Synchronization Models for Interactive Multimedia Presentations Using Timed Petri Nets

MuDer Jeng[*] & YuanLin Wen

Department of Electrical Engineering, National Taiwan Ocean University
Keelung 202, Taiwan, ROC

ABSTRACT

In an interactive multimedia presentation system the user should be allowed to manipulate the presentation sequence such as to repeat a sub-sequence, choose a sequence branch, change the playing speed, and reverse the playing sequence through the keyboard, mouse or program. The synchronization scheme should also allow the user to modify the sequence of the presentation dynamically. This paper presents a formal model that describes synchronization behaviors in multimedia systems, called *Object Slice Petri Nets* (OSPN). Since an OSPN representing a real-world multimedia scenario, may develop a very complex behavior whose qualitative properties are difficult to verify with reasonably computational cost, a systematic synthesis method based on an extension of the *knitting technique* is used for macro level scenario synthesis. The goal is to synthesize well-behaved nets. In this paper, deadlock-free Macro-level OSPNs classified into *static* and *dynamic scenario nets* are proposed and analyzed.

***Index Terms*: Multimedia, Multimedia Synchronization, Timed Petri Nets, Synthesis Technique**

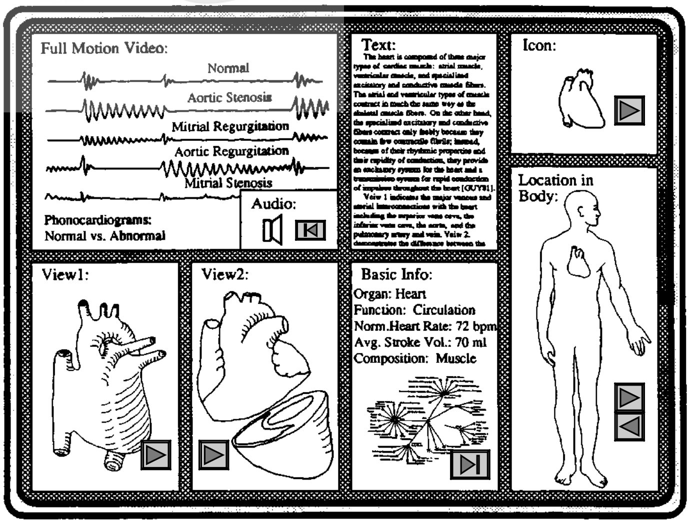
1.Introduction

Currently, multimedia technology is widely used for business, education, and entertainment. The integration of various media increases the power for representing information, and provides a more intuitive and friendly interface for the user. Multimedia refers to the integrated generation, representation, processing, storage, and dissemination of independent machine processable information expressed in multiple time dependent and time independent media such as text, images graphics audio, and video. Multimedia presentation consists of a set of temporally related action, which have to be executed in a special intended sequence defined by the user. This scheduling is generally called synchronization of actions. The media characteristics and the temporal interdependencies must be clearly established to ensure the proper scheduling of the synchronized presentation. A pre-orchestrated multimedia presentation consists of a set of temporally related actions that are executed in a predefined sequence.

Fig. 1.1 is an example of interactive multimedia presentation system. It is a adapted example from [1] by adding some operation buttons, by which the user can be allowed to dynamically manipulate the presentation sequence through the mouse.

---

[*] MuDer Jeng and YuanLin.Wen. are with the Department of Electrical Engineering, National Taiwan Ocean University, Keelung 202, Taiwan, R.O.C.(e-mail: b0162@ind.ntou.edu.tw).



Organ: Heart
Function: Circulation
Norm.Heart Rate: 72 bpm
Avg. Stroke Vol.: 70 ml
Composition: Muscle

Fig. 1.1 An example of interactive multimedia presentation system application in anatomy and physiology instructor.

Our work in this paper differs from earlier works in several major aspects [1,3,14]. We provide a formal specification for synchronization and for the retrieval and presentation for multimedia elements. We propose an explicit for synchronization, which can be applied to various applications without assuming a particular application environment.

In this paper, we focus on synchronization problems of the temporal relationships among objects, we propose multimedia synchronization models based on timed Petri nets, called *Object Slice Petri Nets* (OSPN) to model synchronization properties in the conceptual layer. An OSPN that is an extension of an *Object Composed Petri Nets* (OCPN) [1] to consider user interactions for handing the presentation sequence. It is assumed that the underlying hardware and software is capable of proper execution and presentation of the composed multimedia objects to meet the temporal constraints as defined by the user. Our proposed methodology concentrates on the synchronization at two levels, called the *macro* and *micro* levels. The units of synchronization at the *macro* and *micro* levels are objects and object slices called *logical data units* (LDU) respectively. At the micro level, finer-grained synchronization is achieved by dividing each object into a sequence of LDUs, each with its own fixed synchronization interval. For a video, the range of an interval can be from the interval of an object to that of a frame. LDUs are defined to model the synchronization of continuous media data where playing of objects at a media level finer than the level of objects is required.

To compose an interactive presentation, we define several special sub-net modules, called *Skip, Reverse, Stop-&-Resume, Choice, Scaling-Speed, Wait-for-Choice* and *Wait-Some-Time*. These modules specify all user interaction scenarios at the micro level, or the LDU level. For the part of the presentation where no user interaction is involved, we construct the scenario at the macro level, or the object level. At this level, every place represents an object.

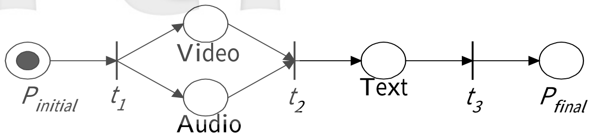


Fig. 2.1 An OCPN example.

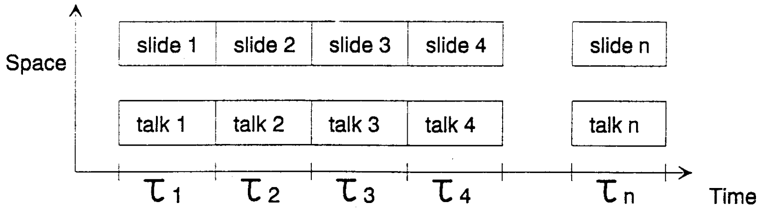


Fig. 2.2 Timeline of Slide presentation.

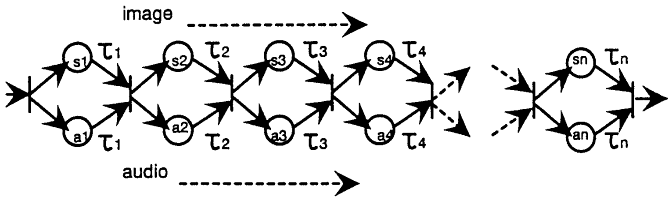


Fig. 2.3 Petri net of Slide presentation.

After a net model is constructed, it is necessary to analyze qualitative properties of the model to discover any anomalous behaviors. Important properties of Petri nets include *safeness*, *liveness*, and *reversibility*. They are related to freedom of status overflow, deadlock-freeness, and re-initialization, respectively, in multimedia.

Considering the contributions to the multimedia synchronization problem as indicated above, we notice that there have been similar approaches in prior work. However, all these approaches cannot deal with all possible user interaction scenarios as in our approach. In addition,their approaches require the modification of the firing rule. One of the advantages of our approach is that because an OSPN is based on a timed Petri net, we do not change the firing rule. As a result, existing analysis methods of the Petri net theory can be used to analyze properties of an OSPN.

The paper is organized as follows: In Section 2, we notice some similar approaches in prior research [3,14,21] and several definitions for multimedia system. Section 3, we present a formal model, *Object Slice Petri Nets* (OSPN) for describe synchronization behaviors and different phenomena in multimedia systems. The rules for constructing OSPNs are presented and deadlock-free *Dynamic Scenario nets* are proposed in Section 4. In Section 5, we analyze the system model for three properties in multimedia, i.e. safeness, liveness, and reversibility. In Section 6, we show an application of OSPNs to a *computer assisted instruction* (CAI) system. Finally, conclusions and future research are given in section 7.

2. Related Survey

Petri nets as a modeling tool that has the ability to describe physical time relationships for multimedia presentations. For the purpose of modeling timed-driven systems, the notion of time was introduced in Petri nets. They are known as *Timed Petri Nets* (TPN) [2], which allow all kinds of synchronization specifications. In a marked TPN, tokens are assigned to places. A transition is enabled for execution if each of its input places contain at least one token. The firing of a transition causes a token to be held in a locked state for a specified time interval in the output place of the transition. In this paper, it should be noted that all of the time intervals are assigned to places.

In our methodology, *Object Slice Petri Nets* (***OSPNs***) are extended from OCPNs to facilitate the modeling of multimedia synchronization characteristics with dynamic user participation.

An OCPN is also based on a TPN and can be used for modeling synchronization requirements for multimedia objects [1]. An OCPN augments a Petri net with values of time, as duration, and resources used in the places of the net. The execution of an OCPN is similar to that of TPNs where the transition firing is assumed to occur instantaneously, and the places are assumed to have states. The firing rule for OCPNs is the same as for TPNs, as discussed earlier.

For illustrating of the OCPNs, consider the following examples:

Examples 2.1: An example of an OCPN specification of playing audio and video objects followed by text in Fig 2.1.

Examples 2.2: Assume that there are n slides to present. In a time/resource line, we can represent the presentation as two streams of information, which occur concurrently, as shown in Fig. 2.2. Using the Petri net representation, these activities are shown in Fig. 2.3.

Using the OCPN model, it is impossible to describe modifications of the presentation sequence by the user. For instance, the user may wish to stop a presentation, reverse it, or skip a few frames. These cannot be described in the existing OCPN architecture.

An *extended OCPN* (XOCPN) [14] model takes into account the demands of isochronous data, requiring a rate-controlled transmission. It is used under a network environment where several configurations are possible for a *distributed multimedia information system* (DMIS). Each XOCPN has functions for supporting synchronization among multiple streams. A transmitter XOCPN schedules transmission of objects at the source site using information about network delays and channel capacity. On the other hand, at the user site the receiver XOCPN can incorporate mechanisms to compensate for anomalies such as lost or delayed packets.

Dynamic Timed Petri Nets (DTPN) [3] allow user-defined “interrupts” to preempt the net execution sequence and modify the time duration associated with the pre-empted net process. This structure can be adopted to facilitate the modeling of multimedia synchronization characteristics with dynamic user participation, However, its firing rules must be modified. As a result, existing analysis methods of the Petri net theory can not be used to analyze properties of DTPNs.

In the Following several definitions for multimedia Presentation system are briefly reviews.

2.1 Multimedia Composition

Spatial composition is concerned with the combination of objects in space. For pictorial representation of still images and graphics, composition operations include scaling, cropping, overlay, etc. For audio data, composition consists of mixing signals. Other "spatial" audio operations include gain and tone adjustments, which are useful in video conferencing applications to prioritize a speaker's voice amongmany. Tone differences can signify "distance" in participants via signal-distortion techniques. A typical spatial composition for text and image is shown in Fig. 2.4.

Temporal composition requires evaluating the relationships among media and scheduling their retrieval and presentation accordingly. The relationships can occur naturally, as for live audio and video, and they also can be synthetic, consisting of arbitrary temporal constraints on any multimedia data type.

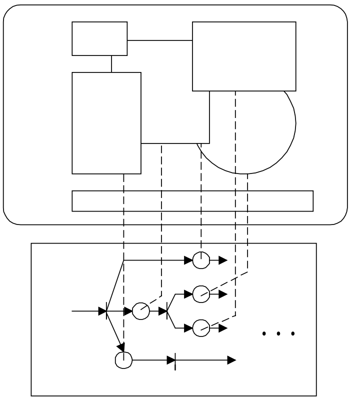


Fig. 2.4 Spatial composition in OSPN.

Most multimedia applications need combinations of the spatial and temporal composition. Consider a multimedia application of electronic magazine. A page in the electronic magazine may contain various media (video, images, text, audio, etc). A reader may browse through "pages" to read articles and view pictures with audio/video presentations.

2.2 Classification of Synchronization Types

We distinguish between time relations within the units of one time-dependent media object itself and time relations between media objects. This separation helps to clarify the mechanisms supporting these types of relations, which are often very different:

1.Intra-object synchronization: Intra-object synchronization is concerned with delivering each object in time to meet the respective playout deadline, i.e. is to keep continuous playback of a media stream at a destination, which refers to the time relation between various presentation units of one time-dependent media object. An example is the time relation between the single frames of a video sequence. For a video with a rate of 25 frames per second, each of the frames must be displayed for 40 ms. Fig. 2.5 shows this for a video sequence presenting a jumping ball.

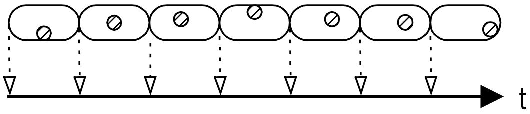


Fig. 2.5 Intra-object synchronization between frames of a video sequence showing a jumping balls.

2.Inter-object synchronization: Inter-object synchronization is required when temporal relationship between or more related objects or streams should be maintained. The temporal relationships that exist among different media components of a multimedia object at the time of recording must be preserved at the time of playback, which refers to the synchronization between media objects. Fig. 2.6 shows an example of the time relations of a multimedia synchronization that starts with an audio/video sequence, followed by several pictures and an animation that is commented by an audio sequence.

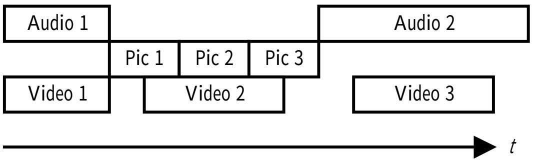


Fig. 2.6 Inter-object synchronization example.

Definition 2.1 (*Logical Data Units*)**:** Time-dependent media objects usually consist of a sequence of information units. Such information units are called *Logical Data Units (LDU's)*[2].

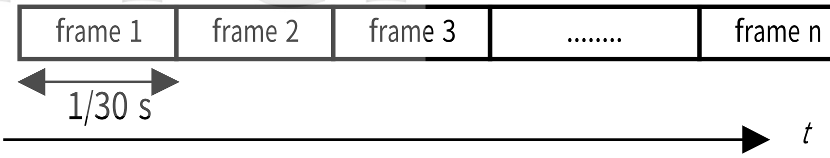


Fig. 2.7 Example of video LDU's.

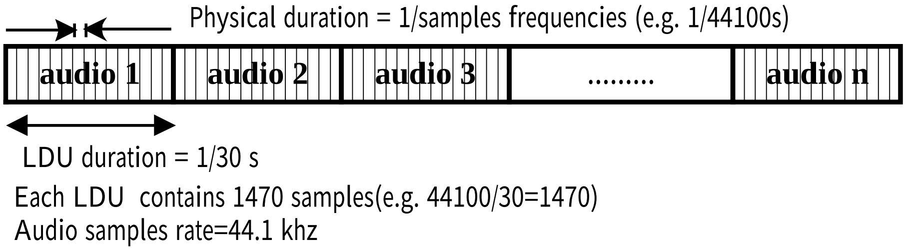


Fig. 2.8 Example of audio LDU's.

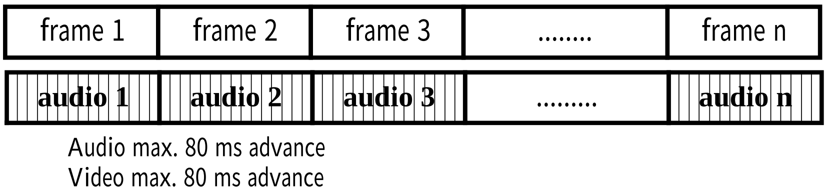


Fig. 2.9 LDU view of lip synchronization.

For digital video, often the frames are selected as LDU's. For example, for a video with 30 pictures per second, each LDU is a closed LDU with a duration of 1/30 s (Fig. 2.7).

In the case of the basic physical units being too small to handle, often LDU's are selected to divide the samples into units of afixed duration. A typical example is an audio stream where the physical unit duration is very small. Therefore, LDU's are formed to comprise 1470 samples. In the example shown in Fig. 2.8, one sample is coded with one byte, and hence, each block contains 1470 Bytes.

2.3 Lip Synchronization

Lip synchronization demands tight coupling of audio and video streams, such type is a inter-media synchronization. Synchronization can be specified by defining a maximal skew between the two media streams (Fig. 2.9).

3. Modeling

3.1 Object Slice Petri Net

An *Object Slice Petri Net* (OSPN) is an extension of an OCPN and can be used for modeling the synchronization pre-orchestrated multimedia presentation. Similar to OCPNs, OSPN augment Petri net models with values of time, as duration, and resources used.

For the presentation of continuous media data, we specify the transmission and playout of the objects at a finer grain media rather than at the level of an object, to facilitate a constant-rate presentation and to maintain temporal relationships among various media.

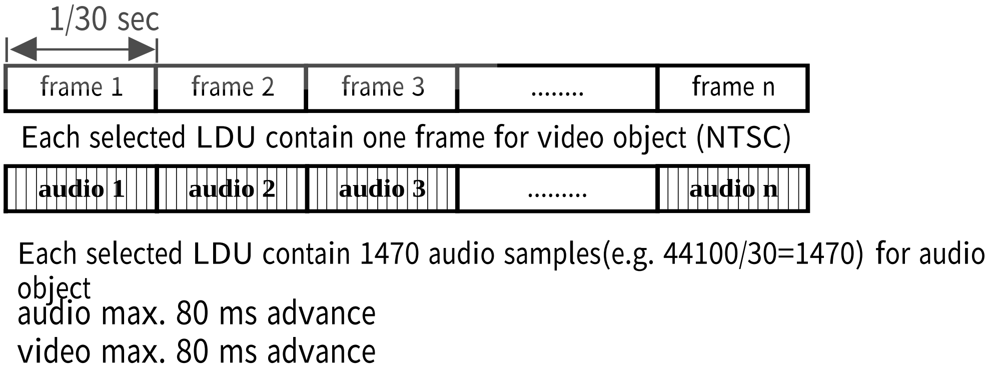


Fig. 3.1 A selected isochronous data format

The finer grain synchronization is achieved by dividing each object into a sequence of object slices, each with its own synchronization interval, which we call *Logical Data Units* (LDU). In other words, an LDU is a division of a complete multimedia object associated with a temporal interval that describes its duration in an OCPN [3].

In our OSPN models, we assume that all of time-dependent media object can be composed by discrete information slices. For example, a video stream can be composed by 30 frames per sec (NTSC), or 25 frames per sec (PAL) and an audio stream can be composed by 44.1Khz samples, i.e., 44100 samples per sec.

For the purpose of applying our models in interactive multimedia presentation environment, we divide the original media objects into media slices, i.e., isochronous data format as Logical Data Units (LDU) [2]. Each LDU holds a part of theoriginal object. LDU is marked with a synchronization interval number as a part of header information. Fig. 3.1 is a selected LDU format. Its duration is 1/30 s. The duration of an LDU is dictated by the request of system response time. For instance, when our interactive choice is executed, the presentation sequence will be changed to our choice of direction in a limited time. In Fig. 3.1, one selected LDU contains one frame of video object (NTSC = 30 frames/sec) and contains 1470 samples (samples rate=44.1Khz,i.e., 44100/30=1470) of audio objects. This finer grained synchronization demands tight coupling of audio and video streams. Synchronization can be specified by defining a maximal skew [2] (80 ms advance) between the two media streams.

3.2 Basic Assumption of modeling

In order to simplify the multimedia system, some additional issues of multimedia presentation are not considered in our system architecture design. Thus, before the model construction procedure is described, our assumptions are given below:

1. Our presentation system does not provide the spatial composition of multimedia objects. We only considered the temporal composition among media objects at the concept layer.
2. All of time-dependent media objects can be composed by discrete information slices. Video objects can be composed by frames, and audio objects can be composed by audio samples.

3. It is assumed that the underlying hardware and software is capable of proper execution and presentation of the composed multimedia objects to meet the temporal constraint. We concentrate on the high level synchronization to obtain the sequence of presentation as defined by the user.

3.3 Reduction Technique

To facilitate the analysis of a large system, we often reduce the system model to a simpler one, while preserving important system properties. Conversely, there are techniques to transform an abstracted model into a more refined model in a hierarchical manner for synthesis. There exist many transformation techniques for Petri nets that preserve liveness, safeness, and boundedness.

We use two rules, Fusion of Parallel Places (FPP) and Fusion of Series Places (FSP)[7] in our proposed method, as shown in Figs. 4.2 and 4.3.

Definition 3.1 (*Complex Logical Data Units*): A *Complex Logical Data Units* (CLDU) denoted as p_m is the fusion of a set of *Logical Data Units* (LDUs) as in Fig. 3.2. Fig. 3.2b is a CLDU replacement of Fig. 3.2a by the FPP Reduction Rule.

Definition 3.2 (*Scenario Place*): A *Scenario Place* denoted as P_{SP} is a set of serial CLDUs i.e., each *Scenario Place* contain a set of serial-parallel CLDUs. Fig. 3.3b where the darker place denote a *Scenario Place* is a replacement of Fig. 3.3a by the FSP Reduction Rule.

3.4 Definitions of Object Slice Petri Net

Definition 3.3 (*Object Slice Petri Net*): An *Object Slice Petri Net* (OSPN) is defined as $C_{OSPN}=\{ T_i \cup T_u, P_M \cup P_m \cup P_{sp}, A, D, U, M\}$

$T_i=\{t_1, t_2, \ldots t_n\}$ is a set of *immediate transitions,* drawn as thin bars.

$T_u=\{t_{u1}, t_{u2}, \ldots t_{un}\}$ is a set of *user interactive transitions,* drawn as thick bars.

$P_M=\{P_{M1}, P_{M2}, \ldots P_{Mn}\}$ is a set of *macro places*, where each place represents a media object, drawn as circle.

$P_m=\{P_{m1}, P_{m2}, \ldots P_{mn}\}$ is a set of *micro places*, where each place represents a *Complex Logical Data Units* (CLDU). drawn as a double circle.

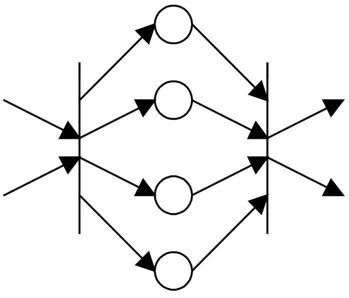
$P_{sp}=\{P_{sp1}, P_{sp2}, \ldots P_{spn}\}$ is a set of *Scenario Place*, where each place represents a set of *Complex Logical Data Units* (CLDU), drawn as a darker circle.

$A = \{T \times P\} \cup \{P \times T\} \rightarrow I$ is a set of directed arcs.

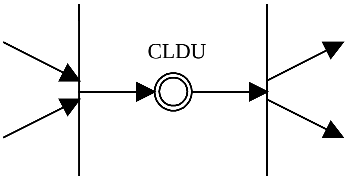
D: $\{P_M \cup P_m\} \rightarrow R$, where R is the set of real numbers, which represents micro level and macro level executed durations.

U: represents a set of media-resource types.

M: $P \rightarrow I$ represents the marking in each place.

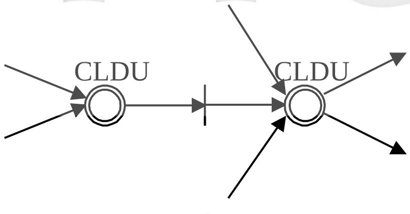


(a)Parallel LDU

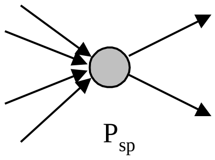


(b) CLDU Reduced form (a)

Fig. 3.2 Fusion of Parallel Places (FPP) Reduction Rule.



(a)Serial CLDUs



(b) Scenario place P_{sp} reduced form (a)

Fig. 3.3 Fusion of Series Places (FSP) Reduction Rule.

3.5 OSPN generation

To convert from media objects to object slices, for example, we consider an OCPN in Fig. 3.4. It is a part of a reduced Scenario (*Scenario place* A) in Fig. 3.19 shown later. The scenario contains 11 time slices, so we can divide this scenario into 11 LDUs. First, we represent it in a time-line model in Fig. 3.5. Second, we divide it into LDUs (τ = 1/30 sec). We obtain an OSPN in Fig3.6. Third, we use the FPP Reduction Rule to reduce each row of Fig. 3.6 to a *Complex LDU* (CLDU). The result is shown in Fig. 3.7.

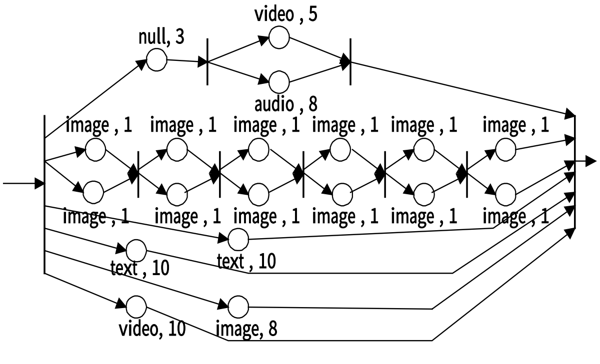


Fig. 3.4 Scenario modeled by an OCPN specification.

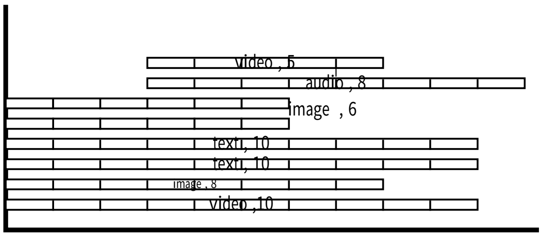


Fig. 3.5 Scenario modeled by time-axes specification.

After the basic OSPN generation, all concurrent media objects (Fig3.4) will be reduced to a sequence of CLDUs (Fig3.7).

3.6 Typical Sub-nets

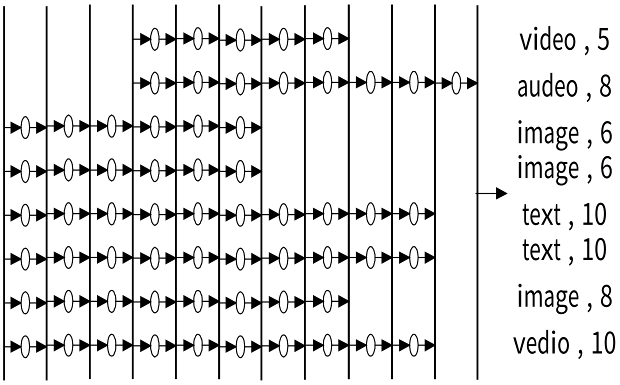


Fig. 3.6 Scenario modeled by a complete OSPN specification.

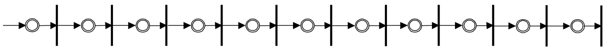


Fig. 3.7 Basic OSPN scenario generation.

After the basic generation of an OSPN, we must define some fundamental structures of OSPN for describing features related to user interactive choices. Fig. 3.8 is a typical sub-net structure of choice. When $\cdot p_m$ fires, one token goes to p_m, i.e., playing the media slice object. In the meantime, if each of the user interactive transitions t_u is enabled, because the priority of t_u is higher than that of t_i, t_u will fire after the execution time of p_m. Thus, the execution sequence is changed to that chosen by the user.

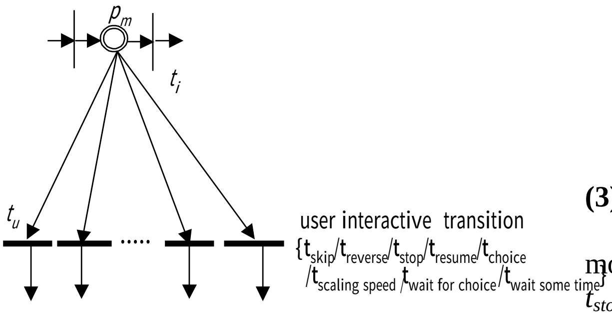


Fig. 3.8 OSPN typical choice sub-net.

In Fig. 3.8, two levels of user interactive choices are defined below.

1.Micro-level sub-nets: These are for modeling user interaction directions related to intra-object *micro places* p_m.

(1) Skip Sub-net:

The skip operation can be modeled as a free choice net in Fig. 3.9, where t_{skip} is a user interactive transition with a higher priority. When p_m has a token and t_{skip} is enabled by the user, t_{skip} is fired. The original execution sequence is skipped by one slice.

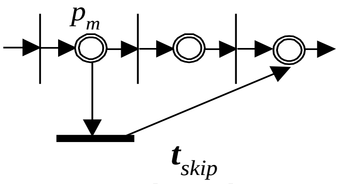


Fig. 3.9 Skip Sub-net.

(2)Reverse Sub-net

The reverse operation can be modeled as a free choice net in Fig. 3.10, where $t_{reverse}$ is a user interactive transition with a higher priority. When p_m has a token and $t_{reverse}$ is enabled by the user, $t_{reverse}$ is fired. The original execution sequence is reversed by one slice.

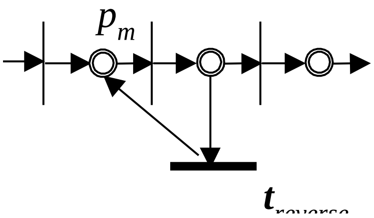


Fig. 3.10 Reverse Sub-net

(3)Stop-&-Resume Sub-net

The stop & resume operation can be modeled as a free choice net in Fig. 3.11. When t_{stop} is enabled, one token stay in the *Hold place* P_h until t_{resume} is enabled by the user. The *Hold place* contains only of video LDUs, not audio LDUs.

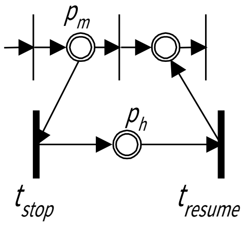


Fig. 3.11 Stop & Resume Sub-net

(4)Scaling-Speed-Fast Sub-net

In this sub-net, we divide an originally processed LDU into smaller slices, To scale the speed Ntimes faster, we divide it into N slices. This operation can be accomplished by skipping N sub-nets. As shown in Fig. 3.12, $t_{scaling\text{-}fast}$ has a higher priority. Note that scaling-speed-fast operations only apply to video objects.

(5)Scaling-Speed-Slow Sub-net

The scaling-speed-slow operation is done by playing the same LDU N times. As shown in Fig. 3.13, when $t_{scaling\text{-}slow}$ is fired, a token in p_m will stay N times longer because $t_{scaling\text{-}slow}$ has a higher priority. Note that scaling-speed-fast operations only apply to video objects.

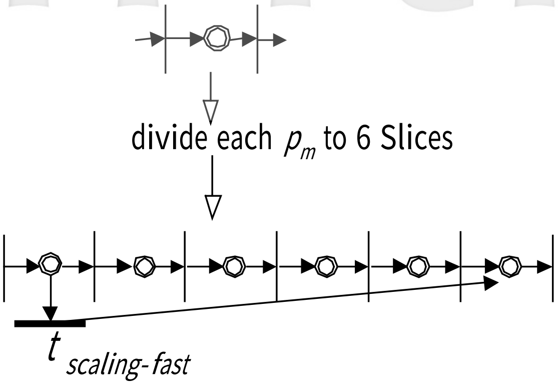


Fig. 3.12 scaling- six-times-fast sub-net

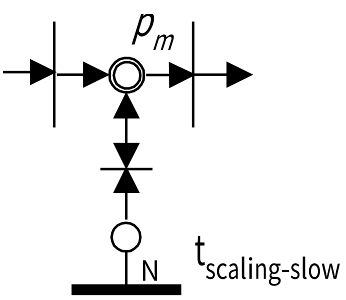


Fig. 3.13 scaling-N-speed-slow sub-net

2. Macro-level sub-nets: These are for modeling user interaction directions related to inter-object *macro places*.

(1)Choice Sub-net

The choice operation can be modeled as a free choice net in Fig. 3.14, where t_{choice} has a higher priority. When p_m has a token and t_{choice} is enabled, t_{choice} will fire.

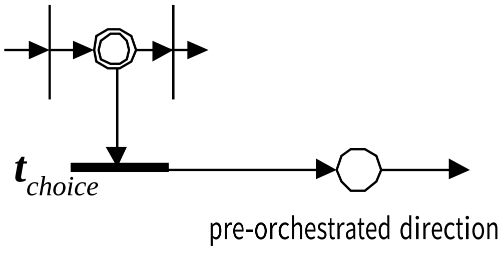


Fig. 3.14 Choice Sub-net.

(2)Wait-for-Choice Sub-net

This sub-net denotes a situation where we need to wait for an user choice, shown in Fig. 3.14. This is done by executing the same LDU (LDU length = 1 frame time) until $t_{user-choice-x}$ with a higher priority is enabled by the user.

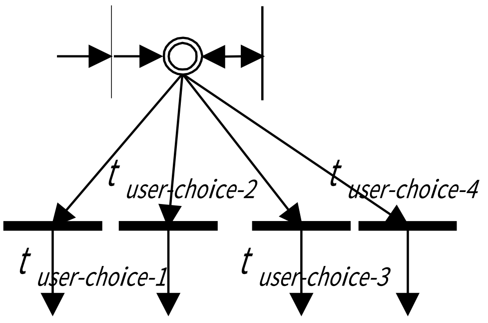


Fig. 3.15 Wait-for-Choice sub-net.

(3)Wait-Some-Time Sub-net

This sub-net represents a situation where we need to wait for an user choice until a specified period. This is done by executing the same LDU (LDU length = 1 frame time) M times $(M=\frac{wait\ \ time}{1\ \ frame\ \ time})$, until $t_{user-choice}$ is fired as shown in Fig. 3.16. Note that $t_{user-choice}$ has a higher priority.

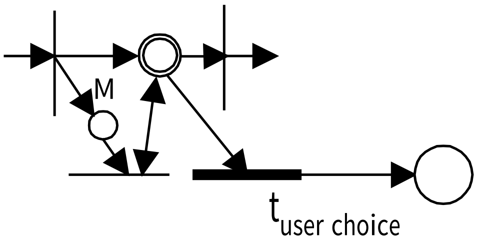


Fig. 3.16 Wait-Some-Time sub-net.

3.7 Scenario Level

An OSPN may develop a very complex structure by using the above sub-nets. Therefore, we define two levels of scenario structures as follows:

Definition 3.4 (*micro level scenario*): A *micro level scenario* is used for modeling *micro places* and describing user interactive choices in detail. The places at this level are all CLDUs.

Figs. 3.17 and Fig. 3.18 show *micro level scenarios* about for *scenarios place* A and E in Fig. 3.19.

Definition 3.5 (*macro level scenario*): A *macro level scenario* is used for representing high-level scenario structure. The scenarios consist of *macro places* and *scenario places*, where the user interactions are implicitly defined in some transitions. At this level, darker place represent *scenario place* P_{sp}. Each *scenario place* represents a segment of a *micro level scenario.* This type of places are user-interactive places, meaning that $|P_{sp}\cdot| > 1$ and $|(\cdot P_{sp})\cdot| = 1$. Each of the other places is a *macro place* P_M, It has exactly one output transition, i.e., $|P_M\cdot| = 1$.

Figs. 3.19 show *macro level scenarios*, place A and place E are *scenario places*, the other are *macro places*.

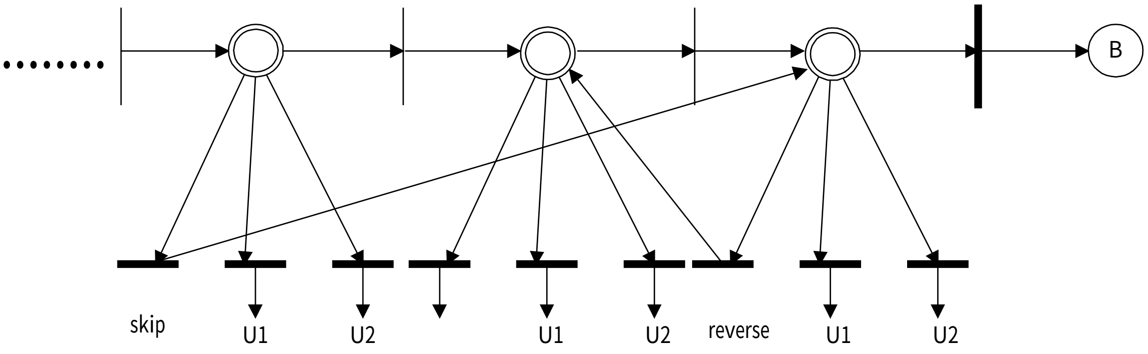


Fig. 3.17 A part of a micro level scenario about scenario place A in Fig. 3.19.

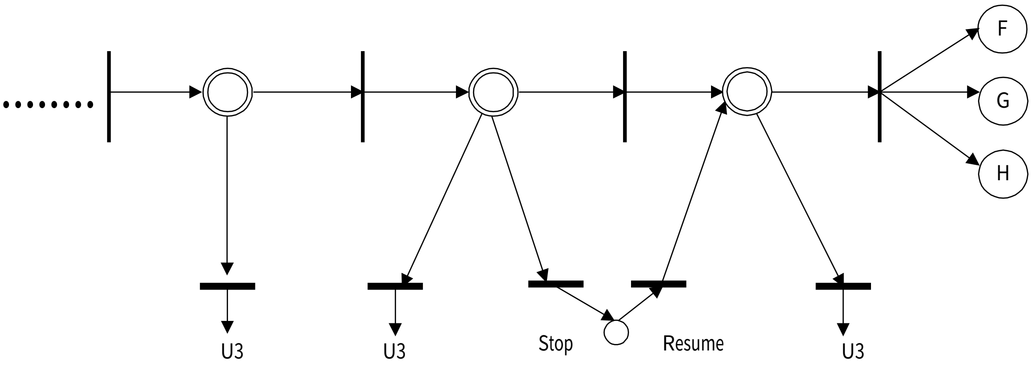


Fig. 3.18 A part of a micro level scenario about scenario place E in Fig. 3.19.

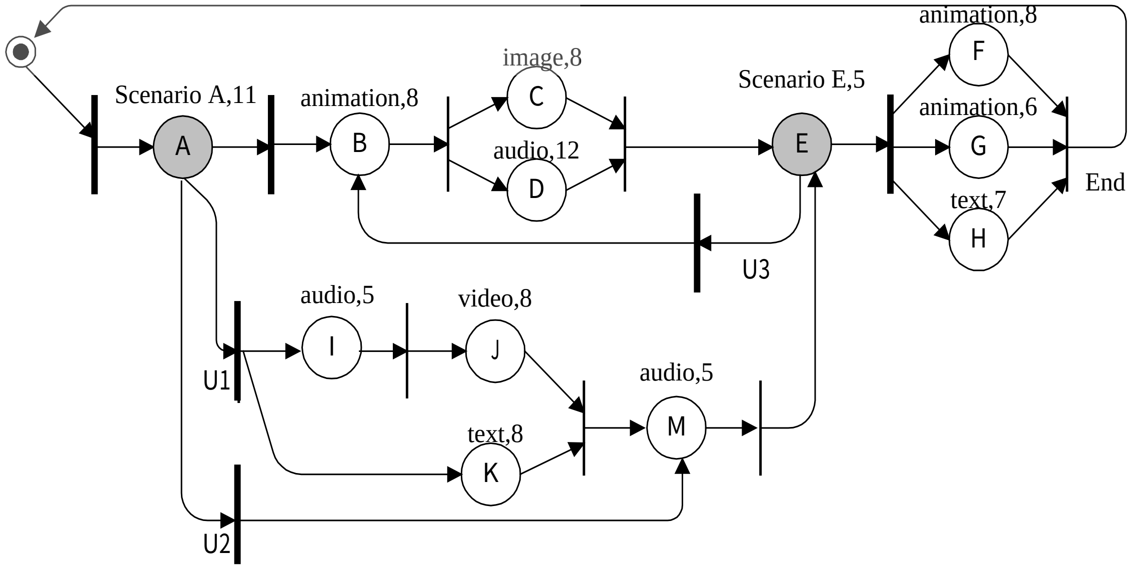


Fig. 3.19 A macro level scenario example.

4. Scenario Synthesis

This section presents a systematic macro-level scenario synthesis approach and formal definitions of two classes of OSPNs deadlock-free *static* and *dynamic scenario nets*.

4.1 Overview of Synthesis Techniques Using the Knitting Technique

A systematic method called *knitting technique* [5] is a rule-based interactive approach. This method has an advantage on reduction of the complexity of synthesis. The knitting technique generates new paths called generations in a Petri net to produce a larger one. The new generations are performed in such a way the resultant net stays live, bounded, and reversible.

In other words the knitting technique contains some simple but effective rules, which can guide the synthesis of Petri nets with the well-behaved properties preserved automatically. The rules contain so-called TT (Transition-Transition) generations among sequential or concurrent processes, and so-called PP (Place-Place) generations sequential or exclusive processes.

As discussed previously, for representing a real-world multimedia macro-level scenario, the OSPN may develop a very complex behavior whose qualitative properties are difficult to verify with reasonably computational cost. To tackle the complexity issue, we extend the above synthesis approach by adding one new rule (i.e., TT.5 as defined later) to compose macro-level scenario nets.

Before presenting our scenario synthesis, we briefly review the knitting technique:

4.2 Definitions Cited

This section reviews definitions cited in [5]. Refer to [5] for detail.

Definition 4.1 (*basic process***)**: A *basic process* (see Fig. 4.1) is a cycle in which 1) $\forall n_i, |\cdot n_i|$

$=|n_i \cdot| = 1$, where $i \in [1, z]$; and 2) a place p_h is marked with a token.

Definition 4.2 (*pseudoprocess, generation point, and joint*): A *pseudoprocess* (*PSP*) (see Fig. 4.2), $\prod$, in a Petri net is either a basic process or a directed path in which any node (transition or place) has only one input node and only one output node except its two terminal nodes: the starting node is defined as the *generation point* $g_{\prod}$ and the ending node as the *joint* $j_{\prod}$.

Definition 4.3 (*Exclusive relationship*): The structural relationship between two PSP, $\prod_1$ and $\prod_2$, is defined as *Exclusive*, that $\prod_1$ and $\prod_2$ are exclusive (EX) to each other, denoted as $\prod_1 | \prod_2$,if $\exists P'_S$ of $\prod_1$ and $\prod_2$.

Definition 4.4 (*Concurrent relationship*): The structural relationship between two PSP, $\prod_1$ and $\prod_2$, is defined as *Concurrent*, that $\prod_1$ and $\prod_2$ are concurrent (CN) to each other, denoted as $\prod_1 \| \prod_2$, if either a) $\exists T'_S$ of $\prod_1$ and $\prod_2$ or b) $\prod_1$ and $\prod_2$ are, respectively, in two separate subnets.

Definition 4.5 (*Sequential/Cyclic relationship*): The structural relationship between two PSP, $\prod_1$ and $\prod_2$, is defined as *Sequential/Cyclic*, If $\prod_1$ and $\prod_2$ are not exclusive, nor concurrent to each other, and either a) $j_{\prod 1} = g_{\prod 2}$ or b) $\forall p \in$ the path from $j_{\prod 1}$ to $g_{\prod 2}$, $p \neq p_h$, then $\prod_1$ is called *sequentially earlier* (SE) to $\prod_2$, denoted as $\prod_1 \rightarrow \prod_2$, and $\prod_2$ is *sequentially later* (SL) to $\prod_1$, denoted as $\prod_2 \leftarrow \prod_1$, $\prod_1$ an $\prod_2$ are also called sequential/cyclic (SQ/CY) to each other, denoted as $\prod_1 \circ \prod_2$.

Definition 4.6 (*maximal set of concurrent places (MSCP)*): A maximal set of places that are concurrent to each other, $\theta = \{p_i | \forall p_i, p_j \in P, p_{I=}p_j$ or $P_i \| P_j \}$. θ_1 is called *sequentially earlier* than θ_2, denoted as $\theta_1 \rightarrow \theta_2$ iff $\forall p_1 \in \theta_1$, either $p_1 \in \theta_2$ or $\exists P_2 \in \theta_2$, $P_1 \rightarrow P_2$. θ_2 is called *sequentially next* to θ_1, denoted as $\theta_1 \mapsto \theta_2$, iff $\forall p_1 \in \theta_1$, either $p_1 \in \theta_2$ or $(p_1 \cdot) \cdot \in \theta_2$. The maximal set of concurrent transitions (MSCT), can be defined similarly.

Definition 4.7 (*local exclusive set*): A *local exclusive set* (LEX) of $\prod_i$ with respect to $\prod_k$, X_{ik}, is the set of all PSP which are exclusive to each other and are equal to or exclusive to $\prod_i$, but not to $\prod_k$. That is, X_{ik} = LEX ($\prod_i$, $\prod_k$) = { $\prod_z | \prod_z = \prod_i$ or $\neg(\prod_z | \prod_K), \forall \prod_{z1}, \prod_{z2} \in X_{ik}$, $\prod_{z1} | \prod_{z2}$ }.

Definition 4.8 (*local concurrent set*): A *local concurrent set* (LCN) of $\prod_i$ with respect to $\prod_k, C_{ik}$ = LCN ($\prod_i, \prod_k$), is the set of all PSP which are concurrent to each other and are equal to or concurrent to $\prod_i$, but not to $\prod_k$, i.e., C_{ik} = C($\prod_i$, $\prod_k$)= { $\prod_z | \prod_z = \prod_i$ or $\prod_z \| \prod_i$, $\neg(\prod_z \| \prod_K)$, $\forall \prod_{z1}, \prod_{z2} \in C_{ik}$, $\prod_{z1} \| \prod_{z2}$ }.

Definition 4.9 (*Handle*): A handle H_1 is an elementary directed path on an elementary circuit, Γ, with starting node of H_1, n_S, branches out to another elementary directed path H_2 which branches back to Γ at the ending node of H_1, n_E. H_1 and H_2 are called complementary to each other. H_1 and H_2 have exactly two nodes in common, i.e., n_S and n_E.

Γ_1 (containing H_1) and Γ_2 (containing H_2) are both elementary circuits (see Fig. 4.3).

Definition 4.10 (*Bridge*): A bridge B is an elementary directed path connecting two complementary handles H1 and H2 (see Fig. 4.3).

Definition 4.11 (*T-component*): Given a Petri net N in which N" is a subnet of N and N" ($\subseteq$N) is a *T-component* (*P-component*) of N iff N" is strongly connected marked graph (state machine) and $P'' = \bullet T'' \cup T'' \bullet$ ($T'' = \bullet P'' \cup P'' \bullet$) (see Fig. 4.4).

Definition 4.12 (*Synchronized Choice Petri net*): A Petri net is synchronized choice, if every TP- (PT-) handle has a PT- (TP-) bridge from its complementary TP-handle to itself. That is, any two outgoing paths from a transition(place), called T'_S [P'_S], must join at a transition T'_E (place P'_E) and these paths do not contain a P'_E (T'_E).

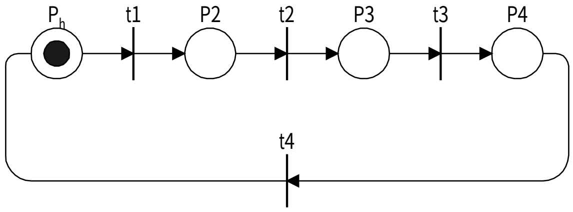


Fig. 4.1 Illustration of a basic process.

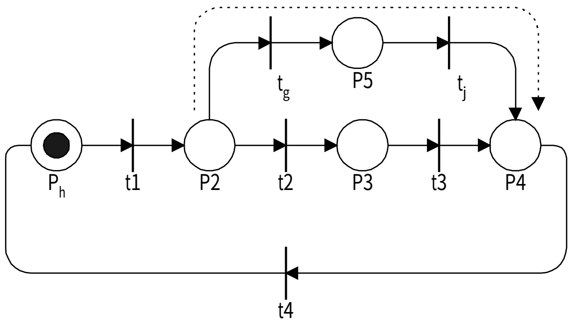


Fig. 4.2 Illustration of a pure generation (PP.1 rule). The PSP [P2 tg P5 tj P4] is a PP.

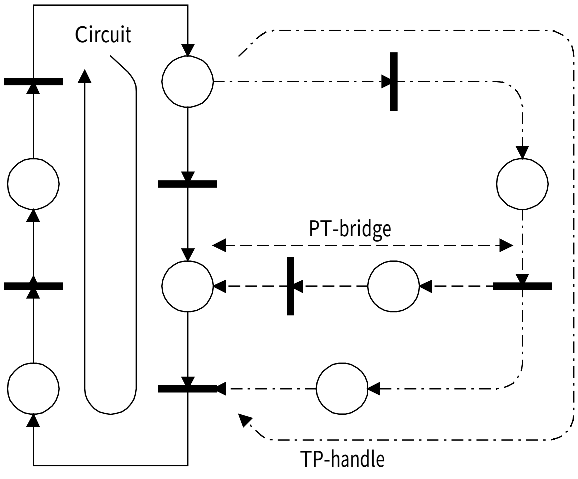


Fig. 4.3 Illustration of PT-handle and TP-bridge.

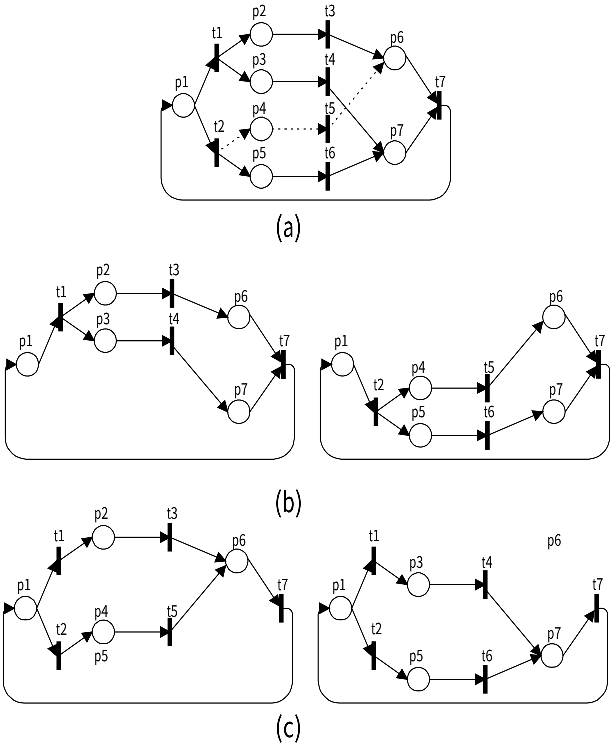


Fig. 4.4 Illustration of T-Components ((b)) and P-Components ((c)).

4.3 Knitting technique generation rules

The TT and PP rules are described as follows:

TT Rules:

For a new path $\prod^{\omega}$ from t_g to t_j, with $\neg(t_g \mid t_j)$,

1) TT.I: If $\prod_g = \prod_j$, then return (see Fig. 4.6).

2) TT.2: If $t_g \leftarrow t_j$ or $t_g = t_j$, then insert a token in a place of $\prod^{\omega}$.

3) TT.3: If there was no path from t_g to t_j prior to this generation or they are respectively in two cycles without direct connection, then generate a new TT-path from t'_g to t'_j such that t_g, t_j, t'_g and t'_j are in a cycle in this order.

4) TT.4 (see Fig. 4.7):

   a) TT.4.1: Generate a TP-path from a transition $t_x \in \prod_x$ $\forall \prod_x \in X_{gj}$ to a place $P_{k1} \in \prod^{\omega}$.

   b) TT.4.2: Generate a virtual PT-path from the place $P_{k2} \in \prod^{\omega}$ with $P_{k2} \in \cdot t_j$ to a transition $t_y \in \prod_y$ $\forall \prod_y \in X_{jg}$.

PP Rules:

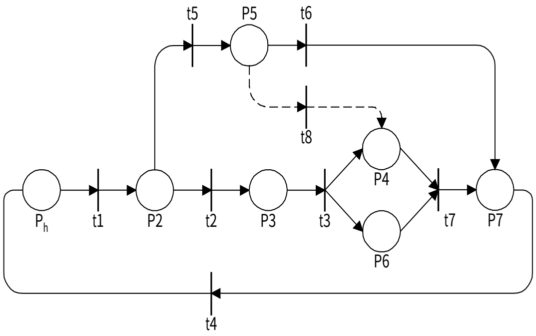
For a new path $\prod^{\omega}$ from $P_g \in \prod_g$ to $P_j \in \prod_j$, with $\neg(P_g \parallel P_j)$,

1) PP.1 If $\prod_g = \prod_j$, then return.

2) PP.2

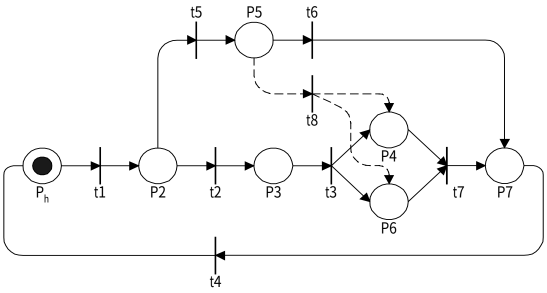
   a) PP.2.1 Generate a virtual *PT-path* from a place $P_x \in \prod_x$, $\forall \prod_x \in C_{gj}$, to the transition $t_{k1} \in \prod^{\omega}$ with $t_{k1} \in p_g \cdot$.

   b) PP.2.2 Generate a *TP-path* from a transition $t_{k2} \in \prod^{\omega}$ to a place $p_y \in \prod_y$ $\forall \prod_y \in C_{jg}$ (see Fig. 4.5).



(a)

Fig. 4.5 (a) PSP [P5 t8 P4] is generated form P5 to P4. (b)New paths [t8 P6] are generated to fit the PP.2.2 rule.



(b)

Fig. 4.5 Illustration of pure PP generation (PP.2 rule).

4.4 Scenario synthesis base on the knitting technique

In our interactive multimedia presentation system, a scenario that is independent of the user interaction sequence is modeled as a *Static Scenario Net*. That is, the user interaction transitions are mutually independent. This can be created by using the TT generations rules among sequential or concurrent scenario synthesis and the PP generations rules among sequential or exclusive scenario synthesis.

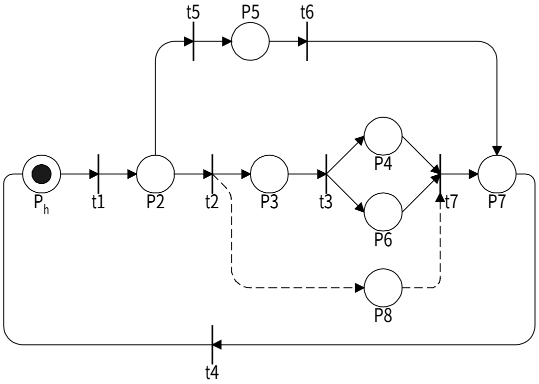


Fig. 4.6 Illustration of TT.1 generation.

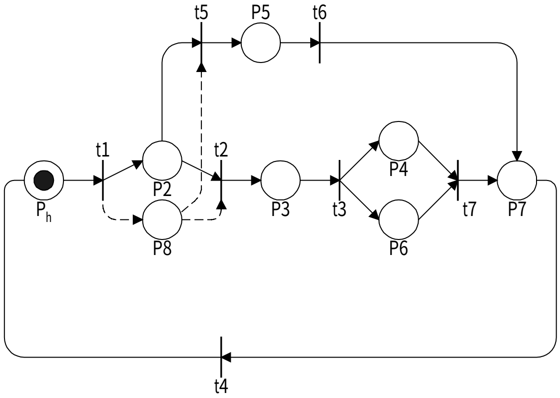


Fig. 4.7 Illustration of TT.4 generation.

We now define *Static Scenario Nets* as follows:

Definition 4.13 (*Static Scenario Nets*) An OSPN $N^s=\{T_i \cup T_u, P_M \cup P_{sp}, A, D, R, M\}$ is called a *static scenario net* (SSN)(see Fig. 4.8). if the following conditions are satisfied:

1. The Petri nets are synthesized using the *knitting technique*, starting with a *basic process*.
2. N^s is strongly connected and no circuit of N^s has a *PT*- nor *TP-handle*.
3. $\forall t \in T_u, |\cdot t| = 1, \cdot t \in P_m, (\cdot t)\cdot > 1$ *and* $\{(\cdot t)\cdot\} \subset T_u$
4. At the *scenario place* $\forall p \in P_{sp}, |p\cdot| \geq 1$, i.e., the forking arc can only be constructed at the *scenario place*.
5. At the macro level $\forall p \in P_M, |p\cdot| = 1$, i.e., each place has exactly one output transition.

In our interactive multimedia presentation system, a scenario that is depends on the user interactive sequence is modeled as a *dynamic scenario net*. That is, the firing of some user interactive transition may affect the firings of others. We now define *dynamic scenario nets* below:

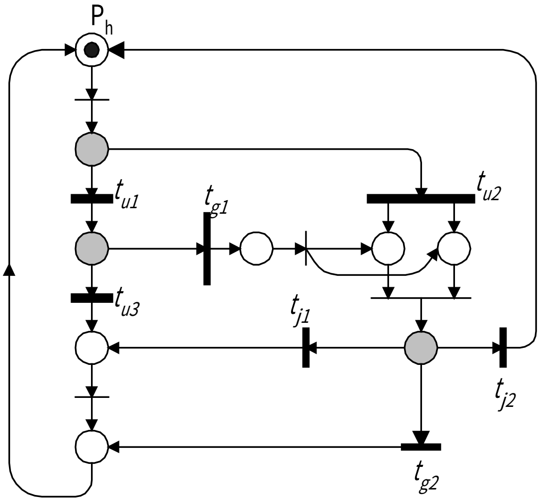


Fig. 4.8 Example of a static scenario net.

Definition 4.14 (*dynamic scenario nets*): An OSPN $N^d = \{T_i \cup T_u, P_M \cup P_{ps} \cup P_s, A \cup A_s, D, R, M\}$ is called an *dynamic scenario nets* (DSN) if the following conditions are satisfied:

1. $P_s=\{p_{s1}, p_{s2}, \ldots p_{sn}\}$ is a set of so-called *status places*, which are used to record the inter-relationship of *user interaction transitions*
2. $\forall p \in P_s,\ |\cdot p| = |p\cdot| = 1, \cdot p = t_g \in T_u$

$and\ \ p\cdot = t_j \in T_u$

3. $A_s = \{T_u \times P_s\} \cup \{P_s \times T_u\} \to I$ is a set of directed arcs or inhibitor arcs.
4. A DSN is synthesized from an SSN adding some TT-Bridges following TT.5 generation rule below.
5. $\forall p \in P_s, \exists T_1, T_2 \subset TCS,$ $p_h \cup \cdot t_g \cup \cdot t_j \subset T_1, p_h \cup t_g \cdot \cup \cdot t_j \subset T_2$, where TCS is the set of all T-components of N^s.
6. $\forall p \in P_s,$ $p \not\subset \{\cdot t_g\} \cup \{t_g \cdot\} \cup \{\cdot t_j\} \cup (t_j \cdot\}$.
7. $\forall p \in P_s, \exists t_1 \subset \{(\cdot t_g)\cdot\},$ $|\cdot t_1| = 1, \exists t_2 \subset \{(\cdot t_j)\cdot\}, |\cdot t_2| = 1$.

4.5 Dynamic scenario nets generation rules

For a new path $\prod^{\tau}$ from t_g to t_j when $t_g, t_j \in T_u$,

TT.5: IF the following conditions are satisfied:

(1) $\exists T_1, T_2 \subset TCS,\ p_h \cup \cdot t_g \cup \cdot t_j \subset T_1,$ $p_h \cup t_g \cdot \cup \cdot t_j \subset T_2$

(2) $P_s \not\subset \{\cdot t_g\} \cup \{t_g \cdot\} \cup \{\cdot t_j\} \cup (t_j \cdot\}$

(3) $\exists t_1 \subset \{(\cdot t_g)\cdot\}, |\cdot t_1| = 1,$ $\exists t_2 \subset \{(\cdot t_j)\cdot\}, |\cdot t_2| = 1$

Then

(1) Generate a status place p_s between t_g and t_j ; and respectively generate a *TP-path*, *PT-path* from t_g to p_s and from ,p_s to t_j.

(2) Generate a place p_b with a token between t_j and t_g ; and respectively generate a *TP-path*, *PT-path* from t_j to p_b and from p_b to t_g .

Example 1: When we wish to generate a status place to the net, we check three-conditions:

(1) $\exists T_1, T_2 \subset TCS, p_h \cup \cdot t_g \cup \cdot t_j \subset T_1,$ $p_h \cup t_g \cdot \cup \cdot t_j \subset T_2$;

(2) $P_s \not\subset \{\cdot t_g\} \cup \{t_g \cdot\} \cup \{\cdot t_j\} \cup (t_j \cdot\}$;

(3) $\exists t_1 \subset \{(\cdot t_g)\cdot\}, |\cdot t_1| = 1, \exists t_2 \subset \{(\cdot t_j)\cdot\},$. $|\cdot t_2| = 1$

The T-components for the SSN in Fig. 4.8 are shown in Fig. 4.10(b) -(h).Fig. 4.10(f) is the T-component of T_1, where $p_h \cup \cdot t_{g_1} \cup \cdot t_{j1} \subset T_1$. Fig. 4.10(g) is the T-component of T_2, where $p_h \cup t_{g1} \cdot \cup \cdot t_{j1} \subset T_2$. Fig. 4.10(c) is the T-component of T_3 ,where $p_h \cup t_{g2} \cdot \cup \cdot t_{j2} \subset T_3$. Fig. 4.10(d) is the T-component of T_4 ,where $p_h \cup \cdot t_{g_2} \cup \cdot t_{j2} \subset T_4$. Thus we can add a status place to the net of Fig. 4.8. Fig. 4.9 shows the resultant net that is live, bounded and reversible refer to section 5.

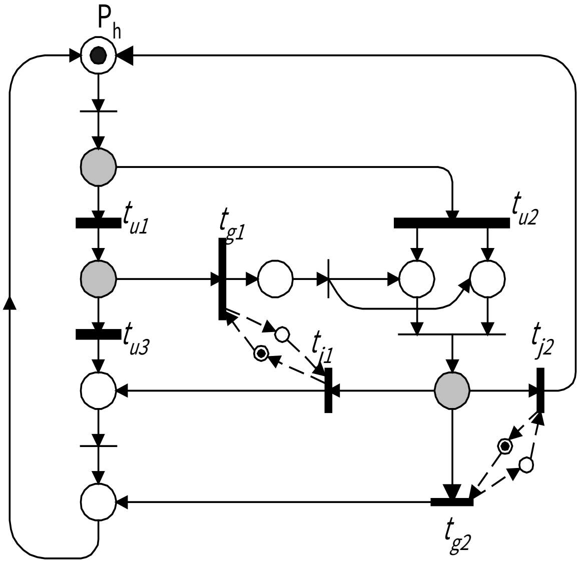


Fig. 4.9 Example of a dynamic scenario net

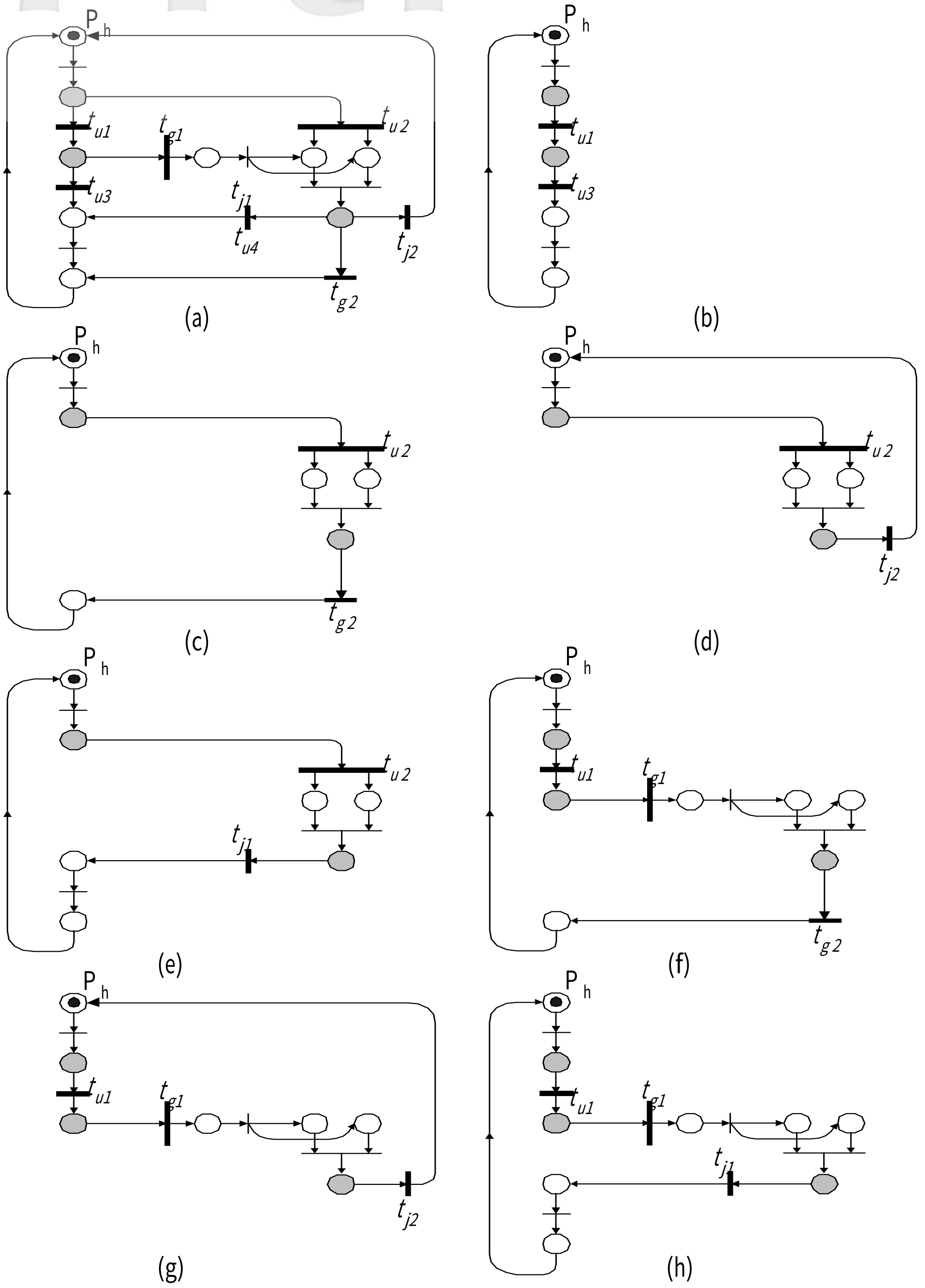


Fig. 4.10(b)-(h) are T-components of (a); (f) is the T-component of T_1 ,where $p_h \cup \cdot t_{g_1} \cup \cdot t_{j1} \subset T_1$; (g) is the T-component of T_2, where $p_h \cup t_{g1} \cdot \cup \cdot t_{j1} \subset T_2$; (c) is the T-component of T_3, where $p_h \cup t_{g2} \cdot \cup \cdot t_{j2} \subset T_3$; (d) is the T-component of T_4,where $p_h \cup \cdot t_{g_2} \cup \cdot t_{j2} \subset T_4$.

5. Properties Analysis

This section describes some properties interpretation in multimedia systems and analyzes the system model for three properties,i.e., safeness, liveness, and reversibility.

5.1 Properties related to multimedia presentation system

In multimedia presentation system, *Reachability* shows that each state can be reached. Applying the property you can verify all the data, that you can access in your view. The purpose of *liveness* is to verify whether deadlocks happen in the conceptual layer. A transition is live if and only if under any reachable marking M we can find a firing sequence that transforms M to a marking that enables the transition. If a Petri net is *conservative*, then the weighted number of the tokens in the net does not change. However, the initial and final number of tokens always remains equal to one, as discussed.

5.2 Properties Analysis

Theorem 1: A Static Scenario nets is a synchronized choice (SC) net and it’s structurally live & structurally bound.

Proof: We synthesized OSPN Static Scenario using the knitting technique method. In a static scenario net N^s, no circuit has a PT-, nor TP-Handle, so it’s a strongly connected net . We know that it’s TP-decomposable net. It’s equivalent to the synchronized choice net. This proof is given in reference [5], and thus, an SSN is structurally live & structurally bound net.

Theorem 2: A Dynamic Scenario nets is safe, reversible, and live.

Proof: We know that SSN is synthesized by the knitting technique. When we do not add any status place to SSN, the original net is a SC net. Then we use TT.5 generation rule to add a status place to SSN. TT.5 generation has only added one status place and two arcs to connecting them between t_g and t_j, and $\exists T_1, T_2 \subset TCS,\ p_h \cup \cdot t_g \cup \cdot t_j \subset T_1,$ $,\ p_h \cup t_g \cdot \cup \cdot t_j \subset T_2$ where TCS is the set of all T-components of N^s, and T_1 and T_2 belong to two different but intersected T-components, $p_h \cup \cdot t_g \cup \cdot t_j \subset T_1$, and $p_h \cup t_g \cdot \cup \cdot t_j \subset T_2$, we know that there exist a firing sequence that transfer a token in one place to another among the set of place $\{\cdot t_g, t_g \cdot, \cdot t_j, p_h, t_j \cdot\}$. That is, a token in any of the above places can reach any other of the above place. Similarly, when we add more status places, we restrict that $P_s \not\subset \{\cdot t_g\} \cup \{t_g \cdot\} \cup \{\cdot t_j\} \cup \{t_j \cdot\}$. Thus, we obviously obtain that every *user interactive transitions* is potentially fireable, every transitions of a DSN net N^d can fire at least once. Using the same method as above, we can obtain a DSN satisfying DSN generation rules in restriction TT.5 that is still safe, reversible and live.

6. Application

In this section, we present a detailed CAI system[19], which uses the ideas and models presented in the preceding sections. This example system contains both time independent and time dependent media objects in the form of audio, image and textual data that require synchronization at the presentation level. Fig. 6.1 - 6.2 show some screens of the system. This is a course of morality and health in primary schools for educating students to behave well in daily life.

Fig. 6.1 is the main screen of the system, which can be modeled by a static scenario Part-A in Fig. 6.5 is an SSN structure, where there have four buttons, i.e., **“設定說明”, “結束”,”我會按時作習養成良好習慣”** and **“我會與人相親相愛並對人有禮貌”**. When **”我**

會按時作習養成良好習慣” is pushed, the scenario will go to that “I will behave well on daily scheduling” as shown in Fig. 6.3. When **“我會與人相親相愛並對人有禮貌”** is pushed, the scenario will go to that “I will love and be polite to every person”. Fig. 6.2 is a micro level generated and reduced as discussed in section 4, where the scenario contain *wait-for-choice* and *choice* sub-nets.

Fig. 6.3 is the screen after **”我會按時作習養成良好習慣”** is pushed, It can be constructed by a *static scenario net*. Part-B in Fig. 6.5 is an SSN structure. There have six buttons, i.e., “**返回**”, “**說明**”, “**生活習慣**”, **“按時作習”**, “**餐桌禮儀**”, and “**飲食習慣**”. When “**生活習慣**” button is pushed, the scenario will go to “life habits” in Fig. 6.4. When **“按時作習”** button is pushed, the scenario will go to “Scheduling on Time”. When “**餐桌禮儀**” button is pushed, the scenario will go to “Diet habits”.. When “**飲食習慣**” button is pushed, the scenario will go to “Table manners”. In this scenario each button represents a user interaction transition. These scenarios are mutually independent.

Fig. 6.4 is the screen after **“按時作息”** is pushed, It can be constructed by a *dynamic scenario net*. Part-C in Fig. 6.5 is it a DSN structure. There have four buttons to describe “life habits”, i.e., “**返回**”, “**測驗**”, “**上午**” and “**下午**”. When “**返回**” is pushed, the scenario will go back to Fig. 6.3. When “**測驗**” is pushed, the scenario will go to the Instruction-Test Unit. When “**上午**” is pushed, the scenario will go to “Morning Scheduling in one day”. When “**下午**” is pushed, the scenario will go to “Afternoon Scheduling in one day”. In this scenario, the firing of “**測驗**” is dependent on whether “**上午**” and “**下午**” are fired ahead or not, i.e., “Morning Scheduling and Afternoon Scheduling in one day” must be fired once before the Instruction-Test Unit.

7. Conclusions and Future Research

In this paper, we propose multimedia synchronization models based on timed Petri nets, called *Object Slice Petri Nets*. In the modeled system, the user is allowed to manipulate the presentation sequence such as to repeat a sub-sequence, choose a sequence branch, change the playing speed, and reverse the playing sequence through by keyboard, mouse or program. The formal model can describe synchronization behaviors and different phenomena in multimedia systems. We notice that there have been similar approaches in prior work [3,14]. However, all these approaches can not deal with all possible user interaction scenarios as in our approach. In addition, their approaches require the modification of the firing rule. One of the advantages of our approach is that because an OSPN is based on a timed Petri net, we do not change the firing rule. As a result, existing analysis methods of the Petri net theory can be used to analyze an OSPN.

For representing a real-world multimedia scenario, OSPN may develop a very complex behavior whose qualitative properties are difficult to verify with reasonably computational cost. We also proposed a systematic synthesis method based on an extension of the *knitting technique* used for our macro-level scenario synthesis to synthesize well-behaved Nets. By this way, we present a class of *Static Scenario Nets* and *Dynamic Scenario Nets*. *Static Scenario Nets* can be applied to scenarios independent of the user interactive sequence. *Dynamic Scenario Nets* can be applied to scenarios dependent of the user interactive sequence. Sub-net structures have been defined for special characteristics of the system.

Fig. 6.1 An CAI system.

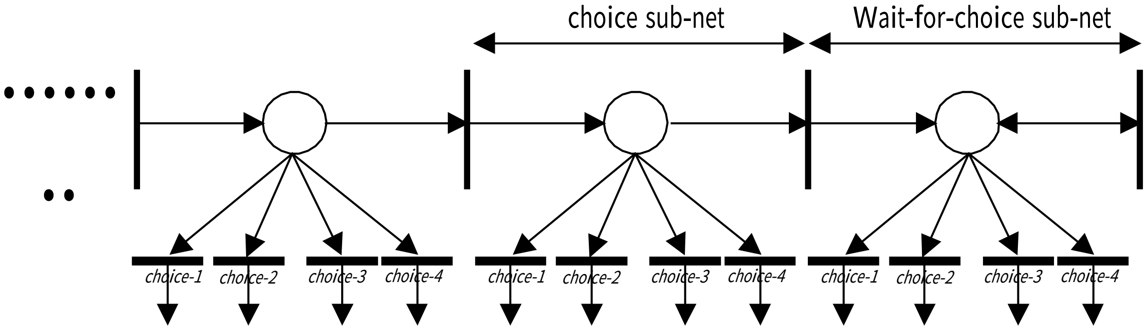


Fig. 6.2 A part of a micro level scenario in Fig. 6.1.

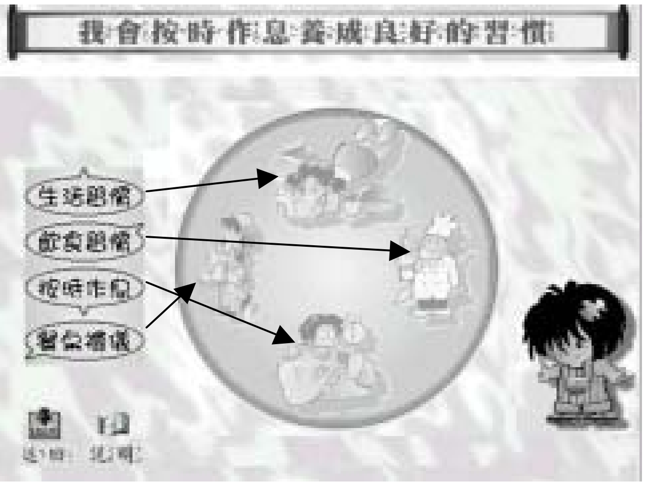


Fig. 6.3 The screen after **”我會按時作習養成良好習慣”** is pushed.

Fig. 6.4 The screen after **“按時作息”**is pushed.

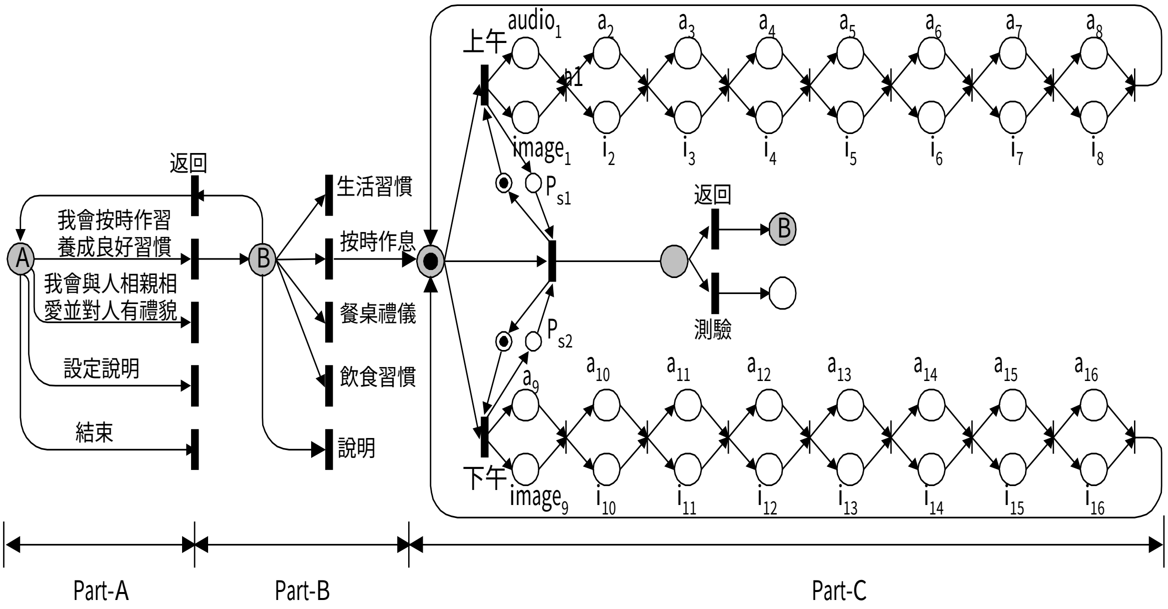


Fig. 6.5 A macro level scenario of the CAI.

The proposed methodology is currently being implemented in a software package. For constructing temporal relationships among media streams, an OSPN event editor is being developed for helping the user to create multimedia documents with the specified temporal relationships. However, a timeline representation is more understandable and instinctive for users to create multimedia documents. Thus, the automatic transformation technique between a timeline representation and an OSPN model will be realized.

In this paper we only considered the temporal composition among media objects in the concept layer. The spatial composition is another important topic for multimedia synchronization. A complete model that deals with temporal and spatial compositions is difficult to be constructed. We leave these research topics to future work.

References

[1] T. D. C. Little and A. Ghafoor, "Synchronization and storage models for multimedia objects," *IEEE Journal on Selected Areas in Communications*, Apr. 1990, vol. 8, pp413-426.

[2] G. Blakowski and R. Steinmetz. "A media synchronization survey: Reference model, specification, and case studies." *IEEE Journal on Selected Areas in Communications*,Jan.1996, vol. 4,pp5-35 No. 1.

[3] B. Prabhakaran, S. V. Raghavan, "Synchronization models for multimedia presentation with user participation " *Multimedia System* , 1994, vol. 2, pp53-62,.

[4] R. Steinmetz, "Synchronization properties in multimedia systems" *IEEE Journal on Selected Areas in Communications*, Apr. 1990, vol. 8, pp401-412.

[5] D. Y. Chao and D. T. Wang "Two Theoretical and Practical Aspects of Knitting Technique: Invariants and A New Class of Petri Net" *IEEE Transactions on Systems, Man, and Cybernetics*, Dec. 1997, vol. 27, part B, no. 6.

[6] T. Murata, "Petri Nets: Properties, Analysis and Application," *Proceedings of IEEE*, Apr. 1989, vol. 77, pp. 541-580.

[7] Feng Chu and Xiao-Lan Xie "Deadlock analysis of Petri Nets Using Siphons and Mathematical Programming" *IEEE Transactions on Robotics and Automation*, Dec. 1997, vol. 13, , no. 6.

[8] Javier Esparza and Manuel Silva "Circuits, Handles, Bridges and Nets," *LNCS*, *Advances in Petri Nets*. ,1991, pp. 210-242..

[9] F. DiCesare and M. D. Jeng, "Synthesis for Manufacturing Systems Integration," Chapter 3, *Practice of Petri Nets in Manufacturing*, Chapman & Hall, Ltd., London, UK, 1993, pp. 103-146.

[10] K. H. Lee, J. Favrel, "Hierarchical reduction method for analysis and decomposition of Petri nets", *IEEE Trans. Systems, Man, and Cybernetics*, Mar./Apr. 1985,vol. 15, no. 2.

[11] C. Hamblin, "Instants and intervals", *1st conf. Int. Society for the Study of Time*, 1972, pp. 324-331.

[12] C. Parris, "Computer or teleputer ?", *Commun. ACM*. 1990,Vol. 33,pp.29-36, Sept.

[13] D. Y. Chao and D. T. Wang "Knitting technique with TP-PT generations for Petri net synthesis ," *Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics*, 1995, Vol.2 , pp.1454 –1459.

[14] M. Woo , N. U. Qazi and A. Ghafoor "A Synchronization Framework for Communication of Pre-orchestrated Multimedia Information", *IEEE Network* , 1994. Jan./Feb. , pp. 52-61.

[15] M. E. Hedges, R. M. Sasnett, and M. S. Ackerman," A construction set for multimedia applications," *IEEE Software*, Jan. 1989, pp. 37-43.

[16] Y. Y. Al-Salqan and C. K. Chang, “Temporal relations and synchronization agents” *IEEE Multimedia*, Sum. 1996 Vol.32, pp30 –39.

[17] K. Yoon abd P. B. Berra, “TOCPN: interactive temporal model for interactive multimedia documents,” *Proceedings of 1998 IEEE International Workshop on Multi-Media Database Management Systems*, Aug. 1998 , pp.136 –144.

[18] T. D. C. Little and A. Ghafoor, " Spatio-temporal composition of distributed multimedia objects for value-added networks," *IEEE Computer*, Oct. 1991, Vol. 24, pp42 - 50.

[19] 魏慶雲、翁全志、謝秉芳、翁鶴瑞、范佐安、劉昭政, ”道德與健康”, 1996,臺灣省政府教育廳.

[20] D. Y. Chao, “Knitting Technique and Structural matrix for deadlock analysis and synthesis of Petri nets with sequential exclusion.” , Dec. 1997 資管評論 Vol. 7.

MuDer Jeng received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1992.He is a full Professor of Electrical Engineering at National Taiwan Ocean University, Keelung, Taiwan, R.O.C. His research interests include Petri nets, neural networks, computer integrated manufacturing, semiconductor manufacturing systems, multimedia systems, and underwater robotics. In the above areas, he has authored over 70 publications in journals, books, and conference proceedings. Dr. Jeng received the Franklin V. Taylor (Outstanding Paper) Award from IEEE Systems, Man and Cybernetics (SMC) Society in 1993. He is a member of Phi Tau Phi. He has served as a Guest Editor of a 1998 special section of IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING and a 1999 special issue of the *International Journal of Intelligence Control and Systems*.

YuanLin Wen is now working toward the Ph.D. degree in the Electrical Engineering Department at National Taiwan Ocean University, Keelung, Taiwan, R.O.C.