A Formal Model for the Specification and Analysis of HLA-based Distributed Multimedia Interactive Simulation using Hierarchical Time Stream Petri Nets

Akram Hakiri[1,2], Michel Diaz[1,2]
[1]CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France
[2] Universit de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France
Email: {Hakiri, Diaz}@laas.fr

***Abstract*—This paper proposes a formal model for the specification and analysis of distributed multimedia simulation. This model is based on Hierarchical Timed Stream Petri Nets (HTSPN), which has been proposed for specifying temporal and logical constraints in high level multimedia description and simulation. It takes into account a powerful synchronization definition between different flows issued from distributed multimedia systems. A simulation was done using a special Java-based framework to assess the methodology and analyze the expression and interpretation power of HTSPNs. For instance, 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-Formal Model; Distributed Multimedia Simulation; HLA; HTSPN.

I. Introduction

The specification and the verification of temporal and logical properties of distributed multimedia interactive simulation is a fundamental step to be conducted before implementation. Therefore, on one hand, synchronization schemes [6] 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, HLA-based applications [1]) need structural approaches to specify the synchronization scenarios between intra-flow, inter-flows and inter-objects to allow an adequate management of the system resources. This paper suggests to use a formal model based on Hierarchical Stream Timed Petri Nets to specify and analyze synchronization constrains 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 proposed model is applied to an HLA based simulation which includes audio, video and interactive streams issued from a selected application. Using the power modeling of Petri Nets suggests the specification of a requested quality of services in distributed asynchronous multimedia application. The synchronization scheme developed here discus applications that involve HLA and are built on HLA-RTI APIs. Its aim is 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.

This paper is organized as follow: after a brief introduction, Section II introduces the motivation of multimedia formal specification. In Section III Petri Nets have been selected for specifying distributed multimedia applications. Section IV presents a set of QoS requirements to be used in distributed multimedia simulation. Section V introduces the formal model and shows analysis results. Transport architecture is presented in Section VI and conclusion is given in Section VII.

II. Motivation

In the general case, flows need to satisfy natural synchronization constraints and synthetic synchronization constraints between applications. The natural synchronization constraints are intrinsic to the flow itself and need to be respected when presented to the remote hosts to ensure the comprehension of the associated information. For example, in SECAM systems, a video frame is be displayed 25 times per second. These constraints are given by codecs. Synthetic constraints are imposed by the application itself and results from the abstract global synchronization specified by developers. For instance, an audio stream must start when a given event occurs. To handle the granularity of these constraints, synchronization units have to process the information units, 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.

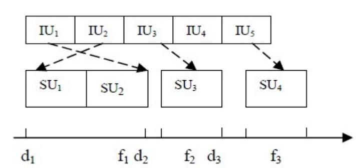

Figure 1. Correspondence between Information Units and Synchronization Units

Figure 1 shows the correspondence between the information units (UI) and the synchronization units (SU). Each synchronization unit, noted SU_i, is associated to a sequence of information units (IU_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. The analysis of the possible expressive power required for distributed interactive systems lead to select a formalism based on Petri nets (PN).

Other research contributions have been proposed to provide a formal approach for the specification of the distributed multimedia communication. Authors in [2] and [7] used Petri Nets to simulate a complex military simulation system with HLA, to manage its concurrent properties and to specify the synchronization problems for the simulated commands. Author in [10] explores the impact of Time Service Management in HLA (HLA Time Management Service) 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. [8] presents an approach based on Colored Petri Nets to reduce the bandwidth usage for distributed simulations using HLA. In [9], the authors propose a colored temporal PN model for simulating the federation execution. The proposed model aims to assist developers of HLA simulations to design high-level simulation and to specify the constraints of the simulation.

However, as we outlined in our previous work [3] also with other related works, these approaches do not provide a structured model, and do not provide a comprehensive qualitative analysis of the simulation. Furthermore, no quantitative analysis was presented, particularly when specifying the temporal constraints and the performance analysis of information exchanged during the simulation. As a consequence, in this paper we use the same Hierarchical Time Stream Petri Nets formalism 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.

Our contribution, is an extension this previous works, but with another validation tool, uses primarily a temporal model because it induces a required flexible management of system resources and allows expressing of the non determinism that may occur when a time 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. That is, we can find a tradeoff between two targets: synchronization of stream to reduce the end-to-end latency and eliminating delay jitter. Hence, we aim to improve those QoS parameters and we add real-time scheduling approach for stream synchronization.

III. Hierarchical Time Stream Petri Nets

The HTSPN [4] (see also the HTSPN formalism in our previous work cited in Section II) 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. 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 [5], timed links allow the modeling and the formal specification of the transversal semantics of the application layer.

Composite Component: the 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.

 ISBN: 978-1-61208-129-8

IV. QOS REQUIREMENTS IN DISTRIBUTED MULTIMEDIA SIMULATION

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. The intra-stream synchronization ensures the compliance with the time constraints of the timing units for each stream. The synchronization level is given for each flow by the temporal validity intervals (nominal delay, allowable jitter) of each synchronization unit. The intra-stream jitter is given by (1) and illustrated in Figure 2. $\tau(n)$ is the arrival time of object *n* and the maximum allowable jitter intra-flow (equation (2)) is then $2\times\epsilon$'.

$$\epsilon'_{min} \leq \tau(n-1) - \tau(n) \leq \epsilon'_{max} \quad (1)$$

τ(n) is an intra-flow object presented at time n, 2*ϵ' is the intra-flow allowable jitter.

$$\begin{aligned} -\epsilon'_{min} &= T' - \epsilon' && (2) \\ \epsilon'_{max} &= T' + \epsilon' && (3) \end{aligned}$$

To ensure the receipt of *n* objects within a time interval,

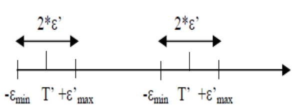

Figure 2. Jitter in intra-stream

one has to guarantee the constraints of the quality of service of the intra-flow, i.e. the maximum value of the global jitter of the synchronization of n objects is given as the sum of all jitters that exist between all consecutive units (2*ϵ per period).

For the intra-stream synchronization, the QoS requirements, that should be satisfied for instance when an audio and a video streams need to be synchronized, depend on the communication variability. For instance, at the receiver site, if two units of two different flows arrive at 2 different times t1 and t2, the correctness of their synchronization has to be deduced from the specification and the presentation constraints: the synchronization scheme should provide the acceptable interval for synchronized units of the flows, and should define some actions to eliminate the streams discontinuities. As an example, if a flow is behind the other(s) (is late) de-synchronization will occur and the discontinuity may become visible (when sound is no more synchronized with video, this problem is called "Lip-Synchronization). Relation (3) and Figure 3 specifies a periodic traffic, with period T, and an inter-flow jitter equal to 2*ϵ for one period.

$$\begin{aligned} \epsilon_{min} &\leq \tau(x1, x2) \leq \epsilon'_{max} && (4) \\ \epsilon_{min} &= T - \epsilon && (5) \\ \epsilon'_{max} &= T - \epsilon && (6) \end{aligned}$$

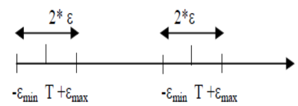

Figure 3. Inter-stream Jitter

It is clear from the above equations that the flows must be sent periodically. In particular, the packet size is a very important criteria 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. Table I gives the packet size with respect to the data that has to be transmitted through a network. it shows how audio data should be prepared and sent over Networks. Column 1 presents the sampling frequency that produces the audio data. Column 2 gives the time necessary to produce an IP packet (1518 Bytes). For example, with a 8 KHz sample frequency, the time required to produce this packet is 189 ms. In order to fulfill the QoS requirements for distributed media application, this delay need to be short enough because it delays the packets and implies the quality of the interactivity and of the presentation at the receiver side. Then, Columns 3, 4 and 5 show the size of the frame, given an interval of time, to satisfy the quality of the presentation. It seems that a delay of 20 ms is for sure a correct value because it sends a high audio delay quality with the respect to the frame size. The requirement of low latency means that it is better for the senders to send small packets frequently rather than large packets seldomly. 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

Table I
RELATIONSHIP BETWEEN PACKET SIZE AND PROCESSING TIME

Frequency (Khz)	Time (ms) IP Packet	Frame Size in 50ms	Frame Size in 30ms	Frame Size in 20ms
8	189	400	240	160
11	69	1101	660	440
22	34	2201	1321	881
44	17	4403	2642	1761
96	8	9606	5764	3843

also can be the propagation delay in the network. Distributed multimedia applications are not only presentation driven; they are also data-driven.

Therefore, a formal model must describe both the data and their presentation. In addition, the model must provide means for representing the logical and temporal compositions of their interactions. In order to specify the best choice for an audio packet size, Figure 4 gives the functional point able to satisfy our interactive requirements. It displays the variability of the packet size with respect to the time needed to produce the packet. These curves provide the Temporal Validity Interval (TVI) which will be used in our formal model. It follows that the best value of TVI is given by [15, 20, 25], where 20 ms is the nominal time to produce a packet, and there is a maximal drift per period of 2*5=10 ms.

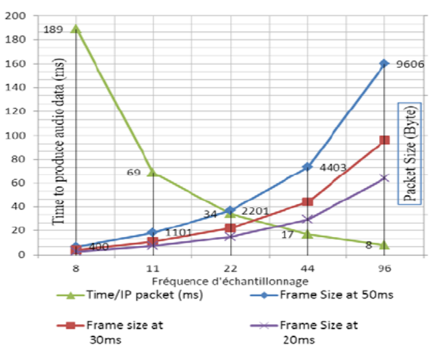

Figure 4. Variability of packet size with the processing time

The first curve in Figure 5 indicates the time required to produce a 1518 Byte audio data frame. From the left, this curve shows for example that a voice corresponding to a sequence of samples of 8 bits at a frequency of 8 kHz leads to produce a sample every 125 ms; for a packet size of 1518 Bytes, the sender must wait 189 ms to produce and start to send only this packet. This value does not fulfill the QoS requirements needed to transmit the packets of interactive applications. A 22 kHz sampling allows a processing and production time equal to 34ms. However, it does not lead after these 34 ms to a packet that exceed the proposed maximum size of 1518 Bytes, it is about 2201 Bytes. As said before, we selected 50 ms as the maximum time allowed to produce a packet at the sender side, and as a consequence it is clear in Figure 4 that it is not possible to send a full IP packet. If a 22 KHz sampling frequency would have been selected, it would have fulfilled all temporal and length QoS requirements: the delay to produce the audio data frame is 20ms for a packet size of 881 Bytes, but the packet size is rather short. Using the 50 ms values lead to start the specification of the formal model.

Notice 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 results 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, for example first by ensuring the synchronization in each streams, second between the different multimedia streams, and, third by ensuring the synchronization between these multimedia flows and the control flows of the distributed interactive simulation.

V. FORMAL MODEL OVER HLA-RTI

Basically, the application (Figure 5) is a platform for distributed interactive simulation, and it allows end users to interact by voice, video and distributed simulation events sent in real time. Such an application consists of at least three streams: the audio and video streams captured by a camera and the flow coming from the modification of the virtual environment of the distributed simulation. The synchronization scheme considered involves three types of flow synchronization:

- Intra-stream synchronization between the objects of each flow
- Inter-stream synchronization between the audio and video streams to meet the timing constraints often called Lip- Synchronization.
- Inter-stream synchronization between the two (audio / video) streams and the control stream of the distributed interactive simulation.

The intra-stream synchronization considers one flow, the inter-stream synchronization considers all flows, and specifies the acceptable inter-stream drift. The constraints of intra-stream synchronization which must be verified for each flow are:

- Units have an audio synchronization nominal duration of 20 ms by assuming a jitter of 5 ms. That is to say, the temporal validity interval of each unit of the sync audio is [15, 20, 25].
- The video synchronization unit has 40 ms as a nominal duration and a jitter of 10ms. The synchronization interval validity of the video is then [30, 40, 50].
- The synchronization units of the distributed interactive simulation flow have a nominal duration of 20 ms wit a 5 ms jitter. The temporal validity interval of this flow is then [15, 20, 25].

The corresponding HTSPN synchronization model is defined by a three levels representation: the link level considers

 ISBN: 978-1-61208-129-8

the application level, and depends on the developer choices (the application reference is given in Figure 6. The temporal validity interval, [60, 80, 100], at this layer corresponds to the inter-stream synchronization and will be explained later on. It means that the transition will be fired in the interval min, max=[60, 100], the time being started when the transition is enabled, i.e. when the places have all one token.

Thus, knowing that the sound has to be produced and sent in less than 20 ms (Figure 4), and that the image in less than 40 ms (given by the application), we measured the processing of the interactive event: it has been found to be 16. The delay of the interactive flow must be driven by the audio stream because the audio media is the most time sensitive one, and the audio stream will be then selected master stream: it control the time schedule for the firings of the transitions. Therefore, the number of places of this stream must be a multiple integer of the number of the video and simulation units. As a consequence, the synchronization transition will be defined at the rendezvous which occurs at a period equal to the LCM (Least Common Multiple) of the nominal durations of the three streams, i.e. at time equal to 80 ms, the LCM of (16, 20, 40). 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 10ms: this drift is

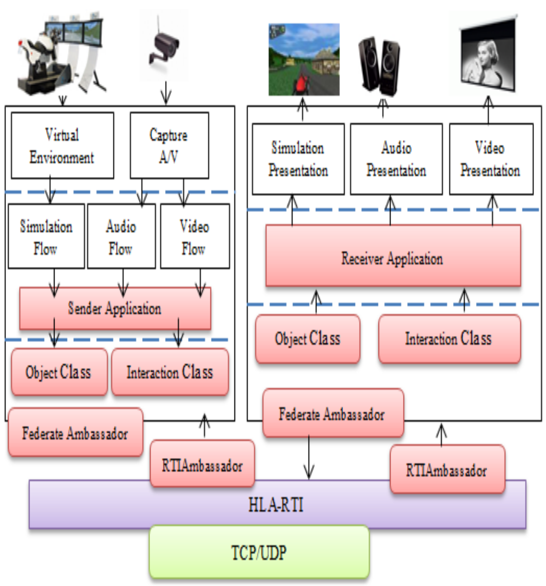

Figure 5. Platform used for the formal specification

achieved by the treatment of 2 units of synchronization, which is the treatment of 4 units of sync audio and 5 units of flow synchronization Interactive. The formal modeling of this approach is given by three hierarchical levels and 5 HTSPN nets. Figure 6.a shows the highest level. This highest level specifies the full constraints of the inter-stream synchronization between the audio stream, modeled by Aud, the flow of the interactive simulation, modeled by the Sim, and the video stream, modeled by VID.

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. Figure 5 describes the synchronization architecture of the distributed interactive simulation governing the HLA-RTI middleware. Within the composite layer, Places, from Sim1 to Sim5, represent the objects of the distributed simulation, AUD1 to AUD4 represent the audio objects and places VID1 and VID2 describe the video objects. Each circle represents a data packet: to ensure the synchronization between the streams, at the model defines a synchronization for each set of 5 packets of distributed simulation (i.e. 544 bytes per packet), of 4 audio packets (of size 881 byte packet) and of 2 video packets. The composite layer fulfills this inter-stream synchronization and prepares the link synchronization Layer.

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 30ms compared to the other flows. This control must be applied with a maximum granularity of 20 ms, corresponding to two units of video synchronization, 4 units of sync audio and 5 units of sync interactive flow. The purpose of this architecture is to express all the specified timing requirements. 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 for the flows. The interactions are persistent and will be able to natively transport the flows between the Federates. Finally, the places Sim1 to Sim5 represent the objects of the distributed simulation.

As described in Figure 6, the first point of inter-stream synchronization is of type "MASTER", with the audio stream as "MASTER" is placed at the point go after a nominal duration equal to 80ms (100ms maximum). This synchronization is likely to induce the acceleration (respectively deceleration) of the video and audio flows after 5 units of synchronization and can also cause a delay or the loose of the video stream. The abstract place *Sim* is specified by the subnet shown at the top of Figure 5. This HTSPN model controls explicitly the advance of the interactive simulation flow with respect to the other flows. Given the jitter units of 10 ms for the video and of 5 ms the audio stream, then after 5 intra-synchronized objects of

 ISBN: 978-1-61208-129-8

the interactive flow, this stream can be up to 20ms ahead of 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(T1,T2) should be used with two arguments that are the minimum and the maximum values used in the temporal validity interval; for example tick(12,20) has to be used.

VI. MULTIMEDIA TRANSPORT ARCHITECTURE

Sender and receiver are involved in the stream transmission. As a requirement of the HLA-RTI middleware, both participants are federates and should follow the HLA rules in order to be compliant with the specification. Hence, RTI supports both "Reliable" and "Best Effort" communication mode. Since Multimedia stream need to be send continuously, it is necessary to optimize the throughput and the reduce the end-to-end latency. This solution need UDP-based "Best Effort" transport protocol.

As we outlined in Section IV, multimedia packets need low latency to meet the QoS requirements, therefore it is mandatory to schedule a stream transmission task in order to share the system resources with other tasks. The synchronization interval validity are used to meet the requirements of the schedule system interval timer provided by the underling operating system. The interval timer allows the application to schedule periodic timer events. Thus, the application receives and requests timer messages at the Temporal Validity Interval (TVI) given in each arc of the HTSPN model- that is, the TVI allows the application to schedule the timer events within the TVI resolution, that is the timer interval of the *upadeintercation()* and *sendInteraction()* function is caller in this regular time resolution. In fact, real-time stream transmission over large scale networks adds latency and jitter due to the router scheduling and admission control within the router queues. Using the the value admissible in the TVI is twofold:(1) the re-synchronization of the media frames in the presentation layer at the receiver side without using reliable stream control (TCP protocol), the end-to-end latency can be carefully controller before the stream being displayed, and (2) allows the receiver buffer handling the received stream with minimum frame lost and eliminates jitter issues. Likewise, The longer the reconstruction buffer is, the larger the jitter can be reduced.

VII. CONCLUSION

We have presented a formal model based on Hierarchical Temporal Stream Petri Nets for the synchronization of distributed interactive multimedia systems. This model is able to describe applications implemented using an HLA distributed simulation. It offers a good modeling power for at the same time the expression and the analysis of temporal constraints in such systems. It also 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.

Taking into account all these constraints early in the design process leads to a rather efficient development of distributed applications and reduces the cost of this development. Our future work is to design and implement by this model a full distributed application that has been developed to remotely teach car drivers.

REFERENCES

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

[2] P. Senac et al., *Modeling logical and temporal synchronization in hypermedia systems*, IEEE Journal on Selected Areas in Communications, Vol.14, N1, pp. 84-103, January 1996

[3] A. Hakiri and al., *Multi-level Model for Synchronizing Temporal Streams on HLA-Based Distributed Multimedia Applications Using HTSPN*, Second International Conferences on Advances in Multimedia (MMEDIA), 2010, pp. 140-147

[4] M. Diaz and P. Senac, *Time Stream Petri Nets a model for timed multimedia information*, in *Petri Nets: Fundamental models, Verification amd Applications*, ISBN 978-1-84821-079-0, 2009

[5] R. Willrich et al., *Multimedia Authoring with Hierarchical Timed Stream Petri Nets and Java, Multimedia Tools and Applications*, Journal of Multimedia Tools and Applications, Volume 16 Issue 1-2, January-February 2002

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

[7] L. Xie et al., *Complex System Simulation Based on Petri Net Combined with HLA*, 1st International Workshop on Education Technology and Computer Science, 2009. vol. 3, pp .205-208.

[8] M. B. Kpatcha et al., *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, pp. 143-166

[9] S. Combettes and A. Nketsa, *Interoperability Compliance Validation Of HLA Federations Based On Colored Petri Nets*, 2003 EURO SIW 03E-SIW-084.

[10] R. Guha, M. Bassiouni, and G. Schow, *A Framework For Modeling High Level Architecture (Hla) Using Petri Nets*, University of Central Florida. Department of Computer Science. Orlando, FL 32816

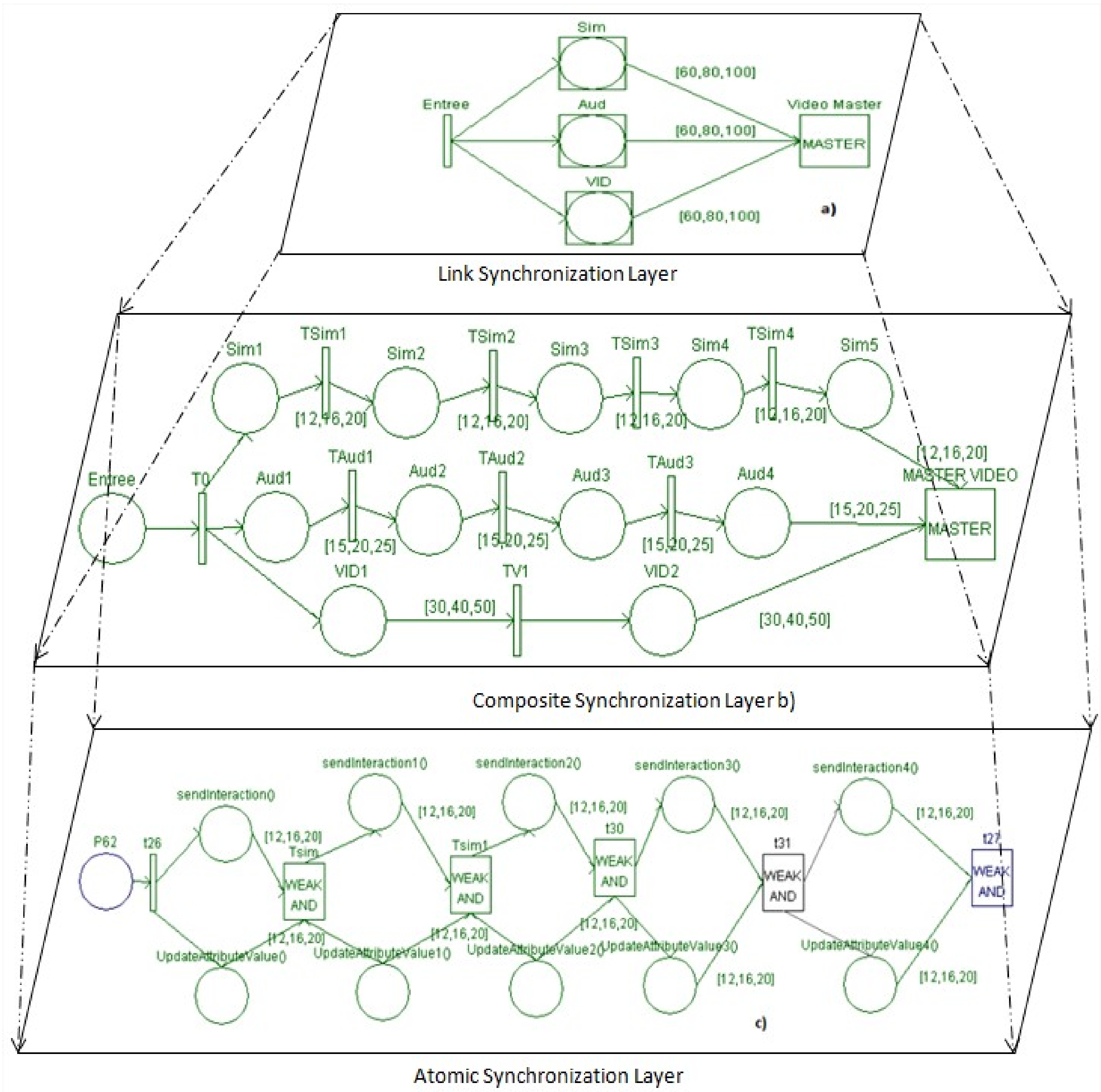

Figure 6. Distributed Multimedia Synchronization scheme over HLA-RTI. a) The link synchronization Layer, directly connected to the application, b) the composite synchronization layer for the inter-stream synchronization Layer and c) the atomic synchronization layer which of the intra-flow synchronization