

Adapting Spatial Constraints of Composite Multimedia Objects to achieve Universal Access

Ahmed Gomaa
Rutgers University, CIMIC
ahgomaa@cimic.rutgers.edu

Nabil Adam
Rutgers University, CIMIC
adam@cimic.rutgers.edu

Vijayalakshmi Atluri
Rutgers University, CIMIC
atluri@cimic.rutgers.edu

Abstract

A composite multimedia object (cmo) is comprised of different media components such as text, video, audio and image, with a variety of constraints that must be adhered to. The constraints are 1) rendering constraints that comprise the temporal and spatial constraints between different components, and 2) behavioral constraints that include the security and fidelity constraints on each component. Different users have different 3Cs, which are: capabilities (e.g., monitor size), characteristics (e.g., age) and credentials (e.g., subscription to service). The focus of this paper is on addressing the problems of (1) specifying a consistent cmo that “automatically” adapts its spatial constraints to different user’s devices. (2) Identifying the conflicts that might occur between the temporal and spatial constraints when having different monitor resolution that displays the cmo by means of reachability analysis of colored time Petri net (3) Resolving the identified conflicts automatically to render a cmo that is error-free when rendered at different user devices.

1. Introduction

A composite multimedia object (cmo) is comprised of different media components such as text, video, audio and image, with a variety of constraints that must be adhered to. The constraints include 1) rendering relationships that comprise of temporal and spatial constraints among different components, 2) behavioral requirements that include the security and fidelity constraints on each component. Typically, all these constraints are specified at the time of authoring the cmo. Associated with each component is a set of parameters, such as display information, modality, and playback duration. Putting the cmo constraints and parameters together leads to a cmo plan that describes how the cmo will be rendered at the user device.

Different users have different 3Cs: Capabilities (e.g. monitor size), Characteristics (e.g. age) and Credentials (e.g. subscription to services). A user requesting access to a cmo may not possess the required credentials to meet

the security constraints, or his/her computer system may not possess the desired capabilities to meet the spatial or fidelity constraints. This necessitates modification to the original cmo [AAGA03].

The Spatial constraints are the constraints that specify the spatial relationship between the media components. Let us assume that a part of the composite multimedia object has the following spatial constraints: Image B ABOVE image A, where Image A = 200 x 300 pixels and Image B = 200 x 300 pixels. If the user monitor resolution is $\geq 400 \times 600$, then there will be no problem, but if the user monitor is $\leq 400 \times 600$ for example, 640×480 then we either: (1) Drop image A or B, (2) Resize A or B or (3) Change the spatial constraint ABOVE

If we resize the images, we will not affect the rendering relationships, but if resizing the image may lead to misinterpretation by the viewer, e.g., image A is an x-ray showing a cancer tumor, it will be better to drop the spatial constraint, which will affect the rendering relationships, but will not lead to a misinterpretation. Furthermore, if the monitor screen may not hold more than one component at a time the system may also drop the temporal constraints, leading to further adaptation of the rendering relationships. The different resolution options appears in Figure 1

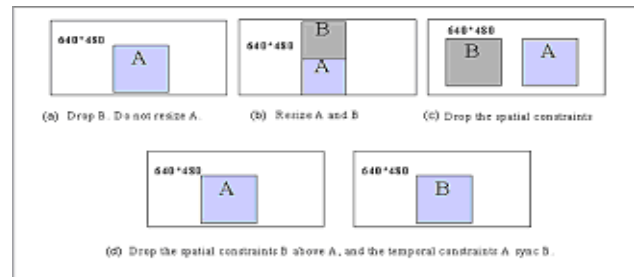


Figure 1: Different resolution for mismatches. (a) Drop one component. (b) Resize all components. (c) Drop spatial constraints. (d) Drop spatial and temporal constraints.

Existing approaches deal with the spatial constraints problem simply by either resizing the components to be

displayed to fit the users monitors, using the available space and allowing minimum overlapping [BS00] or by presenting a cascade layout of the components [LL04]. None of the existing approaches considers the interrelated *cmo* constraints, such as temporal or security constraints along with the spatial constraints. The primary draw back of not considering all the constraints is that: by simply trying to fit every thing in the user monitor, the *cmo* presentation might not make a lot of sense to the viewer.

In this paper, we allow the *cmo* author to classify the constraints at the authoring time as either optional or mandatory. This classification enables the dropping of the optional constraints if they do not match the user 3Cs instead of simply dropping the components that do not satisfy the constraints due to the lack of user 3Cs as both [AAGA03], [AAAB01] suggest. We focus our study on the different user monitor resolution effect on the spatial and temporal *cmo* constraints and the conflict that might occur between those constraints due to different user monitor resolutions.

1.1 The problem statement

The focus of this paper is on addressing the problems of (1) specifying a consistent *cmo* that “automatically” adapts its spatial constraints to different user’s devices. (2) Identifying the conflicts that might occur between the temporal and spatial constraints when having different monitor resolution that displays the *cmo*. (3) Resolving the identified conflicts automatically to render a *cmo* that is error-free when rendered at different user devices.

The paper is organized as follows: section 2 presents preliminaries for our approach. Section 3 discusses the proposed approach. Section 4 presents the related work. Conclusion and future work are discussed in Section 5.

2. Preliminaries

In this section, we present the modeling of an adaptive *cmo* by including the temporal, fidelity and security constraints [AAG03]. A color Petri net [J86] is a mathematical tool with graphical representation, where places are circles, tokens a colored dots, transitions are boxes, and arcs are arrows between places and transitions. A place has an associated color set type determining the value of data that the place may contain. The data is represented in the form of dots named tokens. Tokens are differentiated by colors, which may be arbitrary data values. The transitions are the active part of a system that can change the value of the data. A token can enter a place only if its color is an element in the color set of the place. The arcs are incidence functions that pass the data

from places to transitions and from transitions to places. The marking of a place is the existence of tokens in it. The state of the Petri net is the distribution of the tokens in all the places of the net.

2.1 Temporal, fidelity and security constraints in color Petri net

In order to effectively represent the temporal, fidelity and security constraints of the multimedia object, [AAGA03] propose to use a color-time Petri net where each place models a media component that is associated with a set of parameters, $\langle da, fd, pd \rangle$ where *da* is the display attribute, *fd* is the component fidelity and *pd* is the playback duration of the component. Unlike traditional color Petri net, they use two types of tokens: a regular token (represented as a filled dot) and a hole (represented as a circle). When a token enters a place, it is said to be available after the duration of the place has elapsed. On the other hand, when a hole enters a place, it becomes available as soon as it enters. As far as the dynamic behavior of the Petri net is concerned, tokens and holes behave the same way.

3. Proposed Approach

The major steps of our proposed approach are to (1) define the spatial constraints specification for the *cmo*, (2) develop a conceptual model using Color time Petri net (MMCTPN), (3) map the spatial constraints specification to MMCTPN, and (4) analyze MMCTPN by using the model analysis techniques to identify the spatial constraints specification mismatches with the user monitor. (5) Reconstruct the spatial constraints to suit the user monitor.

3.1 The spatial constraints specification

[PYY03] specifies the spatial attributes of a component P_1 as follows: 1) the starting address AR_2 of a reference component P_2 , which is the upper left corner coordinates of this component. 2) The display attributes DA_1 of the component P_1 , which is the height, width and, priority 3) the spatial relation RR between P_1 and P_2 . [PYY03] calculates the starting address of a component P_1 using an address function AF , where $AR_1 = AF(AR_2 \times DA_1 \times DA_2 \times RR)$. To determine if there is enough space to insert the component P_2 , we need to calculate the list of largest empty rectangles based on the spatial relations between the two components. We use the incremental adding/ deleting algorithms proposed by [BS00] for this task, where the available empty rectangles are reduced/ increased by the new added/ removed rectangle. The

input of the algorithm is the available largest empty rectangles (LER) and the output is the remaining largest empty rectangles after inserting the second component as a new rectangle (R), where R is obtained using the starting address of the inserted component and its height, width and priority. We simply consider if the added rectangles falls within the boundaries of the monitor. If the inserted rectangle $(R) \subset (LER)$, then the available space in the monitor $AV(P)$ for the component P is set to 1 by the value function VF, else to -1. The Value function is defined as follows: $AV(P) = VF(AR_j(P), DA_j(P), LER)$.

3.2 Developing a conceptual model using Color time Petri net- (MMCTPN).

In this definition, we build on the work of [AAGA03] and [PYY03] by presenting the spatial constraints in Petri net along the temporal, security and fidelity constraints. We model all the cmo constraints in one uniform model to allow the authoring of an adaptable cmo.

The following definition states that there exist four types of places. 1) Media place (MP) that represents the component. 2) Address place (AP), that buffers the spatial information for the synchronous components, specifically, (AP) buffers the display information of the component (DA), the starting address of the component (AR), the list of empty rectangles (LER) and the availability of space for the coming component (AV). Delay place (DP) that helps in prioritizing the firing of spatial transitions. Both (AP) and (DP) has a null value for (AR) and (DA). A null place (NP) connects any two subsequent temporal transitions where at least one media place exist between them. Each place is associated with a color set (i.e., denoted as $C(p)$), which specifies the set of allowable colors of the tokens {token, s-token, holes) to enter the place. Address place, delay places and null places accept all colors with no time duration. The clock value $D(p_i)$ is used to map the state of a place. The current clock value represents how the place has been active so far from the beginning of its activation. While a place keeps a token, (marked), the clock of the place keeps ticking.

Given a transition T, we denote its input set of places by denoted $\bullet T$ or $\bullet P$ and its output set of places by denoted $T \bullet$ or $P \bullet$, There are two types of transitions T, temporal (TT), and Spatial (ST). A temporal transition TT is enabled only if all of its input places $\bullet T$ are marked and the tokens/ holes and s-tokens are available. Temporal transitions that are enabled fires. A spatial transition ST is enabled if its input place $\bullet T$ is marked with a token with

type {s_token} and the token is available. ST has one input place $\bullet T$ and one output place $T \bullet$.

The marking M (P_i) of a place P_i represents the number of tokens of each color contained in place P_i . Four-color categories represent the spatial, fidelity and security constraints and the modality of the media place, which are defined in the color set ?.

S_0 is an initial state that represents all the available colored tokens in all places at a certain marking. If a colored token is not available, it is represented as a special color $\langle H \rangle$. Security token has the form of $\langle C \rangle$ that denotes the initial credential marking, if it is not available; it is represented as $\langle CH \rangle$. Fidelity token has the form of $\langle F \rangle$ that denotes the initial fidelity marking, if it is not available; it is represented as $\langle FH \rangle$. Modality token has the form of $\langle M \rangle$ that denotes the initial fidelity marking, if it is not available; it is represented as $\langle MH \rangle$. The spatial token has the form of $\langle LER_0, DA_0, AR_0, AV_0 \rangle$, where the attributes denote the largest empty rectangles available, the component display attributes, the component starting address, and the space availability for this component respectively. In more broad terms, $MC_j, MF_j, MM_j, LER_j, DA_j, AR_j$ and AV_j denote the j^{th} state of the corresponding functions. MC, MF, MS records the marking of each place in each stage.

The dynamic behavior of the MMCTPN can be described in terms of state that represent the distribution of tokens in the net. An initial state is represented by a set of colored tokens in the source place. The verification of the cmo plan starts by firing the output transition of the initial place. The marking of the MMCTPN is changed according to the transitions firing rules.

Let MMCTPN denotes Multimedia Color Time Petri net, where $MMCTPN = (P, T, H, \Sigma, D, S_0)$ where $P = \{p_1, p_2, \dots, p_n\}$ is a finite set of places $T = \{t_1, t_2, \dots, t_n\}$ is a finite set of transitions, where $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

$T: T \rightarrow \{ST, TT, IT, UT\}$ identifies a transition is either temporal, or spatial

$SP: P \rightarrow \{MP, AP, DP, NP\}$ identifies a place is a Media Place, an Address Place, a Delay Place, or a Null place.

$H =$ There exists a set of types = {token, s-token, hole} and a type function H, such that $\forall y \in m(p), H(y) \in \{token, s-token, hole\}$

$\Sigma =$ a finite set of colors, color sets $\Sigma = \{\sigma_1, s_2, \dots\}$,

$D =$ a duration function, $D: MP \rightarrow I^$ is a mapping from media places to clock values*

$C =$ a color function such that $C(p) \subseteq \Sigma$ and $\forall y \in m(p), C(y) \in \Sigma$,

A is the arc function such that: $\forall f(p; t); f(t; p) \in F$; there is an associated set of colors Σ with each arc, where each place has the same color function.

S_0 : initial state.

3.2.1 Firing rules of MMCPN: A state of MMCTPN is defined as $S = (Sp, F, C, M)$ where, SP is the spatial marking, F is the fidelity marking and C is the security marking and M is the modality marking.

The state of MMCPN is advanced at each transition firing. The j^{th} state of MMCTPN is represented by form of $[\langle \text{LER}_k, \text{DA}_k; \text{AR}_k; \text{AV}_k \rangle, \langle \text{MF}_k; \rangle, \langle \text{MC}_k \rangle, \langle \text{MM}_k \rangle]$ that presents (S_p, F, C, M) respectively.

Temporal transition firing: Given a temporal transition T (T_i) = TT_i , $\forall P \in \bullet \text{TT}_i$ and $\forall P \in \text{TT}_i \bullet$, for any place p_i marked with $m(p_i)$, and if $H(m(p_i)) = \text{token}$, then $m(p_i)$ is said to be available if it remains in p_i for $D(p_i)$. If $H(m(p_i)) = \text{hole}$, then $m(p_i)$ is said to be available upon arrival to p_i . If $H(m(p_i)) = \text{s-token}$, then $m(p_i)$ is said to be available if it remains in p_i for $D(p_i)$.

TT_i is said to be enabled if $\forall p_i \in \bullet \text{TT}_i, m(p_i) \neq \emptyset$ and if all s-token, token, or hole in $m(p_i)$ are available. An enabled TT transition fires.

Spatial transition firing: Spatial transitions do not consume any tokens from its input place. Given a transition T (T_i) = ST_i , $\forall P \in \bullet \text{ST}_i$ and $\forall P \in \text{ST}_i \bullet$. For any place p_i marked with $m(p_i)$, and if $H(m(p_i)) = \text{s-token}$, then $m(p_i)$ is said to be available upon arrival to p_i . ST_i is said to be enabled if $p_i \in \bullet \text{ST}_i, m(p_i) \neq \emptyset$, and if any s-token in $m(p_i)$ is available. An enabled ST transition fires.

3.2.2. MMCTPN state: When transition T_i fires in the $(j-1)^{\text{th}}$, the state is advanced to the j^{th} state with functions defined below.

1. Firing a temporal transition TT_i results in a new marking M' as follows:

$$\forall \bullet P \in \bullet \text{TT}_i, \text{ and } \forall P \bullet \in \text{TT}_i \bullet,$$

$$\text{i. } m'(\bullet p) = \emptyset$$

$$\text{ii. } m'(p \bullet) = m'(\bullet p) - A(P, T) + A(T, P)$$

1. Firing a spatial transition ST_i results in a new marking M' as follows:

$$\forall \bullet P \in \bullet \text{ST}_i, \text{ and } \forall P \bullet \in \text{ST}_i \bullet,$$

$$\text{i. } m'(\bullet p) = m'(\bullet p)$$

$$\text{ii. } m'(p \bullet) = m'(\bullet p) + A(\text{ST}, P)$$

Where A is the arc function, where $A = (\text{AR}_j, \text{LER}_j, \text{AV}_j)$ that computes the list of starting address of the following component AR_j , the largest empty rectangles LER_j , and the space availability AV_j

$$\circ \text{AR}_j(P \bullet) = \text{AF}((\text{AR}_{j-1}(\bullet P), \text{DA}_{j-1}(\bullet P), \text{DA}_{j-1}(P \bullet), \text{REL}(P \bullet, \bullet P))$$

AF is an arc function. The function inputs are the previous media place start address, the previous media place display attributes, the following media place display attributes and the relation between the previous and following media places. The function

output is the starting address of the following media place.

$$\circ \text{DA}_j(P \bullet) = \text{DA}_{j-1}(P \bullet) \text{ for media place.}$$

DA is the display attributes height and width of the media place.

$\circ \text{LER}_j(P \bullet) = \text{RF}(\text{LER}_{j-1}(\bullet P), \text{AR}_j(P \bullet), \text{DA}_j(P \bullet))$
RF is an arc function. The function inputs are the largest empty rectangles available in the previous state, the display attributes of the following media place and the starting address of the following media place. The function output is the largest empty rectangle set after including the new media place.

$\circ \text{AV}_j(P \bullet) = \text{VF}(\text{LER}_{j-1}(P \bullet), \text{AR}_j(P \bullet), \text{DA}_j(P \bullet))$
 $\text{AV}_j(P \bullet)$ is an arc function. The function input are the largest empty rectangle set from before adding the new media place $\text{LER}_{j-1}(P \bullet)$, the starting address of the new media place AR_j and the height and width of the new media place DA_j . If the new media place range falls within any empty rectangle range, the function output is 1, but the media component does not fit entirely in any empty rectangle range, the function output is -1

A spatial transition ST is enabled once AV_j has a value. Available spatial transitions fires. After a spatial transition fires, all the tokens in its input place remain unchanged. The address place AP following the spatial transition ST buffers the spatial token values. The spatial tokens values includes the output of the address function AF that states the starting point of the following media place and the output of the AV Availability function that states the space availability to the following media place.

3.3 Mapping spatial constraints specification to MMCTPN.

The basic idea is to represent the fidelity and security constraints as color sets associated with the places. While colored tokens represent the credentials, characteristics and the modality and fidelity capabilities of the client (user 3Cs), colored holes represent the absence of the user 3 Cs. The time interval associated with a place represents its playback duration.

The temporal and spatial constraints among the different media components are represented in the Petri net model itself by appropriately connecting the places to the transitions. Similarly, tokens are also associated with one of the colors in the color set. As an example, if a place is associated with a color set = {medscape.com member}, and a time interval 10, which indicates the security credentials required by the component to display represented by this place, and the duration as to how long the component is to be played. This component will be triggered only if a token with color image enters this

place. On the other hand, if a hole with the color image enters, the component will not be activated, and therefore, the playback duration of the component is zero, hence will not play. This is the case of dropping a component from the cmo.

3.4 Analysis of MMCTPN:

Once the spatial constraints of the *cmo* are modeled using Petri net along with the temporal and security constraints, reachability of different places based on user 3Cs can be analyzed. With the user 3Cs as colored tokens, we perform a reachability analysis of the MMCTPN and use the resulting reachability tree to identify the places that cannot be displayed due to spatial capabilities of the user monitor.

Definition1: A marking M is covered from a marking M_0 if a sequence of firings that transforms M_0 to M exists.

In our model, the *cmo* shall be correct if all the components have enough resources to be displayed as stated at the authoring time.

Defintion2: Since the *cmo* state is equivalent to marking of its corresponding MMCTPN, Correctness of *cmo* is equivalent to the reachability property in MMCTPN.

3.4.1. Identify unreachable places due to spatial capabilities: We generate the reachability tree from the MMCTPN by using Jensen algorithm for color Petri net reachability tree algorithm [J86]. Each node in the resulting tree is a matrix with j rows representing the places of the net and i columns representing the color associated with each place. We apply the following algorithm to identify the place that cannot be displayed due to the user monitor. We search in each node in the tree for the places that has a negative value. The negative value represents unavailable space required to present the media component.

Input: R , reachability tree with n nodes.
Output: list of places that cannot be fully displayed.
 For each node R_k {If R_k is a leaf {Process node()}
 else {goto child node}; Process node ()
 for ($k = 1$ to m) //where m is the number of places
 if $R [i] [j] < 0$; add $R [i] [j]$ to print array }

3.4.2. Example: The following figure presents a parallel image (300x 300) and text (300*200) appears. The spatial transition states that the image is above the text followed

by a video (400x 400). P_{init} and P_{fin} are the initial and final places respectively. On a user monitor of 640*480, The initial Marking MS_0 , P_{init} has the initial state, Where $LER_0 = 640*480$, $AR_0 = null$, $DA_0 = \langle null, null, null \rangle$, $AV_0 = 1$.

When $T1$ fires, the Arc functions RF , AF and VF takes the token parameters. The token parameters when $T1$ fires are: $\langle LER_k, DA_k; AR_k; AV_k \rangle$ that is:

$\langle \langle 640, 480 \rangle, \langle null \rangle, \langle null \rangle, 1 \rangle$

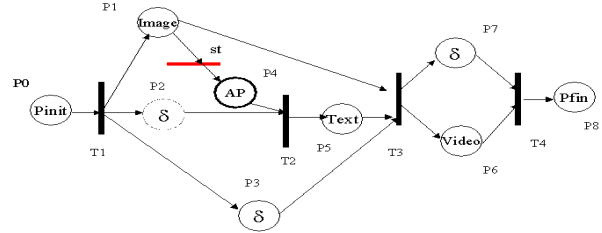


Figure2: cmo modeled in MMCTPN

The display attributes for the media place are inherited from the place previous state where $DA_j (P\bullet) = DA_{j-1} (P\bullet)$, hence $DA = \langle 300, 300, 1 \rangle$.

- AF function calculates the starting address of the media place where $AR_j (P\bullet) = AF ((AR_{j-1} (\bullet P), DA_{j-1} (\bullet P), DA_{j-1} (P\bullet), REL (P\bullet, \bullet P))$ thus $AR_1 = \langle 0, 0 \rangle$. The starting address of the delay place remains as the previous place $\langle null \rangle$
- RF function calculates the largest empty rectangles where $LER_j (P\bullet) = RF (LER_{j-1} (\bullet P), AR_j (P\bullet), DA_j (P\bullet))$. $LER_1 = \langle R1, R2 \rangle$ where $R1$ and $R2$ are the largest empty rectangles after adding the image. The delay place display attributes are set to null. $R1$ is $\langle 640*180 \rangle$ and $R2 = \langle 340*480 \rangle$
- VF function checks if the image $AV_j (P\bullet) = VF (LER_{j-1} (P\bullet), AR_j (P\bullet), DA_j (P\bullet))$, thus $AV_1 (P\bullet) = 1$. The delay place display attributes are set to null
 The Marking for the image becomes
 $\langle \langle R_1, R_2 \rangle, \langle 300, 300, 1 \rangle, \langle 0, 0 \rangle, 1 \rangle$
 The Marking for the dummy place becomes
 $\langle \langle null \rangle, \langle null \rangle, \langle null \rangle, null \rangle$

When ST fires, the Arc functions transfer the parameters to the address place giving a marking of $\langle \langle R_1, R_2 \rangle, \langle 300, 300, 1 \rangle, \langle 0, 0 \rangle, 1 \rangle$ and the spatial relation "ABOVE"

When $T2$ fires, the Arc functions RF , AF and VF takes the token parameters from the address place. The display attributes for the media place are inherited from the place previous state where $DA_j (P\bullet) = DA_{j-1} (P\bullet)$, hence $DA = \langle 300, 200, 1 \rangle$.

- AF function calculates the starting address of the media place where $AR_j (P\bullet) = AF ((AR_{j-1} (\bullet P), DA_{j-1} (\bullet P), DA_{j-1} (P\bullet), REL (P\bullet, \bullet P))$, thus $AR_2 = \langle 300, 0 \rangle$.
- RF function calculates the largest empty rectangles where $LER_j (P\bullet) = RF (LER_{j-1} (\bullet P), AR_j (P\bullet), DA_j (P\bullet))$

(P•)). $LER_1 = \langle R_2 \rangle$ where R_2 is the largest empty rectangle after adding the text.

- VF function checks if the text $AV_j(P•) = VF(LER_{j-1}(P•), AR_j(P•), DA_j(P•))$, thus $AV_1(P•) = -1$.

The Marking for the text becomes

$\langle \langle R_1, R_2 \rangle, \langle 300, 200, 1 \rangle, \langle 300, 0 \rangle, -1 \rangle$ While if the user monitor suited the cmo constraints, it would be $\langle \langle R_1, R_2 \rangle, \langle 300, 200, 1 \rangle, \langle 300, 0 \rangle, 1 \rangle$.

3.5. Logical units and reconstruction of constraints

After constructing the reachability tree of the Petri net and identifying the media places that cannot fit the user monitor, we reconstruct the constraints to allow the cmo components to be presented on the user monitor. We do not reconstruct the mandatory constraints. If a mandatory constraint cannot be satisfied, the component associated with that constraint is removed from the cmo plan. We only reconstruct the optional temporal and spatial cmo constraints to match the user monitor resources.

The logical unit LU_i is a distinct set of components that have a spatial marking at the same time t . In other words, the logical unit LU_i presents all the components that appear at a certain time on the user monitor. We already have calculated the exact coordinates of each component on the user monitor while generating the reachability tree.

We first check all the components in the logical unit that include any unreachable components (R_1, R_2, \dots) and check for the components, with higher priority first if they fit in the available space in the user monitor where $R_i \subseteq LER_0$.

If true, we drop the spatial constraints and insert R in available LER_0 . For the components that cannot be inserted in the same user monitor, We make sure that the unreachable component can be reached by adding a new logical unit LU_{i+1} for such component such that R_i size \leq monitor size, then generate new temporal constraints for R_i and its related components after dropping the spatial and temporal constraints related to R_i .

We use subnets to generate logical units. A subnet is the part of the net between two subsequent transitions including these two transitions where all places connected with an output arc to the first transition are also connected with an input arc to the subsequent transitions.

If a component exists in a subnet, where the subnet includes more than two transitions and the component is connected to the first and last transitions in the subnet, such component we call a master component. While other components, we call slave components.

3.5.1. Static Components reconstruction: To reconstruct the cmo to enable the adaptation of the spatial constraints, if the master is static, we first remove the output arcs and input arcs to the master component. Second, we divide the components into number of subnets such that the total component duration matches the resulting subnets durations. Third, we connect each component with its related subnet via output and input arcs. We substitute the spatial coordinates of each component in the subnet with the spatial relationships. We focus on the temporal constraints affected by the newly added logical unit.

For each LU_i we remove the unreachable component along with its temporal relations and insert it in a new LU_{i+1} . LU_{i+1} is inserted immediately after LU_i . We remove the unreachable components from LU_i until all components in LU_i are \leq the monitor size LER_0 .

To generate constraints from the logical units that includes the unreachable component do:

For each logical unit that included an unreachable component:

- 1- Delete parallel temporal constraints and spatial constraints associated with the unreachable component
- 2- Initialize new sequential temporal constraints between all components in LU_i and R_i
- 3- Delete all sequential constraints to LU_i components and create new sequential constraints to LU_{i+1} instead. The cmo plan constraints becomes:
 - a. Spatial: Absolute address that fit user monitor.
 - b. Temporal: Parallel relations between all components in a logical unit LU_k
 - c. Sequential relation between sequential components in sequential logical units.
- 4- Send the cmo spatial and temporal constraints to the client in SMIL format. To translate the constraints into SMIL, we use the approach presented in [AAGA03]

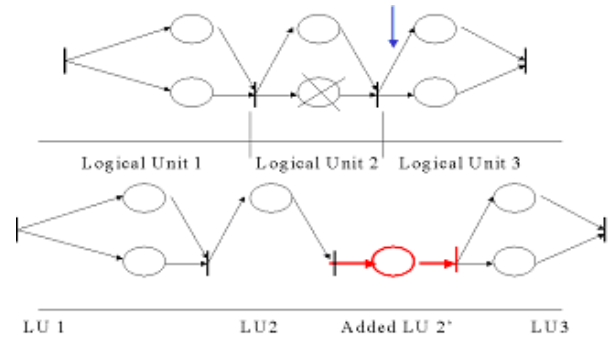


Figure3: Adapting temporal constraints

We postpone sequential relations after LU_{i+1} because we simply divide one logical unit to 2 sequential logical

units. Parallel relations are already taken care of by having subnets. The new cmo plan will be conflict free since the unreachable states are taken care of by allowing more resources to make them reachable.

3.5.2 Continuous Components reconstruction: If the master in the subnet is continuous, i.e., video, we drop the video from the cmo. The reason we drop the video is due to the semantic of the video frames as opposed to the static images in a subnet. If cmo authors present semantically related text to the video frames displayed then when dealing with logical units, one must put the frame semantic into account. We understand that dropping the video to make other components reachable is not appealing, but we plan to address it more properly in our future work that address the change of modality in a cmo to accommodate the user 3Cs. We plan to use video summarization techniques to resize the video and to summarize frames in the other logical units in the subnet. The Timed Text Working Group of W3C and MPEG21 standard are currently addressing this issue.

4 Related work

Multimedia personalization using Petri net has been introduced by [LG90]. [AAAB01] presented a Petri net model for adapted multimedia presentations where they consider dropping the components without using the analysis power of Petri net. [AAGA03] presented an extension to [AAAB01] by adding the security constraints to the model. [PYY03] presents the temporal and spatial relations of a multimedia object as an extension of [LG90]. In contrast to our work, we present all the constraints (fidelity, security, temporal and spatial), in addition we are concerned with the cmo adaptation not only presenting the spatial and temporal constraints. MPEG 4, an encoding multimedia standard, allows object adaptation without considering the spatial constraints between the different audiovisual components in the scene.

5 conclusion and future work

In this paper, we consider the constraints that govern the cmo behavior. If the user device does not allow accommodating all the constraints, either we drop the components related to the constraints that cannot be satisfied or we remove the constraints that cannot be satisfied based on the cmo author specification. However, we do not change the components properties such as size or the components modalities such as converting them to audio or video as we plan to address the modality of the cmo in our future work. More specifically, we present a

new Color time Petri net model that maps the user monitor information along the user credentials, fidelity capabilities and available modality drivers to a composite multimedia object. The model enables the following. 1) The cmo author views all the constraints in one model. 2) Model analysis to identify mismatches between the constraints and the user available resources. Based on the identified mismatches the paper presents algorithms to restructure the constraints to make the cmo adaptable to user devices.

For future research, we will introduce techniques to assure consistency of user interaction into the adaptable cmo, in addition we plan to present techniques to allow changes of the modalities of component to fit the user devices.

References

- [A84] J.F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 1984:23(2):123–154.
- [AAAB01] N. R. Adam, V. Atluri, I. Adiwijaya, and S. Banerjee. A Dynamic Manifestation Approach for Providing Universal Access to Digital Library Objects. *IEEE TKDE*, 2001:13(4).
- [AAGA03] Vijayalakshmi Atluri and Nabil Adam and Ahmed Gomaa and Igg Adiwijaya, Self-manifestation of composite multimedia objects to satisfy security constraints, *Proceedings of the 2003 ACM SAC*, 2003: PP 927-934.
- [BS00] Bell, B., and Feiner, S. Dynamic Space Management for User Interfaces. *ACM Proc. UIST '00*, San Diego, 2000, p.239-248.
- [H71] C.L. Hamblin. *Instants and Intervals*. Stadium Generale, 27:127–134, 1971.
- [LG90] T.D.C. Little and A. Ghafoor. Synchronization and Storage Models for Multimedia Objects. *IEEE J. Selected Areas in Communication* 1990:8(3).
- [LL04] Tayeb Lemlouma, Nabil Layaïda. Context-Aware Adaptation for Mobile Devices. *IEEE International Conference on Mobile Data Management*, Berkeley, CA, USA, 2004:pp. 106-111
- [PYY03] H. Ping-Yu; C.Yuan-Bin; C. Yen-Liang, STRPN: A Petri-Net Approach for Modeling Spatial-Temporal Relations between Moving Multimedia Objects. *IEEE TSE*, 2003: (29), (1), pp 63-76,
- [S88] Yoav Shoham. *Reasoning about Change, Time and Causation from the Standpoint of Artificial Intelligence*. The MIT Press, Cambridge, MA, 1988.
- [WQG94] Woo, R., Qazi, N. U., & Ghafoor, A. A synchronization framework for communication of pre-orchestrated multimedia information. *IEEE Network*, 1994:8(1).
- [RH92] David, R. and Alla, H. *Petri Nets and Grafset: Tools for Modeling Discrete Events Systems*. Prentice-Hall International: UK, 1992.
- [M89] T. Murata. "Petri nets: Properties, analysis and applications." *Proceedings of the IEEE*, 1989:77(4):451-480,
- [J86] Kurt Jensen: *Coloured Petri Nets*. *Advances in Petri Nets* 1986: 248-299