Multi-Level Model for Synchronizing Temporal Streams on HLA based Distributed Multimedia Applications Using HTSPN

Akram Hakiri, Michel Diaz, Slim Abdellatif, Pascal Berthou, Thierry Gayraud

Multi-Level Model for Synchronizing Temporal Streams on HLA based Distributed Multimedia Applications Using HTSPN

Akram HAKIRI [1, 2], Michel DIAZ[1], Slim ABDELLATIF [1, 2], Pascal BERTHOU[1, 2], Thierry GAYRAUD[1,2]

[1]CNRS ; LAAS, 7, avenue du Colonel Roche, 31077 Toulouse, France

[2] Université Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France

Email: {Hakiri, Diaz, Slim, Berthou, Gayraud}@laas.fr

Abstract— The major issue in distributed multimedia simulation is how to specify and implement synchronization scheme satisfying the QoS applications requirements. This paper suggests the use of a powerful synchronization scheme based on HTSPN between different flows issued from distributed multimedia systems to fulfill these requirements. Such an interpreted model permits powerful analysis techniques for validating the quality of service in computer networks before protocol implementation. Consequently, it allows the specification of both the temporal non-determinism of weakly distributed applications and the temporal variability of the multimedia processing. An example is used to demonstrate the capabilities of this scheme to specify the QoS requirements of simulated applications.

Keywords; Synchronization scheme, HLA, distributed multimedia interactive simulation, HTSPN, QoS requirements.

I. INTRODUCTION

Advanced computer multimedia simulation systems are characterized by the integration of virtual environments within time-dependent and time-independent media communication. The key issue which provides integration of all streams within the same system remains the synchronization of and between various kinds of media and data. The word synchronization refers to the temporal relationship between objects in distributed interactive multimedia system. The time-dependent media objects are described by the temporal relations between consecutive units of the media [6]. If these time durations are equals the media is called continuous, like video. It is very interesting to focus in this temporal aspect of distributed multimedia simulation to provide a coherent and consistent interactive simulation.

Therefore, on one hand, synchronization schemes bring important contributions to the emerging concepts of distributed simulation systems, especially when these systems must maintain temporal relations between various streams. On the other hand, distributed interactive applications like DIS and HLA [1] need structural and abstract approaches to specify the synchronization scenarios between intra-flow, inter-flows and inter-objects to allow an adequate management of the system resources.

The objective is maintaining the temporal relations between various streams in distributed multimedia applications. In particular the skew between the streams must be restricted. This concern includes how to specify, analyze and implement synchronization scheme which fulfill all these requirements. Furthermore, the model can also be used for validating the simulation principles and algorithms. The analysis of the possible expressive power required for distributed interactive systems lead to select a formalism based on Petri nets (PN).

This paper suggests a synchronization scheme which fulfills the QoS requirements of distributed multimedia simulation using a formal model based on Hierarchical Stream Time Petri Nets (HTSPN) to specify and analyze synchronization constraints between synchronized units in intra-flow and inter-flow cases for the specification and the verification of the next generation of distributed interactive multimedia simulation.

The suggested model aims to facilitate the editing phase and the development time required to deliver high fidelity simulation that will respect all structural, temporal and logical application related constraints and an example of HLA is given to show the general interest of our approach.

This paper is as follow: after a brief introduction, Section II summaries briefly synchronization supplies, the motivation of using Petri Nets and presents the HLA fundamental aspects. An overview of the HTSPN is given in Section III. Section IV discusses the QoS specification of the distributed multimedia interactive simulation application. The specifications provided by section IV are injected within the formal model based on HTSPN and discuss its usage within HLA Simulation.

II. BACKGOUND AND PURPOSE

Synchronization is addressed and supported by many system componences including communication system, documents. Hence, synchronizationmust be concidered in a multimedia systems.

A. Synchronization issues

The asynchronous aspect of media objects is the foremost characteristic of the distributed multimedia interactive simulation. The exchanged flows between users are defined as a set of information units, called Logical Data Units (LDU), to facilitate the presentation, the editing and the modification of these flows. In general, several granularity levels of LDUs in media objects exit. The level of granularity used is application-dependent. It implies a hierarchical decomposition of media objects. In addition, LDUs can be classified into closed and open LDUs. Closed LDU's have a predictable duration, like media file stored in hard drive. Open LDU's represent input from live source, like a live camera or objects that include user interaction. In such case, it is hard to handle these LDU's, then it is more typical to select them into blocks of a fixed duration, called synchronization Units (SU). Synchronization units have to process the LDU's, providing a way to modulate a synchronization scheme for each flow and to provide an optimal control of each flow with respect to the system

resources. Figure 1 shows the correspondence between the LDU's (UI) and the SU's.
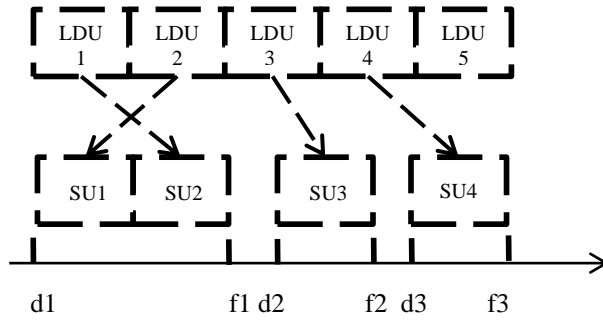

Figure 1: correspondence between LDU's and SU's

Each SU_i is associated to a sequence of LDU_i, together with each starting date d_i and finishing date f_i. The granularity of the information induces the performance of the scheduling protocol. In order to process streams using a set of available resources, the synchronization scheme has to use synchronization units adapted to the specific flow and to the synchronization of the media acquisition.

The specification of the synchronization scheme should be specified using a formal model to conduct an editing and a verification phase before any implementation.

B. Petri Nets Properties

The formal verification of the specifications, when possible, seems essential before any implementation, because it allows the early discovery of errors at the design stage.

Furthermore, the model can also be used for validating the simulation principles and algorithms, and also to derive the executable code and consequently reduce the time needed to develop distributed interactive simulations.

To assess the adequacy of a formal model for specifying a synchronization problem means to fulfill the following three basic criteria:
- The model must provide a good expressive power: is must be complete for a given problem.
- Sufficient power analysis: it must be supported by verification techniques for the system properties.
- Adequate power modeling: the model must be able to abstract and hide not relevant information to master its complexity.

Formalism for synchronizing distributed interactive simulations must be primarily a temporal model because it induces a required flexible management of system resources. It also allows the expression of the non-determinism that may occur when a timed de-synchronization occurs between different distributed streams, especially when these flows are very heterogeneous, such as the union of streaming media (audio, video, images) and streaming interaction flows coming from the actors of the virtual environments.

The analysis of the possible expressive power required for distributed interactive systems lead to select a formalism based on HTS Petri nets (HTSPN).

C. The High Level Architecture (HLA)

HLA is a general purpose, reusable software architecture for the development and execution of distributed simulation application. In HLA terminology, a set of simulations that is capable of interoperating is a federation, and the individual simulations are federates (see figure 2).

HLA is defined by three documents:
- Rules: are principals and conventions that should be respected to provide the interoperability and the reusability of the simulation exercise.
- Object Model Template: a formal for specifying simulation data in term of hierarchy of object class, attributes, interactions and interaction parameters.
- Interface Specification: provides services and communication mechanism, forming a piece of software to ensure the information exchange usually implemented within Run-Time Infrastructure (RTI).

RTI is software which implements the interface specification of the HLA. It provides services in producer/consumer paradigm. RTI provides a C++ library (other languages like Java, C#, Ada exist), libRTI, through which federates can exchange data.
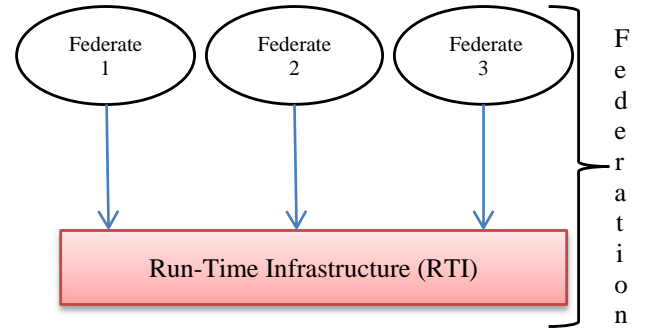

Figure 2: HLA Federation conceptual view, with federates exchanging data through the RTI.

Within libRTI, the class *RTIAmbassador* bundles the services provided by the RTI. All requests made by a federate on the RTI take the form of a *RTIAmbassador* method call. The abstract *FederateAmbassador* identifies the callback functions each federate is obliged to provide.

D. research opportunities

From the above discussion, contributions suitable for distributed multimedia involve media synchronization modeling [10, 13], compositional approaches for media object synchronization specifications [5,14], providing formal approach for expressing distributed simulations [8], the impact of Time Service Management in HLA to specify an engine based on Stochastic Petri Nets to run the distributed simulation, and proposes the use of HLA as a platform of reference to compare different approaches for partitioning and distributing application executions [9], duration calculus [11], event based synchronization mechanism [7, 12]. These approaches do not permit modeling inter-stream resynchronization strategies caused by asynchronous events and interruptions that can stop the synchronization scenario.

As a consequence, it seems necessary, using the same basic formalism (Petri Nets), to extend the power of the previous models to express the spatial, temporal, logical and semantic structures that appear in the distributed interactive multimedia simulation. This capacity of resynchronization after interruption scenarios leads to favor HTSPN.

III. HTSPN

Definition 1:
A TSPN [2, 3] is a tuple (P, T, α, β, M0, IM, SYN), where:
- (P, T, α, β, M0) defines a Petri Nets.
- IM is an interval mapping function that associate a Temporal Validity Interval (TVI) with any outgoing arc from the places, where:
  - A = { a = (p, t) $\in$ PxT | β (p, t) $\neq$ 0}
  - IM : A $\rightarrow$ (Q+$\cup\infty$) x (Q+$\cup\infty$) x (Q+$\cup\infty$), IM(a) $\rightarrow$ (x, n, y), $0 \leq x \leq n \leq y$
- SYN is the typing function that associates a synchronization type with a transition:
  - SNY: T$\rightarrow$ {and, weak-and, or, strong-or, master, or-master, strong-master, weak-master}.

Definition 2:
A HTSPN [3, 7] is a Hierarchical TSPN, i.e. a tuple HTSPN = (R, S, Pin, FS, Fin) such that:
- R = (P_r, T_r, α_r, β_r, M0$_r$, IM$_r$, SYN$_r$, PT$_r$) is a TSPN, called the root of the HTSPN, extended to provide the link, atomic and composite place type modeling:
- (P_r, T_r, α_r, β_r, M0$_r$, IM$_r$, SYN$_r$) defines the TSPN.
- PT$_r$: $P_r$$\rightarrow$\{atomic, composite, link\} is a type place assignment function.
- S = \{S_i | i$\in$ I\} is a finite set of Structured Time Stream Petri Net (STSPN):
  - S_i = (P_i, T_i, α_i, β_i, M0$_i$, IM$_i$, SYN$_i$, PT$_i$)
  - (P_i, T_i, α_i, β_i, M0$_i$, IM$_i$, SYN$_i$) defines a STSPN.
  - PT$_i$: $P_i$$\rightarrow$\{atomic, composite\}
  - The sets of net elements are pairwise disjunctions:
  - $\forall$ (i,k) $\in$I, i$\neq$k, (($P_i\cup T_i$)$\cap$($P_k\cup T_k$) $\neq\varnothing$).

A STSPN is a class of TSPN built using exclusively the seven synchronization operators (before, meets, overlaps, during, starts, finished and equals) defined in [8].
- Pin $\subset$ P is the set of initial places of all nets in S, where Pin = $\cup_{i\in I}$P.
- FS: C$\rightarrow$S is a function that associates a composite place with an element of S and where
  C = \{p$\in$P|$\exists$i$\in$I, PT$_i$(p) = composite\} $\cup$ \{p$\in$P$_r$|PT$_r$(p)=composite\} \}.
- Fin: S$\rightarrow$Pin is a function that associates an element of set S with an input place.

Indeed, the HTSPN model is an extended Petri Net model that used timed arcs for the modeling of multimedia processing (communication, presentation...). The temporal jitter appearing inside weakly synchronous multimedia systems is modeled by the arc Temporal Validity Interval (TVI). These arcs TVI are tuple [x, n, y], where x, n and y are respectively the minimum, the nominal and the

maximum admissible durations of the related processing's. Such way of multimedia systems modeling allows the expression of both the temporal non-determinism of weakly synchronization in distributed multimedia applications and the admissible temporal variability of multimedia objects. Temporal drifts between multimedia streams can be fully and accurately specified with the help of 9 different synchronization semantics that can be selectively associated with transitions. As a consequence, HTSPNs appear to be a powerful tool for the formal modeling, analysis, verification and simulation of distributed multimedia simulation systems.

HTSPN models allow three fundamental concepts to be formally described with powerful temporal extensions: the atomic, the composite and the link components.

Atomic Component: an atomic component is modeled in HTSPNs by an arc with a TVI and a place associated with one atomic resources type, for example video data with [8, 10, 12] as TVA. Atomic synchronization layers aim to describe synchronization constraints inside atomic components by specifying intra-stream synchronization.

Link Component: a link is modeled in HTSPN by a timed arc (L, t), where L is the link (to be layered) place. The TVA associated with the link introduces the timed link concepts. Using the HTSPN firing rules [2, 3], timed links allow the modeling and the formal specification of the transversal semantics of the application layer.

Composite Component: composite component provides a hierarchical structuring mechanism based on the recursive composition of atomic and composite component through the use of sub-nets. The HTSPN use these composite type places that are not only structurally, but also temporally, equivalent to a (sub) net. A composite layer is able to describe inter-stream synchronization constraints.
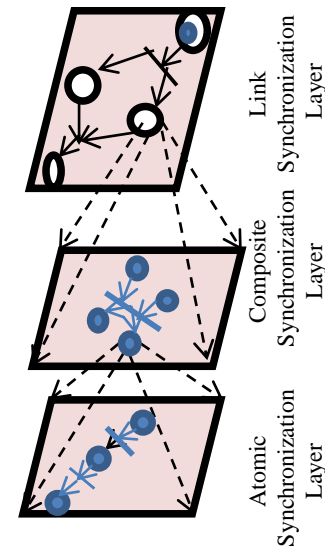

Figure 3 : The HTSPN Model Layers

IV. QUALITY OF SERVICES REQUIREMENTS

Synchronization requirements can be expressed by a QoS specification [3, 13]. The Quality of service in distributed interactive applications consists of static parameters and dynamic parameters. The static parameters, such as image

size, or quality of the audio stream, are important because they act directly on the interval related to the resources of the application (but note that they do not express the mechanisms and algorithms of this application). In a real implementation, a set of dynamic parameters which contribute to the quality of service will require specific interaction procedures to be enforced for the application to behave properly. These parameters are divided into three classes:

- Quality of the media
- Quality of the presentation
- End-to-end latency.

Media synchronization parameters describe the temporal intra-stream, inter-stream and inter-objects synchronization. These temporal criteria define the maximum admissible jitter for images, objects and audio data. For inter-stream synchronization, they define the maximum acceptable time gap between flows. This gap is called the inter-stream drift and is the result of the cumulative effects of all jitters which appear during the presentations of the successive objects of the different flows.

The quality of the mono-media presentation describes the quality of the discontinuity of a single stream. This discontinuity occurs for instance when data are lost; it can cause a significant loss of synchronization, and it becomes very important to optimize the quality of the presentation at the receiver side to present the application.

The end to end latency defines the maximum allowable transfer delay between two remote entities. This period corresponds, for example, to the delay when a sender pronounces a word and when the receiver receives the sound. This delay should not exceed a given limit since it affects the interactive communication between the remote users.

A. Media Synchronization Parameters

Consider two flows F1 and F2 modeling as an example an audio stream and video stream (may be images from a camera or objects in the distributed simulation) that require synchronization between them. We define the TVA of the streams by:

$[t_{min}, t_{nom}, t_{max}]$ for stream F1
$[t'_{min}, t'_{nom}, t'_{max}]$ for stream F2

We suppose t_1 and t'_1 the hardware related latency, where:
t_1: Video card Latency
t'_1: Audio Card Latency
Assuming:

- N_p : number of objects sent within F1 during one synchronization period (respectively, N'p number of object of Flow F2 sent during one synchronization period)
- N_d : number of object shift allowing the resolution of hardware latency problems, $N_d \in N$, ad if $N_d < 0$: the images presentation is delayed, $N_d > 0$ it is ahead

Considering the inter-stream drift between to flows called *d*:
- if d>0 Flow F_2 is presented after flow F_1
- if d<0 Flow F_2 is presented before flow F_1
d is given by formula (1):

$$d = d_{decal} + d_p \ (1)$$

- d_{decal}: drift introduced when the synchronization point is moved
- d_p, inter-flow drift which may appears over a period of synchronization.

The values of N_d and d_{decal} could be calculated using relations *(2) and (3):*

- $N_d = round\left(\frac{t'_1 - t_1}{t_{nom}}\right)$ *(2)*
- $d_{decal} = |\ t'_1 - t_1| - (N_d \times t_{nom}) \times (t'_1 - t_1)/|t'_1 - t_1|$ *(3)*

Furthermore, the inter-flow drift in one synchronization period is comprised between two limiting cases which consist in considering the minimum presentation time for one flow and the maximum time for the other which give the flowing relationships:

- $N_p \times (t_{min} - t'_{max}) \le d_p \le N_p \times (t_{max} - t'_{min})$ *(4).*

Sine*:* $d = d_{decal} + d_p$, then we get the ensuing relations:

- $N_p \times (t_{min} - t'_{max}) + d_{decal} \le d \le N_p \times (t_{max} - t'_{min}) + d_{decal}$ *.(5)*

Or *d* is given in the relation (6):

$$-d_{max} \le d \le d_{max} \ (6)$$

Assuming that:
- $N_p \times (t_{min} - t'_{max}) + d_{decal} \ge -d_{max}$
- $N_p \times (t_{max} - t'_{min}) + d_{decal} \le d_{max}$

Hence:
- $N_p \le (d_{max} - d_{decal})/(t_{max} - t'_{min})$
- $N_p \le (d_{max} + d_{decal})/(t'_{max} - t_{min})$

Consequently:
- $N_p = E\{min\{\frac{d_{max} - d_{decal}}{t_{max} - t'_{min}}; \frac{d_{max} + d_{decal}}{t'_{max} - t_{min}}\}\}$ *(7)*

Thereby, giving the previously relations, especially using formula (2) and (7) which provides calculating the shift rendezvous et the inter-stream synchronization period, we can retrieve the number of objects that should be used within the HTSPN synchronization scheme. This formal model provides the design of synchronization mechanisms to be used in asynchronous distributed multimedia applications.

Table 1 presents the data synchronization parameters for the video and audio cards of the test platform (will be explained later on) where T1, T2 are respectively 248 ms and 100ms and the hardware latency of commercial Human-in-the-Loop simulator denoted T3 is about 65ms (may vary from 60ms to 75ms).

Table 1: Synchronization parameters involved for various streams within the considered PLATFORM

parameters	Audio/Simulation	Audio/video
N_d	1	-3
d_{decl}	-55	2
d_{max}	120	120
N_p	5	2

Thus, we can retrieve the number of objects that should be used within the HTSPN synchronization scheme

In fact, these mechanisms can generate a dis-continuity of the application, especially when an object is lost or backward within the communication system, it can't be represented at the receiver site. So, it is necessary to manage the latency at all the communication entities involved in the application.

B. End-to-End Latency Management

The packet size is a very important criterion that has to be carefully chosen for the QoS constraints to be fulfilled. Indeed, for example for audio data, the length of the packet affects the time required to produce it. The packet audio system needs low end-to-end latency.. The requirement of low latency means that it is better for the senders to send small packets frequently rather than large packets seldomly. As a result, distributed multimedia interactive applications tend to be defined by sending small packets frequently.

Let us assume that the acceptance purpose is to provide a 150 ms end-to-end latency: 50 ms can be taken as the maximum time allowed for preparing and sending a packet, also for processing and presenting it in the receiving application, and also can be the propagation delay in the network.

It should be noted that if some problems come from the network, and if then the different flows are not received at the same time, some application incoherence could result and the corresponding flows need to be re-synchronized, if possible, at the receiver side.

For example, as applications of distributed simulations incorporate multimedia flows, together with flows resulting from the interactive system control, they may become incoherent after crossing a (wide area or other) disrupting network. To ensure consistency between these flows, an adequate synchronization scheme between these flows is necessary and has to be specified.

Such synchronization between the flows can be defined by successive steps based on the specification of multimedia objects describing all the temporal dependency of the included objects in the media object.

V. THE FORMAL MODEL

When considering the HLA simulations, the HLA-RTI can react both as the transport layer and control layer. In some case it is possible to use Object Classes to control the synchronization scheme and the Interaction Classes to transport data. Due to the distributed aspect of this asynchronous application, the HLA model can be applied to alleviate the temporal constraints and carry on more flexible control to the simulation and interactive streaming from the sender application to the remote hosts. We applied the synchronization model to a real existing simulation platform used in car driving schools.

Basically, the network simulation platform is a distributed interactive simulation platform where users interact with each other over HLA-RTI middleware. Figure 4 describes the platform architecture governing the HLA-RTI middleware at the receiver side.

The developed formal model involves three flows, an audio stream, a video stream and an interactive stream issued from the human control with a real simulator consoles. The selected values of the temporal validity interval of each flow are respectively [20, 25, 30], [40, 50, 60] and [16, 20, 24] and the associated drifts are respectively ±5 for the audio stream, ±10 for the video and ±4 for the simulation flow.
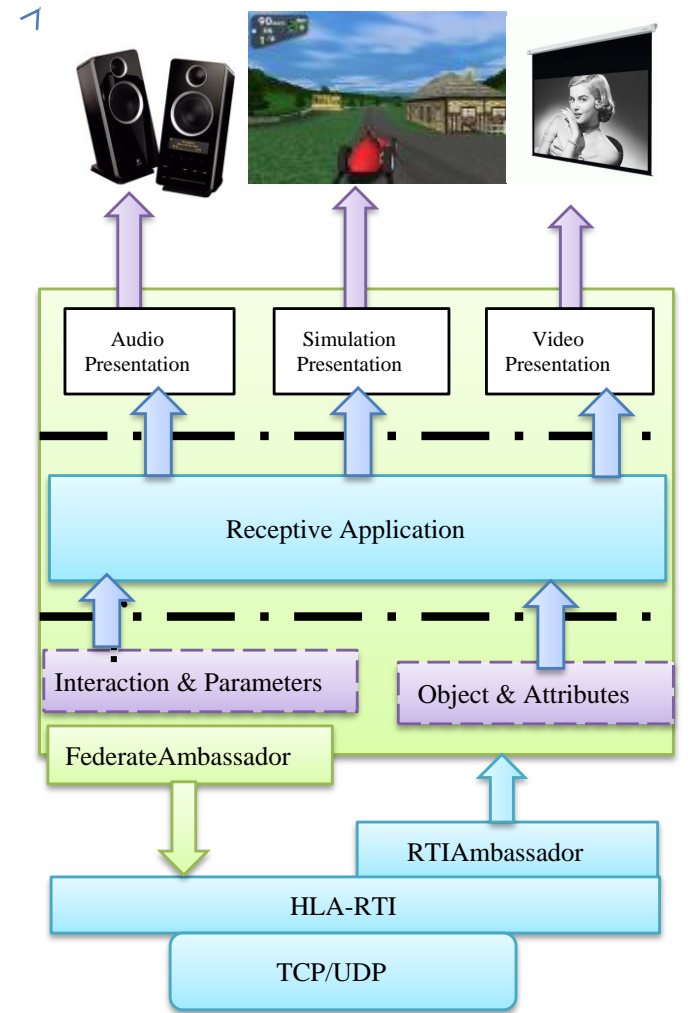

Figure 4: Simulation Platform at the receiver side

The synchronization parameters in equations (2) and (7) are shown in table 1. It should be noted that Np denotes the number of media object during one synchronization period. Firstly, the audio stream is compared to the video stream to check the synchronization between both the two flows, and then the simulation flow is compared with the audio stream. Indeed, the audio stream should be selected as a master flow. The number of object that must be sent within one synchronization period between the audio and the simulation remains equal to 5, where this parameter should be equal to 2 between video and audio stream, i.e. that the simulation stream can be delayed regarding the audio stream, which need to be here the master stream; and the video stream can be also delayed when considered with the audio stream. Therefore the audio stream must fulfill both the two synchronization requirements of the application.

The goal of the synchronization is to exactly reproduce the presentation the temporal relation as they existed during the capturing process at the sender site. In the scope of source/sink synchronization scenario, data streams are created are which presented at the sink with the preservation of several streams. The managements at the sink are to adapt the presentation to the available resources. In summary, we must emphasize that the primary demand is the presentation of data according to the temporal relations which exist during the capturing process of media objects.

Figure 5 describes the HTSPN synchronization engine of the application. The Figure illustrates synchronization scenario in distributed multimedia interactive application and illustrates the behavior of the application caused by the staggered rendezvous which resolve the problem of the inter-stream synchronization of this application.

This solution uses the hierarchical capabilities provided by the HTSPN. This specification is carried out from a HTSPN comprising 5 HTSPNs (intra-stream audio and video are not shown in the figure because the lack of space) and 3 hierarchical levels: the highest level is called the link synchronization layer and it specifies at the presentation of the receiver host the full constraints of the inter-stream synchronization between the audio stream, modeled by *Audio* place, the flow of the interactive simulation, modeled by the *Simulation* place, and the video stream, modeled by *Video* place (see figure 5.a).

The second level (see figure 5.b) is called the composite synchronization Layer. Within this layer places S1to S5 represent the simulation object flow of the HLA distributed simulation application, places AUD1 to AUD4 point out the audio objects and finally places VID and VID 2 illustrate the video objects. The third layer is called the atomic synchronization layer (see figure 5.c) and it specifies the intra-stream synchronization layer. Both the atomic or composite components and materials are managed at the HLA-RTI level. The link layer is independent of the middleware, and it represents the application level.

The developed model specifies especially the control that should be implemented to ensure that video stream and simulation stream must be controlled by the audio stream, and because this last is the master stream it should ensure that the delay should not exceed the value of 120ms for all other controlled flows. The temporal validity interval, [80, 100, 120], at this layer corresponds to the inter-stream synchronization and it means that the transition will be fired in the interval min, max= [80, 120].

The delay of the simulation flow must be driven by the audio stream because it is the most time sensitive one and it will be then selected master stream: it control the time schedule for the firings of the transitions. The video flow is more flexible and it can be slightly delayed.

The granularity of the synchronization is determined by the maximum acceptable inter-stream drift. As the audio stream has a possible drift of ± 5 ms, the advance of the interactive flow results only from the cumulative effect of the drifts of this flow. The allowable drift of the video is $\pm$ 10ms: this drift is achieved by the treatment of 4 units of synchronization of the audio and 5 units of the Interactive flow. Regarding the delay in presentation, in order to preserve the logical synchronization between data and various streams, the loss of one video object causes the loss of all its related data packets because this stream is not compressible. As far as for the audio stream, when an audio object is lost then all related video and simulations objects are also lost.

As a consequence, the synchronization transition will be defined at the rendezvous which occurs at a period equal to 4 audio objects that is agree with 5 simulation objects and 2 video objects. This strengthens the master behavior of audio flow which need more careful.

During the simulation, HLA-RTI supports the transmission of audio, video and interactive streaming from the sender application to the remote hosts. It allows both the transport layer and control layer. HLA defines two types of information exchange: the objects and the interactions. Objects are inherently persistent during the simulation, represented by atomic component; they implement the flow control.

The intra-stream synchronization is managed by the objects that control the constraints of quality of service required by the flows. The interactions are persistent and will be able to natively transport the flows between the Federates.
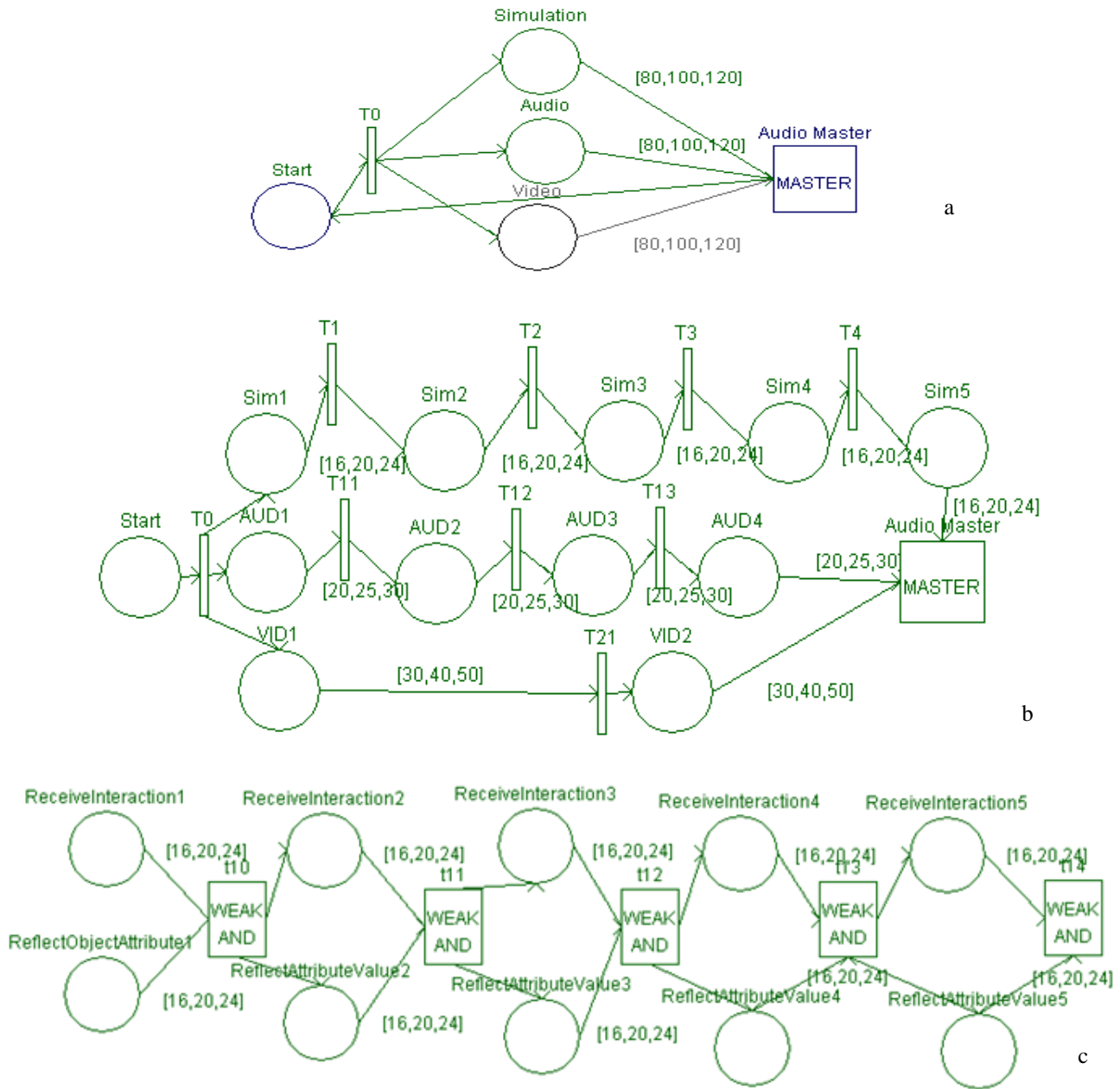
Figure 5: Synchronization Model of the three considered flows: a) the root of the hierarchy specifying the control of all the streams, b) specification of the inter-stream synchronization; c) specification of the intra-stream synchronization of the simulation flows.

HLA Objects and Interactions are shown in within the network in figure 5.c and they ensure the intra-stream synchronization on the interactive simulation flow. For the other flows (audio and video) the intra-stream synchronization scheme is applied with the consideration specific aspects of each stream (these schemes are not presented in figure 5) and bearing in mind the number of objects composing each flow.

This net specifies in particular the control that must be implemented to ensure the adequate synchronization between these three flows, e.g. to ensure that the video stream is no later than 20ms compared to the other flows.

The control of the jitter of this stream should be done by the HLA-RTI middleware to ensure that all constraints of synchronization with the other streams are enforced. The HLA Objects should control independently

each stream using native HLA APIs *UpdateAttributeValues ()* and *reflectAttributeValue ()*. These functions are able not only to control the advance of a flow compared to the others, but also to ensure the intra-stream synchronization. The HLA-RTI APIs *sendInteraction ()* and *receiveInteraction ()* could be used to send data. Because audio and video objects do not need in many cases to be exchanged between federates, their data packet should be send using the HLA interactions. HLA provides many other APIs that can use in the implementation.

As the synchronization is implemented at the receiving side, to schedule the data reception, the API *tick(Tmin,Tmax)* could be used with two arguments that are the minimum and the maximum values used in the temporal validity interval; for example tick(0.16,0.24) can be used for the simulation stream.

VI. CONCLUSION

In distributed multimedia interactive simulation, several aspects regarding the synchronization must be considered. In this contribution, a HTSPN formal model is presented offering at the same time a good power modeling for the expression and analysis of temporal constraints involving the intra-stream and the inter-stream synchronization. The intra-stream synchronization is defined as a synchronization of the LDU's of one media object. The inter-stream synchronization is referred to synchronization between media objects. The required QoS of the temporal relationship to be presented to the viewer are studied regarding the user's perception. This involvement allowed us to specify precisely, completely and in a unified way the multi-level logical, temporal and semantics timing constraints that are fundamental for synchronized distributed applications. we also focused on HLA based simulation application as a case study to specify, analyze and verify the developed model which satisfied the needed QoS.

Additional efforts are required for the development of applications that takes into account formal models before any implementation. Many more ideas, protocols and products have implemented some kind of synchronization. However, it is still a matter of research to find out which involve the prior modeling and specification, especially in distributed asynchronous environment.

ACKNOWLEDGMENT

REFERENCES

[1] SISO-STD-004.1-2004 - Dynamic Link Compatible HLA API Standard for the HLA Interface Specification (IEEE 1516.1 Version)

[2] P.Senac, M.Diaz, A.Leger, P.De Saqui Sannes, Modeling logical and temporal synchronization in hypermedia systems, IEEE Journal on Selected Areas in Communications, Vol.14, N°1, pp.84-103, Janvier 1996

[3] Michel Diaz. Patrick Sénac. Time Stream Petri Nets a model for timed multimedia information, in M. DIAZ Editor, Petri Nets: Fundamental models, Verification amd Applications, Wiley and ISTE, N°ISBN 978-1-84821-079-0, 2009

[4] Roberto Willrich, Patrick Sénac, Michel Diaz, , Pierre De Saqui-Sannes. Formal Framework for the Specification, Analysis and Generation of Standardized Hypermedia Documents

[5] Roberto Willrich , Pierre De Saqui-Sannes , Patrick Sénac , Michel Diaz, Multimedia Authoring with Hierarchical Timed Stream Petri Nets and Java, Multimedia Tools and Applications, v.16 n.1-2, p.7-27, January-February 2002

[6] Blakowski, G. Steinmetz, R. A media synchronization survey: reference model, specification, and case studies. IBM Eur. Networking Center, Heidelberg; IEEE Journal on Selected Areas in Communications.

[7] Patrick S, Pierre De Saqui-Sannes , Roberto W. Hierarchical Time Stream Petri Net: A model for hypermedia systems. Lecture Notes in Computer Science. Springer. ISBN 978-3-540-60029-9

[8] Ratan. K, Guha, M. A.B. A Framework For Modeling High Level Architecture (Hla) Using Petri Nets. University of Central Florida. Department of Computer Science. Orlando, FL 32816

[9] Kpatcha M. B, Kouakou D, D.Kand Helena. S, Exploring impact of time management services on HLA-based Petri Nets Simulation Engine. Simulation Methods and Applications: Simulation Practice and Theory. Volume 9, Issues 3-5, 15 April 2002, Pages 143-166 .

[10] S. Thong, K. Cooper, and M. Ito, "FormalMethods for Modeling Multimedia Synchronization Requirements" Proc. Of IC:lVP'9,5 - Int. Conf on Network Proto- C O ~ S , Tokyo NOV 7-10, 1995.

[11] MA HuaDong. Specification and Verification of Multimedia Synchronization in Duration Calculus. School of Computer Science and Technology Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China

[12] H van , Correia N, David A , David J.D, Graham J.R and James V.L. A standard model for multimedia synchronization: PREMO synchronization objects.Multimedia Systems.Éditeur Springer Berlin / Heidelberg. Volume 6, Number 2 / mars 1998.

[13] S.Vuong, J. Goncalves', K. Cooper, M. Ito, and S. Stiubiene.On Modelling Multimedia Synchronization Requirements. Departments of Computer Science and Electrical Engineering. The University of British Columbia

[14] Anisimov, N.A.; Kovalenko, A.A.; Postupalski, P.A.; Vuong, S.T. A compositional approach to the specification of multimedia objectsusing Petri nets. Emerging Technologies and Applications in Communications, 1996. Proceedings., First Annual Conference on Volume , Issue , 7-10 May 1996 Page(s):38 - 41