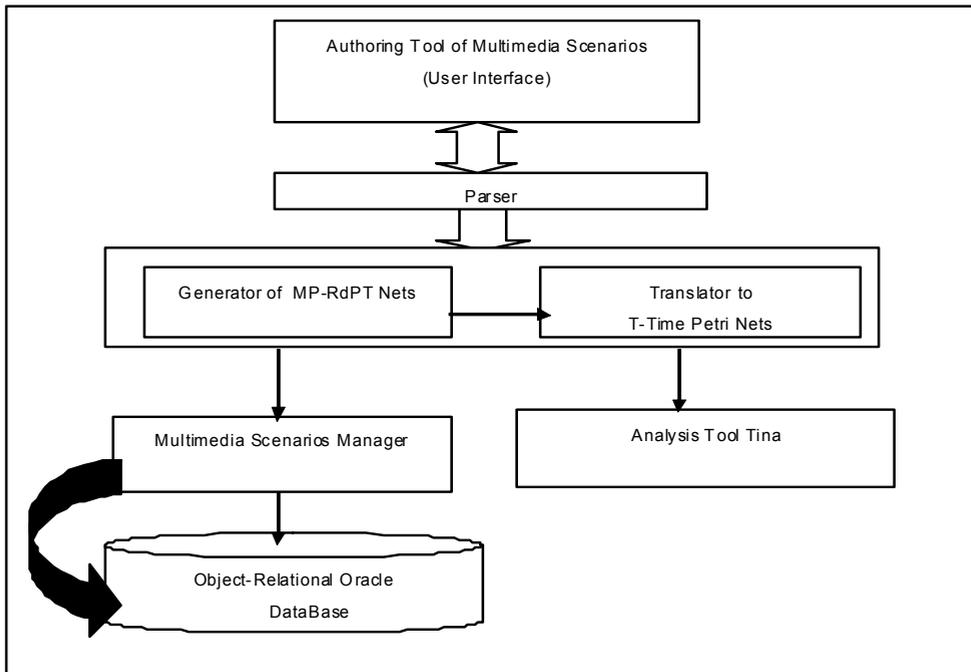
Fig. 7. System prototype architecture

9. References

Adjeroh, D.A & Lee, M.C. (1995). " Synchronization Mechanisms for Distributed Multimedia Presentation Systems, *Proceedings of IWMDMS*, (1995), August, Blue-Mountain Lake, Newark.

Allen, J.F. (1983). "Maintaining Knowledge about Temporal Intervals ". *Communications of the ACM*, 26(11):832–843.

Berthomieu, B.et al., (2004). "The tool TINA – Construction of Abstract State Spaces for Petri Nets and Time Petri Nets". *International Journal of Production Research*. Vol. 42(4) http://www.laas.fr/tina

Blakowski, G. & Steinmetz, R. (1996). " A Media Synchronization Survey : Reference Model, Specification, and Case Studies ", *IEEE journal on selected areas in communications*, Vol. 14(1).

Boronat, F et al. (2008). " Multimedia group and inter-stream synchronization techniques : A comparative study". *Journal of Information Systems*.

Botafogo, B & Moss, D. (1995). "The MORENA model for hypermedia authoring". *Proceedings of the international conference on multimedia computing and systems*, (1995), IEEE Computer Society Press, Los Alamitos, CA.

Chen J-N & Huang, Y-M. (2005). " Using dynamic fuzzy Petri nets for navigation learning. *Exploring Innovation in Education and Research*, Tainan, Taiwan.

Courtiat, J-P. & de Oliveira. R. C. (1996). "Proving Temporal Consistency in a New Multimedia Synchronization Model". *Proceedings of the ACM International Conference on Multimedia*, (1996), Boston, MA.

Gibbs, S. et al. (1993)." Audio/Video Databases : An Object-Oriented Approach ". *Proceedings of IEEE Ninth International Conference on Data Engineering*, (1993), Vienna, Austria.

Ghomari, A & Djeraba, C. (2003). "Towards a Timed-Petri Net Based Approach for the synchronization of a Multimedia Scenario". *Proceedings of the 5th International Conference on Enterprise Information Systems*, April, (2003), Angers, France.

Ghomari, A. & Djeraba, C. (2010). " An approach for synchronization and management of multimedia scenarios in an object-oriented database ". *Proceedings of the International Conference on Research Challenges in Information Science (RCIS)*, Ed. IEEE, ISBN #978-1-4244-4840-1, May (2010), Nice, France.

Hamblin, C. L. " Instants and Intervals ". *Proceedings of the 1st Conference of the International Society for the Study of Time*, (1972), New York.

ISO, (1992) " International Standard. Information Technology Hypermedia/Time-Based Structuring Language (HyTime)". ISO/IEC IS 10744:1997.

Jourdan M. et al. (1997). " A Survey on Authoring Techniques for Temporal Scenarios of Multimedia Documents ", *In Handbook on multimedia computing*, CRC Press.

King, P.R. (1994). "Towards a temporal logic based formalism for expressing temporal constraints in multimedia documents, *Technical Report 942*, Orsay, France: LRI, Université de Paris-Sud.

Little, T.D.C. & Gafoor, A. (1990). "Synchronization and Storage Models for Multimedia Objects ", *IEEE Journal on Selected Areas in Communication*, vol.8(3).

Little, T.D.C. & Ghafoor, A. (1993). " Interval-Based Conceptual Models for Time-Dependant Multimedia Data ". *IEEE Transactions on Knowledge and Data Engineering*, 5(4).

Merlin, P.M. & Farber, D.J. (1976): "Recoverability of communication protocols: Implications of a theoretical study ". *IEEE Trans. Comm.* 24(9).

Na Cheo, J-C. & Furuta, R. (2001) " Dynamic documents: authoring, browsing, and analysis using a high-level Petri net-based hypermedia system. *Proceedings of the ACM Symposium on Document Engineering*, (2001), Atlanta, CA .

Peterson, J.L. "Petri Nets", *Computing Surveys*, 9(3), 225-252, 1977.

Perez-Luque, M.J. & Little. T.D.C. (1996): "A Temporal Reference Framework for Multimedia Synchronization", *IEEE Journal on Selected Areas in Communications* (Special Issue: Synchronization Issues in Multimedia Communication), Vol. 14(1).

Roisin, C & Sèdes, F. (2004). " Time and Documents, *Numeric Document*, 8(4), 23-39.

Shterev, J. "Modelling of Interaction on multimedia Stream and objects by application of Petri nets. *Proceedings of the International Conference on Computer Systems and Technologies*, (2005), Varna, Bulagria.

Senac, P. et al. (1995). "Hierarchical Time Stream Petri Net : Amodel for Hypermedia Systems". *Proceedings of Application and Theory of Petri Nets*, (1995), Giogio de Michelis, Michel Diaz (eds). Lecture Notes in Computer Science N° 935.

Weiss, R. et al. (1995). " Composition and Search with a Video Algebra ". *IEEE Multimedia*, 2(1).

W3C. (2001). "Working draft specification of SMIL, URL: http://www.w3.org/TR/SMIL 2.0
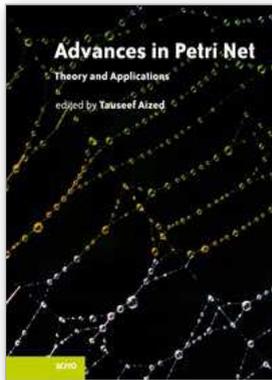
Advances in Petri Net Theory and Applications

Edited by Tauseef Aized

The world is full of events which cause, end or affect other events. The study of these events, from a system point of view, is very important. Such systems are called discrete event dynamic systems and are of a subject of immense interest in a variety of disciplines, which range from telecommunication systems and transport systems to manufacturing systems and beyond. There has always been an intense need to formulate methods for modelling and analysis of discrete event dynamic systems. Petri net is a method which is based on a well-founded mathematical theory and has a wide application. This book is a collection of recent advances in theoretical and practical applications of the Petri net method and can be useful for both academia and industry related practitioners.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds