Towards Hypermedia Documents Design

Roberto Willrich[1]○, Patrick Sénac○•, Pierre de Saqui-Sannes○• and Michel Diaz○
○LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse Cédex, France
•ENSICA, 1 Place Emile Blouin, 31056 Toulouse Cédex, France
Email: {willrich, diaz}@laas.fr, {senac, pdss} @ensica.fr
Phone: (+33) 61.33.62.00 - Fax: (+33) 61.33.64.11

Abstract

The design of hypermedia documents is a non trivial task, where several pieces of information stored in a distributed system are to be integrated and presented together. Specifying hypermedia documents by means of informal or semi-formal models may be a misunderstanding source and do not allow the use of standard analysis techniques to ensure the consistency of hypermedia systems. This paper presents a formal methodology for the development of hypermedia documents. This methodology is based on the HTSPN model (Hierarchical Time Stream Petri Net) which provides facilities for the accurate and unambiguous specification of hypermedia documents. HTSPN further permit application of powerful analysis techniques for insuring hypermedia quality. Finally, the implementation of hypermedia documents is supported with the automatic production of ISO MHEG representation. The latter can then be stocked, exchanged and presented within open hypermedia systems.

1 Introduction

A hypermedia document can be defined as a set of pieces of information, called *components*, which are combined at run-time. Components are possibly made up of different media types (*e.g.*, audio, video, text) and distributed over several multimedia servers. They are organized in a network of nodes interconnected by links. A node that contains one piece of information is called an atomic component (or monomedia node), and a node that contains several pieces of information logically and temporally integrated by a presentation scenario is called a composite component (or multimedia node). Links specify relations between nodes, thus defining that define the browsing semantics of the document.

The huge success of the World Wide Web underlines the utility and the social, cultural, and commercial contributions of hypermedia as a privileged tool for information access and retrieval. However, designing user-friendly, robust, correct and efficient hypermedia systems is a non trivial task. Indeed by tackling large information spaces, hypermedia systems introduce paradoxically the important problems of user disorientation and information overload [Conklin, 87]. Moreover, the design complexity of hypermedia systems is increased by the need for an explicit notion of time for the management of multimedia scenarios [Hardman, 94].

Specifying hypermedia documents by means of informal or semi-formal models may be a misunderstanding source and do not allow the use of analysis techniques to ensure the consistency of hypermedia systems. Conversely, formal models allow the construction of an accurate and unambiguous semantics for hypermedia. Moreover, formal models allow powerful analysis methods (including validation and verification techniques) to be applied before implemen-

---

[1] Supported by a CNPq/Brazil grant.

tation. Additionally, formal models open the way to the formal design of hypermedia systems with the help of hypermedia run-time, making it possible to interpret formal specifications [Coulson, 92] or by compilation towards standardized hypermedia documents structures [Willrich, 96], such as MHEG [ISO 13522] or HyTime [Newcomb, 94].

Several work addressed the formal modeling of hypermedia ([Stotts, 90], [Halasz, 94]) or multimedia ([Little, 90], [Diaz, 94]) systems. However, to our knowledge, there is no formal model that allows to fully and easily express the structures of hypermedia systems through a unified approach ([Sénac 95a], [Sénac, 96]). Taking note of this lack, [Sénac 95a] [Sénac, 96] propose a new model, called Hierarchical Time Stream Petri Nets (HTSPN), aimed to this purpose.

With the Dexter hypertext reference model [Halasz, 94] and the TSPN (Time Stream Petri Nets) model [Diaz, 94] as background, the Hierarchical Time Stream Petri Nets model enables a unified, complete and accurate specification of temporal and logical (intended to conventional link based navigation) synchronization within hypermedia systems. The HTSPN model is a powerful formal method for the specification, verification and prototyping of logical and temporal constraints within hypermedia documents. However, the HTSPN model only addresses logical and temporal synchronization issues; it does not aim at the complete description of hypermedia documents.

This paper proposes a formal framework for the specification, analysis and generation of MHEG hypermedia documents. The starting point is an interpreted[2] version of the HTSPN model [Willrich, 96], I-HTSPN in short, which provides means to specify completely hypermedia documents in an accurate and unambiguous fashion. In order to generate an information structure that can be stocked, exchanged and presented in an open hypermedia system, this paper discusses the translation of an I-HTSPN specification of hypermedia documents into an MHEG representation.

The use of interactive development environments will simplify the authoring task of MHEG hypermedia documents, using for instance graphical editors and analysers. A HTSPN toolkit is being developed. This toolkit will permit the authoring of MHEG hypermedia documents (or document parts) that can be transferred to MHEG databases. These MHEG representations can then be directly used by remote presentation terminals in an open hypermedia system. Each presentation terminal must contain an MHEG engine that is responsible for the interpretation (presentation) of MHEG representations. Figure 1 illustrates the application of the proposed approach for the development of hypermedia documents in a distributed hypermedia system.

The paper is organized as follows. Section 2 identifies the main issues in hypermedia documents design. Section 3 presents our formal development methodology for hypermedia documents. A HTSPN toolkit for hypermedia documents specification and automatic generation of MHEG representation is described in Section 4. Section 5 presents an example of hypermedia document design. Conclusions are finally given in Section 6.

Design of Hypermedia Documents

The main issues in hypermedia documents design are as follows: how the components of a document are structured, how these components are accessed (for instance, the access and necessary information for the data manipulation) and showed to a user (*i.e.*, description of the presentation characteristics), and how the document will be represented in an interpretable or executable code (to be processed by a hypermedia system).

---

[2] Interpreted means that a semantics is given to places and transitions.

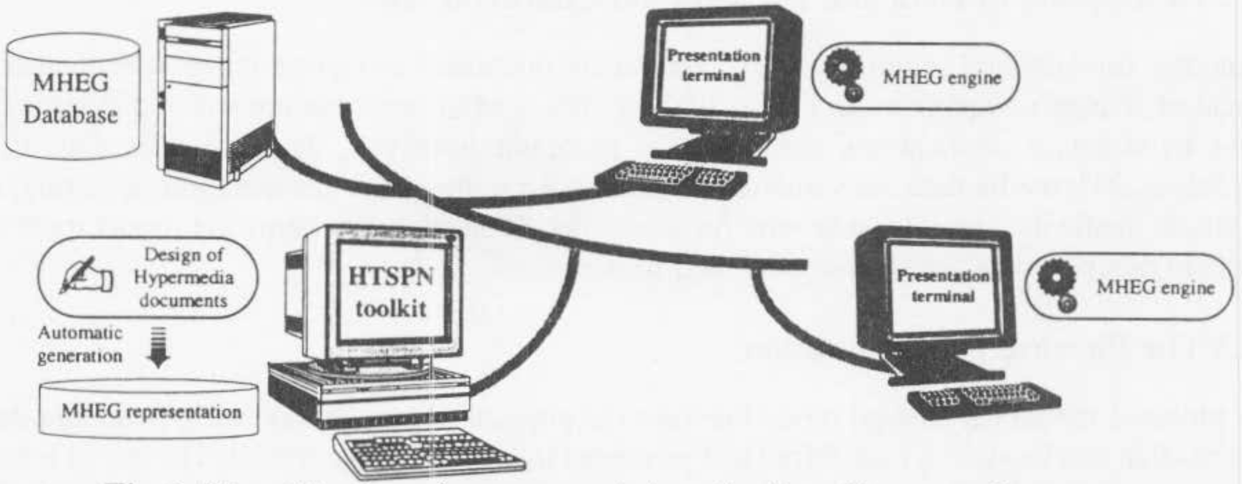


Fig. 1. The utilization of our approach in a distributed hypermedia system

2.1 Hypermedia Documents Structuring

Three fundamental conceptual structures can be identified in hypermedia systems: (i) logical structures that specify, with the help of encapsulation mechanisms, monomedia and multimedia nodes; (ii) temporal structures that specify the temporal behaviour of multimedia nodes (*i.e.*, multimedia scenarios); and (iii) semantic structures that specify links between the nodes of a document.

Logical Structure

As previously mentioned, a hypermedia document is a collection of components (*i.e.*, pieces of information). The logical structuring allows a group of these components to be created and treated as a single object, named a *composite*. For instance, a set of components that express together an idea can be gathered in a composite representing the idea, which prevents from manipulating the individual components. The logical structuring is also useful for building up complex presentations from smaller information groups. For example, a document can be structured in chapters subdivided in sections. In brief, logical structuring introduce the benefits of modularity, encapsulation and abstraction mechanisms.

Temporal Structure

Temporal structures aim to model temporal and synchronization scenarios that specify the dynamic behaviour of components within hypermedia documents. Temporal structures modeling address the following issues:

- Modeling arbitrarily complex intra- and inter-media synchronization schemes. The intra-media synchronization schemes aim to specify those synchronization constraints which must be satisfied by information unities inside each component of the document (*e.g.*, video frames, audio samples, *etc.*) independently of the other components. The inter-media synchronization describe synchronization schemes between several components.
- Modeling temporal non-determinism in intra- and inter-media synchronization schemes. This temporal non-determinism results from the asynchronous or weakly-synchronous behaviour of distributed hypermedia systems.

Semantic Structure

The semantic structure of a hypermedia document is described by means of links that permit the definition of semantic relations between the components of a document.

2.2 Description of Data and Presentation Characteristics

Another fundamental requirement of hypermedia document modeling is the specification of local or remote components and the way these pieces of information are showed. Multimedia and hypermedia applications are not only presentation-driven, they are also data-driven [Schloss, 94]: media data are sometimes manipulated without any presentation occurring; and a single media data may have several presentations. Therefore, a hypermedia model must separately describe data information and their presentation.

2.3 The Physical Representation

A physical model (or storage model) defines the physical representation of hypermedia documents that can be stored, transferred and presented in a hypermedia system. The use of international standards for hypermedia and multimedia interchange formats, such as MHEG [ISO 13522], HyTime [Newcomb, 94] and HyperODA [Appelt, 93], as the final storage model of multimedia documents is greatly recommended. Hypermedia documents can thus be transferred in an open distributed system and presented by a hypermedia system that supports the standardized representation.

Due to the complexity of the involved representation, it seem unrealistic to directly describe hypermedia documents by means of such international standards. Therefore, a logical model is required to make the description of hypermedia documents easier. Additionally, a compilation step can help to translate[3] this logical description into an interpretable code defined by the physical model.

3 A Formal Methodology for the Design of Hypermedia Documents

In this paper, we propose a simplified formal methodology for the development of hypermedia documents which consists of the following steps (figure 2):

(A) Use of the Interpreted HTSPN model for the hypermedia document specification.

(B) Application of analysis techniques on the I-HTSPN specification: to check time inconsistencies of the modelled multimedia scenarios; to detect conflict on the use of shared resources; and to simulate the specification in order to check the logical and temporal behaviour correctness of the document presentation. If some errors exist then go to first step.

(C) Automatic translation of the I-HTSPN specification into an MHEG representation. If the specification is incomplete for the MHEG translation then go to first step.

(D) Final test of the hypermedia document in a real presentation using an MHEG engine. During the presentation, if the behaviour of the document is not that expected, then go to first step.

The following sections present the three main steps of the proposed methodology: the formal specification of the hypermedia document, the analysis of this specification, and the automatic translation of the analysed specification into an MHEG representation.

3.1 The Formal Specification and Analysis of Hypermedia Documents

Hypermedia results from the combination of hypertext and multimedia. Several papers have separately addressed the formal modeling of hypertext ([Stotts, 90], [Halasz, 94]) and multimedia ([Little, 90], [Diaz, 94]). In particular, the Dexter hypertext reference model [Halasz,

---

[3] An automatic or semi-automatic translation.

94] and the Time Stream Petri Net models (TSPN) ([Diaz, 94], [Sénac, 94]) provided advanced foundations for hypertext and multimedia modeling, respectively. Indeed, the Dexter model introduces the basic concepts for defining logical and semantic structures within hypertext/ hypermedia documents and the TSPN model allows a complete and accurate specification of temporal and synchronization constraints in multimedia systems.

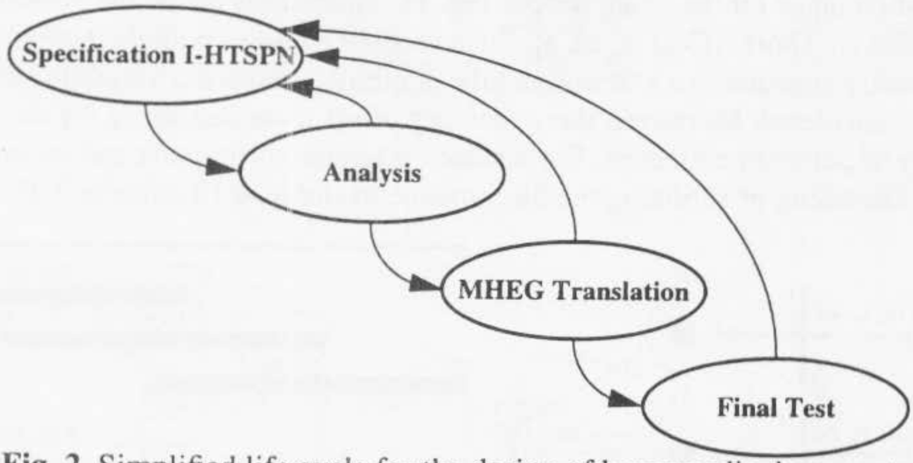


Fig. 2. Simplified life cycle for the design of hypermedia documents

Starting from the above considerations, [Sénac, 95] introduced a unified formal model, called Hierarchical Time Stream Petri Net model (HTSPN), which extends the TSPN model with the logical and semantic structures introduced by the Dexter model. Thus, the HTSPN model enables a complete and accurate specification of the logical, temporal and semantic structures of hypermedia documents. However, the HTSPN model was not designed for the complete formal specification of hypermedia documents. For instance, HTSPN does not provide means to specify the access information of media data or their spatial and audible presentation characteristics. [Willrich, 96] proposed an interpreted version of the HTSPN model, providing means to completely specify a hypermedia document in a final form and to allow the automatic generation of an MHEG representation.

The proposed methodology for the design of hypermedia documents uses an MHEG interpreted version of the HTSPN model (*i.e.*, the I-HTSPN model) for the specification and analysis of documents. Therefore, this methodology uses the TSPN model for the description of temporal structures of hypermedia document, the HTSPN model to include the description of the logical and semantic structures, and the I-HTSPN model to incorporate data information descriptions their presentation characteristics.

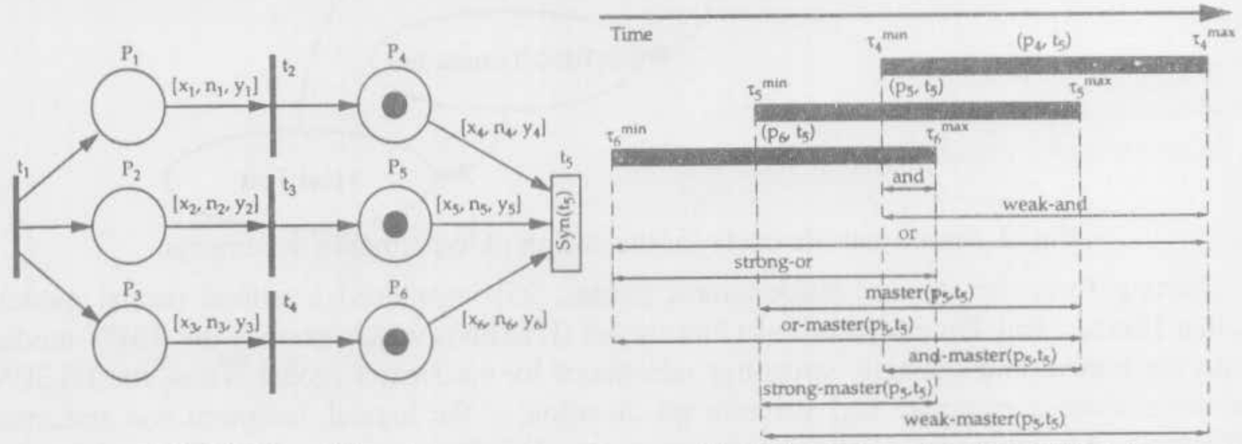
3.1.1 Modeling temporal structures using the TSPN model

This section presents a brief introduction to the TSPN model and shows how the temporal structures of hypermedia documents are modelled and analysed.

The TSPN Model

The Time Stream Petri Nets model (TSPN) ([Diaz, 94], [Sénac, 94]) has been initially used for the modeling of multimedia synchronization scenarios within distributed weakly synchronous systems. This formal model considers the temporal nondeterminism of weakly synchronous systems which entails temporal drift in intra-media and inter-media synchronization schemes (caused by network delays, access delay to databases, *etc.*).

In TSPN, processes and their temporal characteristics are represented as arcs (and related places), called Synchronization Units (SU), associated with temporal intervals. These temporal intervals, called Temporal Validity Intervals (in short TVI), are 3-tuples [x, n, y], where x, n and y are respectively the minimum, nominal and maximum admissible durations of the related processing. For instance, arc $a_1 = (P_1, t_2)$ in figure 3a can represent a video presentation with $[x_1, n_1, y_1]$ as its TVI. Let us consider τ the marking date of P_1 (*i.e.*, the arrival date of a token in this place and thus the start of video presentation), then the 3-tuples $[\tau + x_1, \tau + n_1, \tau + y_1]$, also denoted $[\tau_1^{min}, \tau_1^{nom}, \tau_1^{max}]$, specifies respectively the minimum, nominal and maximum termination dates for the video processing. This interval is called the Absolute Temporal Validity Interval (in short ATVI) of arc a_1. Such modeling allows both the temporal non-determinism of weakly synchronous systems and the admissible temporal variability of multimedia objects to be considered. Moreover, this capability offers more flexibility for the management of the Quality of Service parameters. For instance, a temporal formatter can optimize a temporal layout by stretching or shrinking media components duration [Buchanan, 93].



(a) TSPN specification (b) Firing Intervals of transition t_5

Fig. 3. Synchronization semantics of TSPN

In synchronous systems (*i.e.*, systems with no temporal jitter), the TVI are reduced to one point (*i.e.*, [n, n, n]). Conversely, in asynchronous systems the TVI corresponds to unbounded intervals (*i.e.*, $[x, n, \infty]$). Therefore, the notion of weakly synchronous system, expressed with the help of TVI, offers a generic framework that covers the full range of systems from synchronous to asynchronous ones.

The temporal drifts entailed by the temporal nondeterminism of weakly synchronous systems make impossible, in the general case, to meet the strict temporal constraints of the whole processing involved within a synchronization scheme. For instance, let us consider the synchronization scheme among three multimedia streams[4] illustrated in figure 3a. Assume that considering only nominal durations, this synchronization scheme is a strictly parallel one (*i.e.*, $n_1 + n_4 = n_2 + n_5 = n_3 + n_6$). Introducing the processing temporal variability, it appears that this strict parallel synchronization scheme cannot be ensured any more. In the worst case, in function of cumulative delays in upstream processing (*i.e.*, arcs (P_1, t_2), (P_2, t_3) and (P_3, t_4)) the desynchronization between arcs (P_4, t_5), (P_5, t_5) and (P_6, t_5) could be so important that the intersection of their ATVI becomes empty (*i.e.*, $\bigcap_{i=4}^{6} [\tau_i^{min}, \tau_i^{max}] = 0$).

The above observation leads us to propose three basic synchronization strategies (*i.e.*, firing rules) that privilege a statically or dynamically defined SU:

- a dynamic synchronization strategy, called *strong-or*, driven by the earliest processing (*i.e.*, the first arc that gets the maximum bound of its ATVI);
- a dynamic synchronization strategy, called *weak-and*, driven by the latest processing (*i.e.*, the last arc that gets the maximum bound of its ATVI);
- a static synchronization strategy, called *master*, driven by a selected processing (*i.e.*, only the ATVI of the selected processing is taken into account).

---

[4] Using TSPN, a stream is modelled as a sequence of timed arcs.

These 3 fundamental strategies entail 9 firing rules obtained from a consistent and complete combination of the ATVI of the SU associated with an enabled transition (figure 3b). These firing rules define firing intervals that cover all the possible synchronization instants. Therefore, temporal drifts between multimedia streams can be accurately and fully controlled with the help of those 9 different firing rules that can be selectively associated with each transition of a TSPN.

A TSPN is formally defined as follows:

Definition 1. A **TSPN** is a tuple (P, T, α, β, M_0, IM, SYN, MA), where:

- (P, T, α, β, M_0) defines a Petri Net, where P is a set of places, T is a set of transitions, β and α are respectively the backward and forward incidence functions (that define respectively arcs between places and transitions, and between transitions and places), and M_0 is the initial marking of the Petri net.
- IM is a function that associates an arc outgoing from a place with its TVI, and it is defined as follows:
  - Let A be the set of arcs outgoing from places: $A = \{a = (p, t) \in PxT \mid \beta(p, t) \neq 0\}$
  - IM: $A \rightarrow (Q^+ \cup \infty) \text{ x } (Q^+ \cup \infty) \text{ x } (Q^+ \cup \infty)$, $IM(a_i) \rightarrow [x_i, n_i, y_i]$, $0 \leq x_i \leq n_i \leq y_i$
- SYN is the typing function that associates a transition with its firing rule[5] (*i.e.*, synchronization semantic), and it is defined as follows:
  - SYN: T $\rightarrow$ {*and*, *weak-and*, *or*, *strong-or*, *master*, *or-master*, *and-master*, *weak-master*, *strong-master*}
- MA: $T_m \rightarrow A$ is a mapping relation between *master* type transitions and their related master arcs, where:
  - $T_m = \{t \in T \mid SYN(t) \in$ {*master*, *or-master*, *and-master*, *strong-master*, *weak-master*}$\}$

Modeling Temporal Structures of Multimedia Scenarios

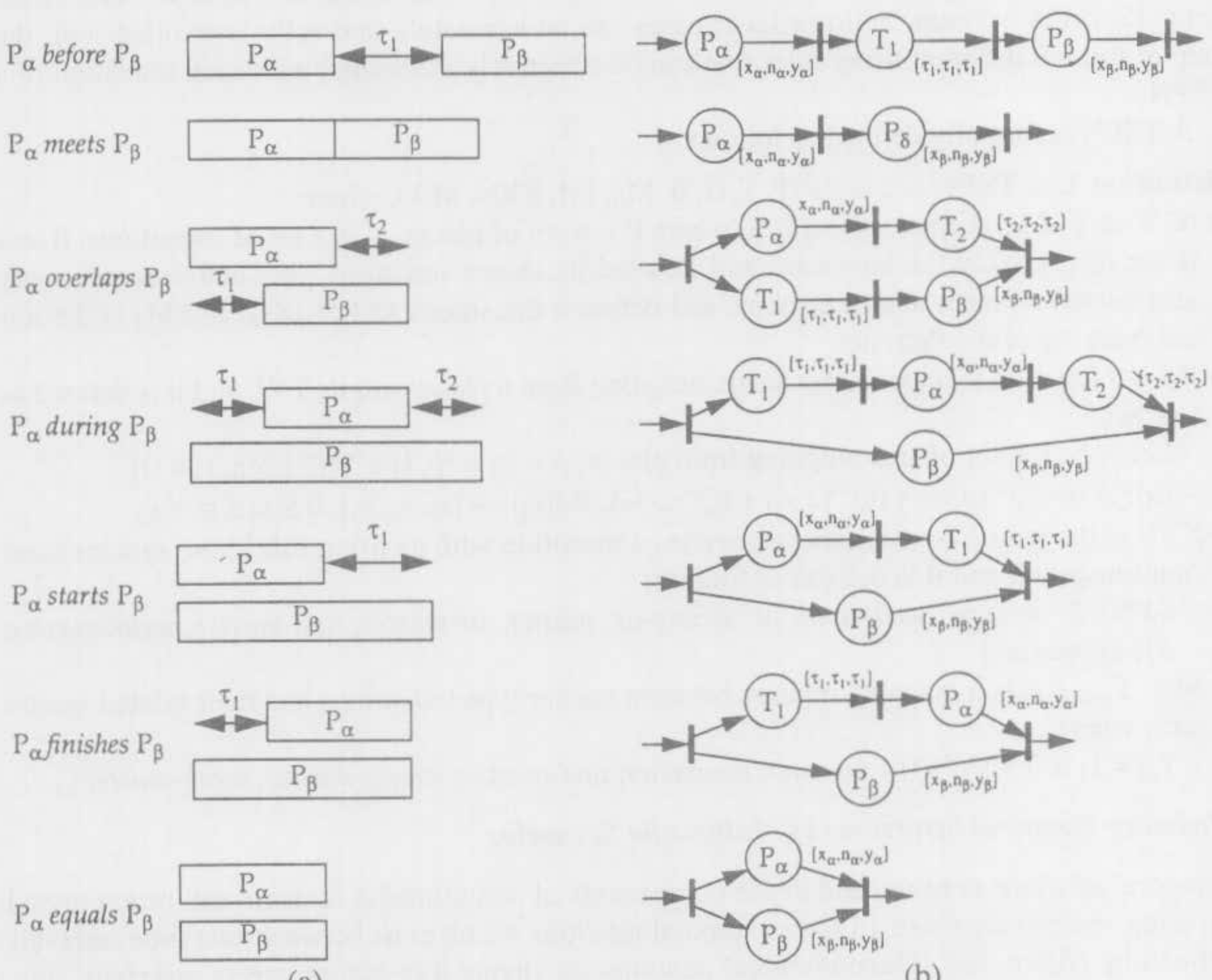
Temporal relations between and inside components of a multimedia scenario can be expressed by using recursively those 13 basic temporal relations which exist between two time intervals defined by [Allen, 83]. These temporal relations are (figure 4a): *before*, *meets*, *overlaps*, *during*, *starts*, *finishes* and *equals*, plus the inverse relations (except equal). These temporal relations can be easily expressed using TSPN, as presented in figure 4. The seven synchronization constructs in figure 4b can be recursively used to specify arbitrary complex temporal and synchronization multimedia scenarios. The so obtained TSPNs are called Structured TSPNs (STSPNs).

Such a structured approach has been used in [Little, 90] with Timed Petri Nets (durations associated to places) and leads to a multimedia interpreted Timed Petri Net model, called the Object Composition Petri Net (OCPN) model. However, in OCPN a presentation has a nominal duration. Using a perfect duration may be seen as «unrealistic» in the domain of distributed weakly synchronous hypermedia systems [Diaz, 94]. Temporal intervals, used by the TSPN model, are better appropriate to express temporal variability and impreciseness.

As they exclusively use the above seven synchronization constructs, STSPN nets have a circuit-free Marked Graph structure and are also defined as safe nets [Sénac, 94]. Therefore, the structure of STSPN allows powerful verification techniques to be applied for checking the temporal consistency of synchronization schemes. These verification techniques are based on a places/transitions reduction algorithm, applied to the original STSPN, followed by a worst case analysis of the resulting STSPN [Sénac, 95]. Moreover, the notion of state of TSPN allows the dynamic behaviour of systems modelled with TSPN to be simulated.

---

[5] These 9 firing rules are equivalent for transitions with only one input arc. Therefore, in the graphic representation of a TSPN, we do not attach any specific firing rule to such transitions.



(a) (b)

Fig. 4. Temporal relations between two intervals

In summary, the TSPN model appears to be a powerful tool for the formal modeling of temporal structures within distributed hypermedia systems:

- it satisfies the requirements of temporal structures modeling identified in Section 2.1;
- its modeling power allows multimedia synchronization scenarios to be easily specified;
- powerful analysis techniques allow the state graph of bounded TSPN to be finitely computed;
- verification methods have been developed for checking the temporal consistency of STSPN;
- the notion of global state allows the dynamic behaviour of systems modelled with TSPN to be simulated.

3.1.2 Modeling Logical Structures of Hypermedia Documents

Logical structures in hypermedia documents aim to associate together several logically related multimedia components. The Dexter hypertext reference model introduces hypertext/hypermedia logical structures from the notions of *atomic and composite components*. Atomic components are related to an encoded data of a single medium. Composite components provide a hierarchical structuring mechanism based on the recursive composition of atomic and composite components. In the sequel, it is shown how the TSPN model has been extended for modeling atomic and composite components.

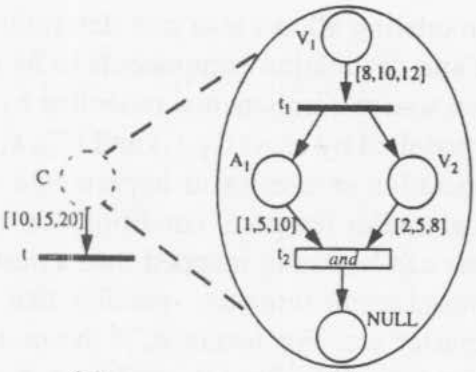
Modeling Atomic Components

The presentations of atomic components such as audio, video, images are submitted to intrinsic or application dependent presentation durations. Moreover, hypermedia systems are to be considered, in the general case, as distributed weakly synchronous systems in which atomic components are distributed over several multimedia servers and combined at run-time.

In weakly synchronous environments, it is unrealistic to use absolute timings. It is more realistic to use temporal intervals for specifying temporal constraints. Moreover, the use of TVIs indicates the flexibility given to the scheduling services to run the synchronization request and involves synchronization tolerances, making the hypermedia document more resource adaptable [Bulterman, 93]. This approach for specifying temporal variability of processing is allowed by the TSPN with the help of TVIs associated with arcs that represent multimedia processing. Therefore, using the TSPN model, an atomic component may be modelled as an arc with a TVI and an *atomic type place* associated with an atomic resource type.

Modeling Composite Components

Composite components are recursively built from atomic and composite components. Therefore, modeling composite components entails the need for hierarchical modeling capabilities. Basic Petri nets allow hierarchical modeling through abstract places representing subnets. This capability is illustrated in figure 5, where place C (figure 5a) is an abstract place representing a multimedia presentation scenario modelled by a STSPN (figure 5b). Therefore, modeling composite components with TSPN involves the use of a new place type, called the *composite type place*, that is an abstract place associated with an underlying subnet.



(a) (b)

Fig. 5. Modeling composite components

A composite place and its related subnet must not only be structurally equivalent (*i.e.*, the subnet must have an input place and an output place), but also temporally equivalent. This notion of temporal equivalence is introduced with the help of a reduction algorithm that allows to reduce every STSPN into one timed arc [Sénac, 95]. The TVI of the reached arc abstractly describes the temporal behaviour of the considered STSPN (*i.e.*, the minimum and maximum traversal durations of the STSPN). This property entails a relation of temporal equivalence between STSPNs, where two STSPNs are temporally equivalent if they can be reduced to the same timed arc. For instance, the STSPN of figure 5b is structurally equivalent to the composite place C and is temporally equivalent to arc (C, t).

The concept of composite type place is a unifying technique for modeling both logical and temporal structures in hypermedia systems. Let us note that this abstraction technique can be used both for specifying coarse discrete synchronization relationships within composite components (*i.e.*, inter-media synchronization relationships) and fine continuous synchronization relationships in atomic components (*i.e.*, intra-media synchronization relationships).

3.1.3 Modeling Semantic Structures of Hypermedia Documents

In the Dexter model, the semantic structure of a document is defined with the help of *link components*, that allow n-ary directed relations between parts inside several atomic and composite components to be specified.

We extend TSPNs to model a link as a timed arc *(L, t)* where *L* is a place of a new type called *link type place* (figure 6). Such a modeling offers several important advantages and helps clarifying the notion of hypermedia synchronization. The TVI associated with arc *(L, t)* allows the notions of timed and dynamic links to be modelled. Timed links must be activated inside a time interval and are able to automatically trigger link activation in function of time constraints. Timed links introduce the notion of active hypermedia systems [Stotts, 90]. Dynamic links are temporary links that appear and can be activated during the presentation of a given part of a multimedia component [Liestol, 94]. Untimed links (that is, the commonly used links in hypermedia systems) are modelled as arcs with $[0,*,\infty]$[6] as TVI.

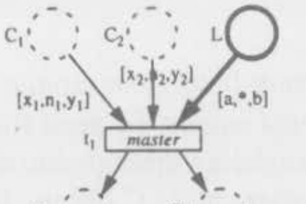


Fig. 6. Modeling of a multi-source and multi-destination link

The use of TSPN for link modeling allows also non-determinism (*i.e.*, multi-choices links) and links with several source and destination components to be easily specified. For instance, link (L, t_1), in figure 6, has two source components, modelled by arcs (C_1, t_1) and (C_2, t_1), and two destination components modelled by arcs (C_3, t_2) and (C_4, t_3).

The usual logical synchronization semantics in hypermedia systems lead synchronization schemes only driven by logical and/or temporal conditions on links. Using the TSPN model, such a synchronization scheme can be easily mapped into a *master* arc associated with a link type place. The *master* synchronization semantic specifies that the related transition is to be fired inside the TVI of the master arc. For instance, if the *master* arc (L, t_1), in figure 6, is enabled at time τ then transition t_1 must be fired inside $[\tau + a, \tau + b]$. If the link logical triggering condition has not been satisfied before time $\tau + b$, then the link is automatically triggered (*i.e.*, transition t_1 is fired) at time τ+b (if the transition t_1 is still enabled at this date). Note, that the link logical triggering condition is always considered as false before τ+a and as true at τ+b.

One of the advantages of such approach for link modeling is to offer a unified way for modeling link, atomic and composite components under the form of timed arcs composed of typed places (*i.e.*, atomic, composite or link type places). This approach greatly simplifies hypermedia modeling. Moreover, the 9 fundamental TSPN's synchronization semantics allow powerful combinations to be done between the temporal constraints of links and those of the related source multimedia components. This capability enlightens the notion of hypermedia synchronization by allowing to take into account the dynamic behaviour and the temporal state of multimedia components in the definition of link activation conditions.

---

[6] Note that, the normal duration of an asynchronous event cannot be known in advance. Therefore, the nominal duration of a link is replaced by the character «*».

3.1.4 The HTSPN model

The three previous sections have shown how temporal, logical and semantic structures can be modelled with a unified formal model by introducing some extensions to the TSPN model. This leads to the definition of a new model, called Hierarchical Time Stream Petri Nets (HTSPN), which has been defined for hypermedia modeling purposes[7].

Definition 2. A **HTSPN** is a 3-tuple (R, C, A, F_c, F_a) such that:

- R is a TSPN, called the root of the hierarchical specification, extended to provide the link, atomic and composite place type modeling. Each composite or atomic type place in R is respectively associated, by the mapping functions F_c and F_a, with a STSPN defined in C or A. Link and composite places in R are graphically represented as bold circles and dotted circles, respectively.
- $C = (C_1,...,C_n)$ is a tuple of STSPNs, with a cardinal equal to the cardinal of the set of composite type places over R and C. The mapping function F_c allows the association between a composite place defined in R or C with a STSPN defined in C. Like in the Dexter model, the only constraint we put on this hierarchy is to be a directed acyclic graph, which means that a subnet must not contains a composite place directly or indirectly related to this same subnet.
- $A = (A_1,...,A_k)$ is a tuple of STSPNs. Each STSPN of A is related to atomic places within R and C by the function F_a. These STSPN are dedicated to the modeling of intra-media synchronization constraints. Therefore, these subnets are usually simpler (in terms of synchronization schemes) and have a finer synchronization granularity than those contained in C. In order to avoid too large graphical representations, continuous synchronization schemes can be expressed, with the help of a synthetic Petri net notation, by specifying the basic synchronization pattern and the number of iterations.

Firing Rules

The TSPN firing rules are extended for HTSPN by considering that the different nets in an HTSPN, denoted as H, progress like a single net. In particular, the nets in H share the same global clock for simulation purposes. The firing of a transition t is submitted to the usual TSPN firing conditions with the following extensions:

- withdraw all the tokens from the subnets related to composite or atomic type places of the prefix[8] of t;
- mark the entry places of the subnets relating to composite or atomic type places of the suffix of t.

These extended firing rules allow an easy modeling of multimedia scenario interruptions and they bring a big improvement in terms of net complexity compared with a modeling based on escape arcs [Prabhakaran, 93]. Therefore, these extended firing rules make it possible, by allowing the removal of tokens in subnets, to easily model asynchronous events issued from links activation that are the basis of traversal semantics of hypermedia systems. Moreover, the notion of state in HTSPN directly derives from the notion of state in TSPN. As a consequence, the simulation and analysis techniques developed for TSPNs can be directly and easily extended for HTSPNs [Sénac, 96].

A three Layer Model

The HTSPN definition entails a three layers conceptual model for hypermedia systems (see figure 7).

---

[7] A general definition of the HTSPN model can be found in [Sénac, 95].

[8] By definition the prefix of t is $\{p \in P \mid \beta(p, t) \neq 0\}$, and the suffix of t is $\{p \in P \mid \alpha(p, t) \neq 0\}$.

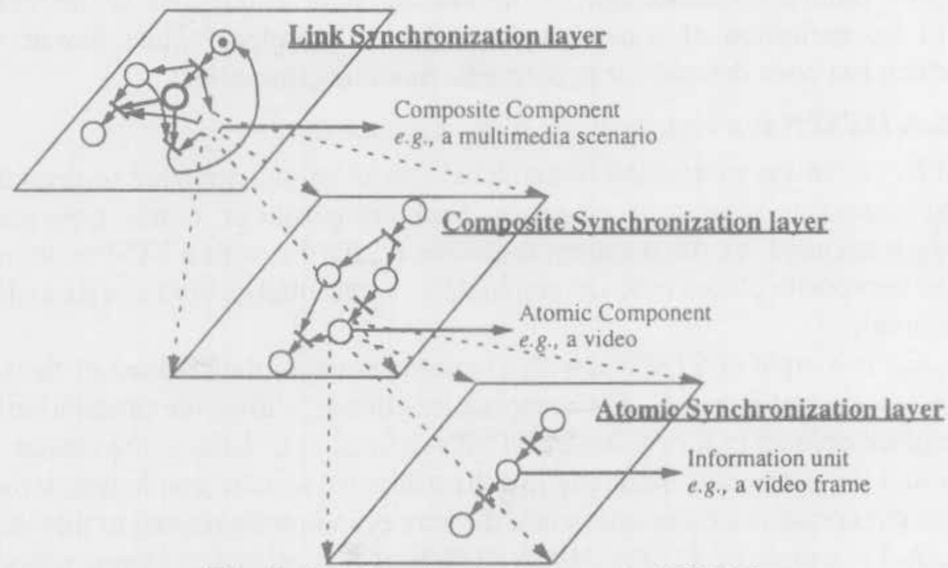


Fig. 7. Synchronization Layers in HTSPN

The upper layer, named the *Link Synchronization Layer*, is dedicated to the formal specification of semantic structures within hypermedia documents. At the link synchronization layer, places and their related outgoing arcs represent link, atomic and composite components. Note that, hypermedia scenarios described at the link layer are not necessarily structured and that nondeterminism and cycles may be used.

The *Composite Synchronization Layer* is dedicated to the formal specification of temporal and logical structures within hypermedia documents. At this layer, atomic and composite places allow the hierarchical structure of Dexter composite components to be modelled. At the Composite layer, multimedia scenarios are modelled as structured TSPN. STSPN are used because their modeling and expressive power are well suited for multimedia synchronization modeling. Furthermore, as previously indicated, STSPNs have «good» structural properties that allow one to check with a polynomial complexity the temporal correctness of the most time-critical parts of hypermedia systems, that is, the multimedia synchronization scenarios.

The *Atomic Synchronization Layer* aims to describe temporal and logical structures related to atomic components. This level is dedicated to the specification of intra-media synchronization.

3.1.5 Data and Presentation Characteristics Modeling

The HTSPN model enables a complete and accurate specification of the temporal, logical and semantic structures within hypermedia documents. However, this model was not designed for the complete formal specification of documents. For instance, the HTSPN model does not provides means to specify both the access information of media data and the spatial and audible presentation characteristics. In this section, we present an MHEG interpreted version of the HTSPN model (I-HTSPN) [Willrich, 96] providing means to completely specify hypermedia document in a final form. Moreover, the I-HTSPN model provides the means to specify all parameters necessary to the automatic generation of MHEG representations [ISO 13522].

The MHEG standard has greatly influenced the I-HTSPN definition. However, we don't directly add MHEG concepts to the HTSPN model. Instead, we define new abstractions based on the MHEG concepts in order to simplify the authoring process and to guarantee the consistency of the hypermedia document specification.

Data Specification

The pieces of information of a hypermedia document may consist of media data types that may be perceived by the user (*e.g.*, audio, video, text) and non media data types that cannot be presented to the user in an immediate way (*e.g.*, executable, HyperODA or HyTime documents).

In the I-HTSPN model, data information is modelled by a set of objects, called *DS* (Data Specification). These objects are instances of various *Data* classes. Each class allows the description of one media type (e.g., *Audio* and *Video* classes), non-media data type or script. For instance, a video stream can be specified by an object of *Video* class. This *Video* object specifies, for example, that this information, called *DataVideo*, is a compressed MPEG video clip, with an original size of 128 (H) x 256 (W) pixels and an original duration of 10 s, and it is stored in /home/dt/video11.mpg.

Presentation Specification

A hypermedia model must provide means for specifying the presentation characteristics of several pieces of information included in a document. Therefore, the author of a hypermedia document must specify the temporal, spatial and audible characteristics of the presentations that compose a document. Moreover, he/she may specify the output device where this presentation is to be rendered to the readers.

In the I-HTSPN model, presentation characteristics of data information are modelled by a set of objects, called *PS* (Presentation Specification). These objects are instances of various *Presentation* classes. Each class allows the presentation characteristics modeling of one media or non-media data type, script, or link, to be described (*e.g., PSAudio* and *PSVideo* classes).

A set of *Channel* objects, called *CS* (Channel Specification), specifies all channels used by a document. A *Channel* object specifies a logic space in which a presentation is positioned and rendered by the user. It describes the channel identifier and its requirements.

For instance, a presentation of object *DataVideo,* of the *Video* class, can be specified by an object of the *PSVideo* class (a *Presentation* class). This object specifies, for instance, one presentation of *DataVideo* to be rendered in the position 100 (H) x 100 (W).

The I-HTSPN Model

With the complete specification of hypermedia documents in mind, the HTSPN model has been interpreted by associating abstractions for specifying the used data information and theirs presentation characteristics. We associate presentation characteristics with each atomic and link type place. This association is realized by the F_{PS} mapping function. This function associates an atomic or link type place with an object defined in *PS* (*i.e.*, the set of *Presentation* objects). Moreover, we add to the HTSPN model two other mapping functions, called F_{DS} and F_{CS} to associate *Presentation* objects with *Data* and *Channel* objects, respectively.

Link type places within HTSPN provides means for representing user interactions. In the proposed framework, a link type place represents a button (if it is associated with only one output arc), or a non hierarchical menu (if it is associated with several output arcs).

The extended HTSPN model results from an interpretation of the HTSPN model. All characteristics of the HTSPN model are preserved, the logical and temporal synchronization schemes and firing rules are maintained as defined in [Sénac, 95].

In short, a I-HTSPN is formally defined as follows:

Definition 3. An **interpreted HTSPN** is a tuple I-HTSPN = (HTSPN, PS, CS, DS, F_{PS}, F_{CS}, F_{DS}) such that:

- HTSPN = (R, C, A, F_c, F_a) defines a HTSPN (see definition 2).
- PS, is a set of objects of the *Presentation* classes.

- CS, is a set of objects of the *Channel* classes.
- DS, is a set of objects of the *Data* classes.
- F_{PS}, is a function that associates a *Presentation* object with an atomic or link place type defined in R and C.
- F_{CS}, is a function that associates a *Channel* object with a *Presentation* object.
- F_{DS}, is a function that associates a *Data* object with a *Presentation* object.

Main Issues in Temporal Behaviour Modeling and Analysis

The inclusion of data and presentation characteristics modeling allows an improvement of the specification power of the behaviour of information items of a hypermedia document. Moreover, the presentation modeling provides means for new analysis types.

With the help of the 9 different synchronization semantics associated with transitions of the HTSPN model, the author of a hypermedia document can define a temporal behaviour of a particular presentation in the case of exception conditions that may be produced during a synchronization. Using the presentation characteristics modeling facilities, the author can describe actions that can be carried out when a presentation of media data has reached a synchronization point and waits for other objects to reach this point. Possible actions are: continuing presentation of the last position of the item (freezing of a video) or presentation skipping.

The I-HTSPN model provides means for applying detailed verification techniques:

- Using the concept of channel, our model provides support to automatically detect conflicts between shared resources. For instance, if two audio presentations (modelled by two atomic places) use the same audio channel, a warning is reported to the author of the document. The technique of resource conflict detection is based on a worst case analysis.
- Based on the data and presentation description and channels required, we can automatically generate a description of all necessary resources for the hypermedia document presentation. For instance, the size of data information and data types of document can be identified. This description can be used to facilitate the installation, operation and management of hypermedia documents.

3.2 Automatic Generation of MHEG representations

This section describes how an I-HTSPN specification can be translated into an MHEG representation [ISO 13522]. This representation provides means for reproducing, communicating, and storing hypermedia documents.

The HTSPN-MHEG translation is defined by pairing the HTSPN model, a graphical modeling of a hypermedia document, and the coding of multimedia and hypermedia information specified by the MHEG standard, an object-oriented representation. This translation is carried out by direct mapping between the elements of an I-HTSPN and MHEG objects.

In the HTSPN model, composite type places provide a hierarchical structuring mechanism based on the recursive composition, in STSPN subnets, of atomic and composite places. The choice adopted for the HTSPN-MHEG translation is that the hierarchical composition of the HTSPN model will be translated into the structure provided by an MHEG object of the Composite MHEG class. This class defines a structure that provides a consistent approach for representing temporal, logical and spatial relationships among a set of MHEG objects. Therefore, the HTSPN hierarchy can be easily mapped into a Composite MHEG object.

The logical, temporal and semantic structures of a hypermedia document are translated into a Composite MHEG object (composed by several MHEG objects) generated from the HTSPN specification. This Composite object doesn't contain the data information itself. This information (specified by *Data* objects of the I-HTSPN model) is represented by a disjointed set of

Content (similar to the definition of atomic component) and Script (a script container) MHEG objects. This characteristic allows the user to reuse data information at different specification levels and to reuse a document structure with several different data information and presentation specifications.

The description of necessary resources for the hypermedia document presentation will be represented by an MHEG object of the Descriptor MHEG class (defined for this purpose). Therefore, the complete MHEG translation of the I-HTSPN specification is composed of three parts (figure 8): the data representation, composed of a set of MHEG objects; representation of the logical, temporal and semantic structures, composed of a Composite object; and the hypermedia document description, composed by a Descriptor object. All MHEG objects generated by our HTSPN-MHEG translation can be regrouped in a Container MHEG object. It provides a container for regrouping several MHEG objects that represent a hypermedia document.

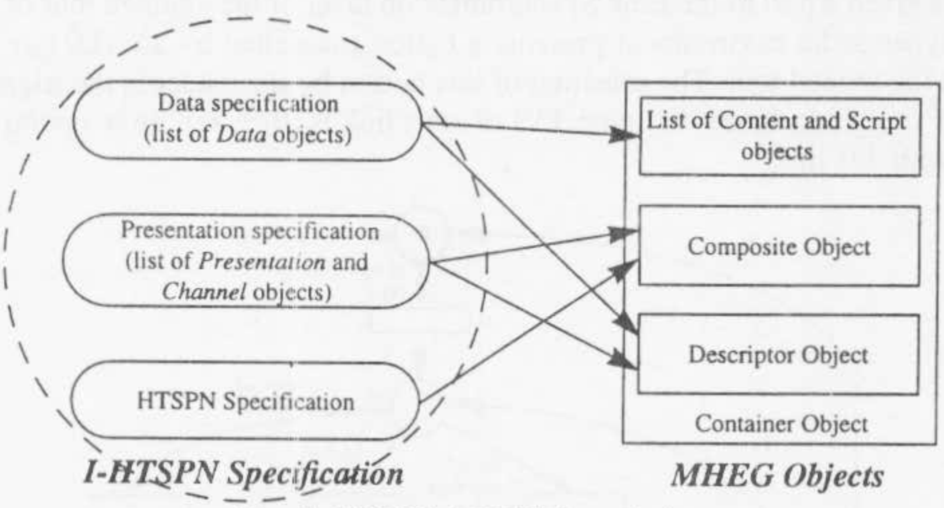


Fig. 8. HTSPN-MHEG Translation

4 The HTSPN-based Toolkit

A graphic environment based on the I-HTSPN model is being developed in order to assist the design of hypermedia documents. The final output of this toolkit is the automatic generation of the MHEG representation of hypermedia documents from a fully analysed formal specification. This graphic environment is an extension of the TSPN environment described in [Fabre, 95]. The HTSPN toolkit is being prototyped on Solaris using the C++ language. This toolkit consists of the following modules:

- The *Graphical Editor* guides the author through a graphical construction of a hypermedia document. The graphical editor makes it easy to specify the *Data* and *Presentation* objects, and the logical and temporal composition of the hypermedia document.
  Data objects can be specified using a *Data Specification* window. This window is a scrolling list of *Data* objects where an author uses commands to add, set-up, rearrange and remove *Data* objects. Like the data specification, channels of a document can be specified using the *Channel Specification* window. The logical and temporal composition of the hypermedia document is specified using a HTSPN graphical editor. By clicking a HTSPN place, a dialogue box is opened.
- The *Analyser* performs temporal analysis and resource conflict detection. If some time mismatch or resource conflicts are not acceptable by the author, the latter can edit the HTSPN again in order to modify the involved synchronization scheme.
- The *Simulator* provides several capabilities to simulate the dynamic and concurrent activities of the modelled hypermedia document.

- After the capture of a correct HTSPN specification of the hypermedia document, the *MHEG translator* can be used to generate its MHEG representation. The MHEG representation of a hypermedia document can be presented by an MHEG engine.

5 Example

In this section we use the I-HTSPN model for the design of a hypermedia document allowing a «guided tour of University». Firstly, the HTSPN specification of this document is presented and discussed. Then, the modelling of data and presentation characteristics is illustrated. Finally, the MHEG translation of the document is presented.

The HTSPN Specification

In figure 9 is given a part of the Link Synchronization layer of the «guided tour of University». First, this hypermedia environment presents a button (modelled by arc ($L0, t_0$)) allowing the user to start the guided tour. The selection of this button by the reader is the triggering logical condition of *start* link. Since the static TVI of *start* link is [0, *, ∞], no temporal constraint is associated with this link.

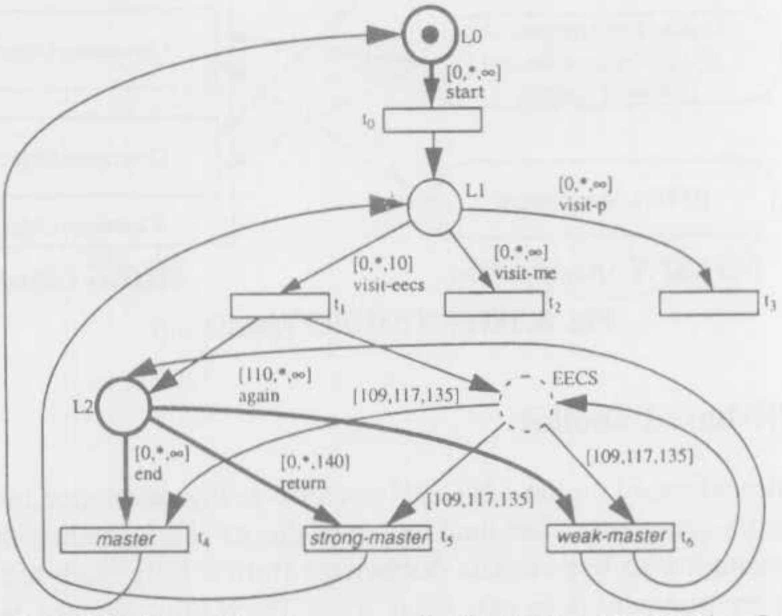


Fig. 9. HTSPN specification example: Link Synchronization Layer

Following the *start* link, the hypermedia environment presents a three choice menu, modelled as place L1. This menu allows, through an activation button, to trigger one among three links, namely *visit-eecs*, *visit-me* and *visit-p* (for a guided tour of the departments of electrical engineering and computer sciences, mechanical engineering, and physics, respectively). Note that according to the TVIs of these three links, the *visit-eecs* link is automatically triggered at relative time 10 if there isn't any user interaction before this time.

Following the explicit or implicit triggering of *visit-eecs* link, the multimedia scenario modelled by *EECS* place, is presented. The STSPN associated with this component type place is given in figure 10. Three links can be implicitly or explicitly triggered during the EECS department presentation:

- the «*end*» *master* type link allows the reader to stop the EECS presentation and going back to the beginning of the guided tour.
- the «*return*» *strong-master* type link allows the reader to stop the presentation and go back to the previous menu (L1). According to the *strong-master* semantics, if there isn't any user

interaction, this link is automatically triggered as soon as the EECS presentation reaches the maximum bound of its TVI (135 time units).

- the «*again*» *weak-master* type link allows the user to replay the EECS presentation. According to the *weak-master* semantics, this link can be triggered during [110, ∞] ([110, max(135, ∞)]). Therefore, the user must wait 110 time units after the start of the multimedia presentation before replaying it.

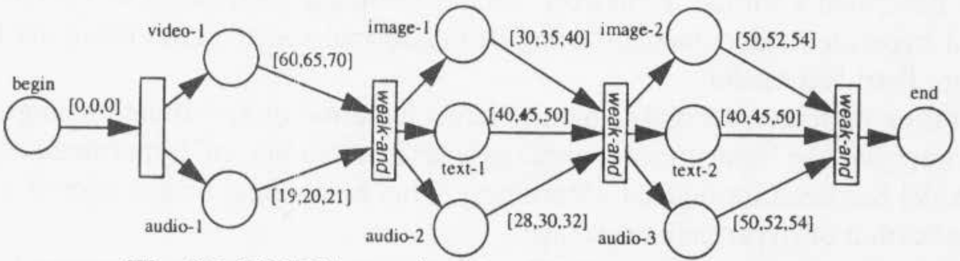


Fig. 10. STSPN associated to EECS composite place

The I-HTSPN Specification

In order to describe the lay-out of the «guided tour of University» document, the author must use the I-HTSPN model to describe the used data and the characteristics of the presentations modelled by atomic and link type places of the HTSPN specification.

For the specification part presented in figures 9 and 10, the author must specify the access and manipulation information of the eight data information used by the composition of the multimedia scenario specified by figure 10. For instance, object *DataVideo-1* (of the *Video* class of the I-HTSPN model) specifies a compressed MPEG video clip stored in /home/dt/video1.mpg, which has an original size of 128 (H)x 256 (W) pixels and an original duration of 10 s. Using the HTSPN-based toolkit, this specification is supported by the *Data Specification* window.

In addition to the data specification, the author must specify all presentations characteristics of the components of the document «guided tour of University». Therefore, for each atomic and link type place of the HTSPN specification, an object of a *Presentation* class must be associated. This object describes the presentation characteristics (including the data information specification) of the presentation modelled by the associated place. For instance, object *Video-1* (of the *PSVideo* class) can be associated with place *video-1* (in figure 10) to specify that its place represents a presentation of *DataVideo-1* in position 100(H) x 100(W). Using the HTSPN-based toolkit, by clicking in place *video-1* for instance, a dialogue box is opened. Using this dialogue box, the author can specify the presentation characteristics.

The HTSPN-MHEG Translation

After the specification and analysis of the «guided tour of University» document, the author can use the HTSPN-based toolkit to automatically generate an MHEG representation of this document. For the specification part presented in figure 9 and 10, the MHEG objects generated by this translation are the following:

- One composite MHEG object (including several MHEG objects) that represents the root of the HTSPN specification. In particular, this MHEG object contains another Composite MHEG object that represents the subnet related to composite place EECS. These Composite MHEG objects represent the logical, temporal, semantic structure of the document, and the presentation characteristics of each component;
- Eight Content MHEG objects generated from the data specification.
- One Descriptor MHEG object that specifies the necessary resources for the hypermedia document presentation.

These MHEG objects can be regrouped in a Container MHEG object. It provides a container for regrouping several MHEG objects that represent the document «guided tour of University». This MHEG representation can the be interpreted (presented) by an MHEG engine.

6 Conclusion

This paper described a formal framework for the automatic generation of ISO MHEG representation of hypermedia documents starting from a formal specification using the Hierarchical Time Stream Petri Net model.

Incorporating high level formal semantics into a hypermedia specification language can significantly improve the robustness, correctness and reliability of hypermedia systems; the HTSPN model has been introduced accordingly. This new model brings several contributions to the specification of hypermedia systems:

- The model allows an easy, formal and unified way for specifying temporal, logical and semantic structures of hypermedia systems.
- The model brings enlightenments on hypermedia synchronization by allowing synchronization schemes combining logical and temporal synchronization to be specified. Moreover, HTSPN takes into account temporal non-determinism and synchronization nondeterminism.
- Asynchronous events able to interrupt at any time a multimedia scenario can be easily modelled and simulated.
- The model is a powerful method for the specification, simulation, prototyping, verification and validation of hypermedia systems. It leads to an incremental and modular specification methodology that simplifies considerably the modeling of hypermedia systems [Sénac, 96]. Moreover, a HTSPN software engineering toolkit has been designed that assists the user in its modeling, simulation, verification and analysis tasks.

Basic HTSPNs [Sénac, 95] only address logical and temporal synchronization issues at the specification level. The I-HTSPN model discussed in the paper specializes a semantics with places in order to specify information items and their presentation characteristics. The use of the I-HTSPN model allows an easy construction of accurate and unambiguous hypermedia documents, and permits powerful verification, simulation and analysis methods to be applied. Moreover, this model provides means for an automatic generation of MHEG representation from fully analysed specification of hypermedia documents.

The proposed framework seems to be a strong underlying basis for producing the specification of hypermedia document in its final form and for translating this specification into an MHEG representation.

Ongoing work includes the development of an I-HTSPN integrated environment that offers user-friendly interface to specify, verify, validate and simulate synchronization constraints inside hypermedia documents. This environment will make it possible to generate MHEG documents structures from I-HTSPN specifications. These documents will be designed to be directly interpreted by an MHEG engine.

7 References

[Allen, 83] J.F. Allen. Maintaining Knowledge about Temporal Intervals. Communication of ACM 26(11):832-843, 1983.

[Appelt, 93] W. Appelt. HyperODA - Extensions for Temporal Relationships. Proposed Draft Amendment (version 2) ISO/IEC JTC 1/SC 18/WG 3/N 2516, 1993.

[Buchanan, 93] M.C. Buchanan, P.T. Zellweger. Automatically Generating Consistent Schedules for Multimedia Documents. Multimedia Systems Journal, 1993.

[Bulterman, 93] D.C.A. Bulterman. Specification and Support of Adaptable Networked Multimedia. ACM Multimedia Systems, 1(2):68-76, 1993.

[Coulson, 92] G. Coulson, G.S. Blair, N. Davies, N. Willians. Extensions to ANSA for Multimedia Computing. Computer Networks and ISDN Systems, 25, 1992.

[Conklin, 87] J. Conklin. Hypertext: An Introduction and Survey. Computer 20(9): 17-41, 1987.

[Diaz, 94] M. Diaz, P. Sénac. Time Stream Petri Nets, a Model for Timed Multimedia Information. In proc. of the 15th Int. Conf. on Application and Theory of Petri Nets, Zaragoza, Springer-Verlag, R. Valette (Ed.), pp. 219-238, 1994.

[Fabre, 95] F. Fabre, P. Sénac, M. Diaz. A Toolkit for the Modeling of Multimedia Synchronization Scenarios. In proc. of the 3th Int. Conf. on Multi-Media Modeling, pp. 187-201, 1995.

[Hardman, 94] L. Hardman, D.C.A. Bulterman, G. Van Rossum. The Amsterdam Hypermedia Model: Adding time to the Dexter Model. Communication of the ACM 37(2):50-62, 1994.

[Halasz, 94] F. Halasz, M. Schwartz. The Dexter Hypertext Reference Model. Communication of the ACM 37(2):30-39, 1994.

[ISO 13522] ISO 13522-1. Information Technology - Coding of Multimedia and Hypermedia Information - Part 1: MHEG Object Representation, Base Notation (ASN.1), 1994.

[Liestol, 94] G. Liestol. Aesthetic and Rethorical Aspects of Linking Video in Hypermedia. In the proc. of the ACM European Conference on Hypermedia Technology, ACM Press, Edingurgh, September 1994.

[Little, 90] T. Little, A. Ghafoor. Synchronization and Storage Models for Multimedia Objects. IEEE J. on Selected Areas in Communications 8(3): 413-427, 1990.

[Newcomb, 94] S. Newcomb, N. Kipp, V. Newcomb. The HyTime Hypermedia/Time-based Document Structuring Language. Comm. of the ACM 34(11):67-83, 1991.

[Prabhakaran, 93] B. Prabhakaran, S.V. Raghavan, Synchronization Models for Multimedia Presentation with User Participation. In proc. ACM Multimedia 93, pp. 157-165, 1993.

[Schloss, 94] G.A. Schloss, M.J. Wynblatt. Building Temporal Structures in a Layered Multimedia Data Model. In proc. of the ACM Multimedia 94, pp. 271-278, 1994.

[Sénac, 94] P. Sénac, M. Diaz, P. de Saqui Sannes. Toward a Formal Specification of Multimedia Synchronization Scenarios. Annals of Telecommunications 49 (5-6): 297-314, 1994.

[Sénac, 95] P. Sénac, P. de Saqui-Sannes, R. Willrich. Hierarchical Time Stream Petri Net: a Model for Hypermedia Systems. In Application and Theory of Petri Nets 1995, Lecture Notes in Computer Science no. 935, G. De Michelis and M. Diaz (Eds.), Springer, pp. 451-470.

[Sénac, 96] P. Sénac, M. Diaz, A. Léger, P. de Saqui Sannes. Modeling Logical and Temporal Synchronization in Hypermedia Systems. IEEE Journal on Selected Areas in Communications 14(1):84-103, 1996.

[Stotts, 90] P.D. Stotts, R. Furuta. Temporal Hyperprogramming. Journal of Visual Languages and Computing 1:237-253, 1990.

[Willrich, 96] R. Willrich, P. Sénac, M. Diaz, P. de Saqui-Sannes. A Formal Framework for the Specification, Analysis and Generation of Standardized Hypermedia Documents. To appear in the 3rd Int. Conf. on Multimedia Computing and Systems. Japan June 17-21,1996.