A Model for Secure Multimedia Document Database System in a Distributed Environment

James B. D. Joshi, *Student Member, IEEE*, Zhaohui Kevin Li, Husni Fahmi, Basit Shafiq, and Arif Ghafoor, *Fellow, IEEE*

***Abstract*—The Internet provides a universal platform for large-scale distribution of information and supports inter-organizational services, system integration, and collaboration. Use of multimedia documents for dissemination and sharing of massive amounts of information is becoming a common practice for Internet-based applications and enterprises. With the rapid proliferation of multimedia data management technologies over the Internet, there is growing concern about security and privacy of information. Composing multimedia documents in a distributed heterogeneous environment involves integrating media objects from multiple security domains that may employ different access control policies for media objects. In this paper, we present a security model for distributed document management system that allows creation, storage, indexing, and presentation of secure multimedia documents. The model is based on a time augmented Petri-net and provides a flexible, multilevel access control mechanism that allows clearance-based access to different levels of information in a document. In addition, the model provides detailed multimedia synchronization requirements including deterministic and nondeterministic temporal relations and incomplete timing information among media objects.**

***Index Terms*—Access control, content-based, multilevel security (MLS), multimedia databases, synchronization.**

I. Introduction

RECENT advances in high-performance computing and networking technologies have resulted in a tremendous growth of distributed multimedia applications in medicine, education, e-commerce, digital libraries, digital government, and many others. Most of these applications require generation, representation, processing, storage, and dissemination of information composed of multiple media types such as text, image, audio, and video [35]. Multimedia technologies are expected to significantly increase productivity, effectiveness, and usability of information systems [9], and can provide an effective medium for computer-to-human and human-to-human communications [33]. Use of multimedia documents over the Internet is becoming a major source of information dissemination and sharing. Internet provides a global platform for distribution of information for large-scale enterprises. Synchronization, storage, and retrieval of multimedia documents have been a focus of research for the past several years. With the growing openness of the Internet, security of multimedia information systems (MMIS) has become a central issue in developing large-scale distributed multimedia applications.

A multimedia document can be composed of various multimedia objects that may be stored in a central archive or distributed over various heterogeneous database servers interconnected by a broad-band network, as depicted in Fig. 1 [40]. Access to distributed servers may be governed by different security policies. The scope of a security policy (SP) over a set of subjects and objects constitutes a security domain. The security policies may vary within each system and across systems, making the integrated MMIS a complex multidomain environment that raises several crucial security challenge [12], [17], [21], [36]. There is a growing need for ensuring secure access to individual media components in distributed multimedia applications. For example, in a Web-based distance learning application, users may have restricted access to various components of a multimedia document. Similarly, in the area of healthcare, medical records of patients may contain personal medical history and radiological data, which need to be protected against unauthorized access. However, in this case, different types of patient information should be easily accessible to various authorized users. For example, a physician must be allowed to access relevant details of a patient's medical history, whereas, a medical insurance personnel should only be allowed to access the information related to the patient's medical expenses, certified causes of illness, etc. The problem of access control in such applications becomes more challenging by the fact that the delivery of multimedia documents requires strict spatio-temporal synchronization of media objects with guaranteed quality of service. An access control mechanism, for instance, should not pose restrictions in fulfilling such requirements. Several security approaches have been used to address the issues related to information systems security. These include *discretionary access control (DAC)*, *mandatory access control (MAC)*, and *role-based access control (RBAC)* approaches [22], [25], [37].

In this paper, we propose a unified multimedia document model to allow secure access to a multimedia document database. The model can use an MAC approach, known as *multilevel security (MLS)* [22],[25]. MLS approach is particularly suited for multimedia applications because it allows access control based on multiple classifications that may be imposed on media objects. The proposed model provides an application-level, user-defined classification schemes on multimedia documents. The model allows the coexistence of multiple secu-

Manuscript received April 7, 2001; revised February 26, 2002. This work was supported in part by grants from the Center for Education and Research in Information Assurance and Security and the National Science Foundation (NSF: 9 972 883). The Associate Editor coordinating the review of this paper and approving it for publication was Dr. Ahmed Tewfik.

The authors are with the Distributed Multimedia Systems Laboratory, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285 USA (e-mail: joshij@purdue.edu; kevinli@abeona.com; fahmi@ecn.purdue.edu; shafiq@purdue.edu).

Publisher Item Identifier S 1520-9210(02)04862-9.

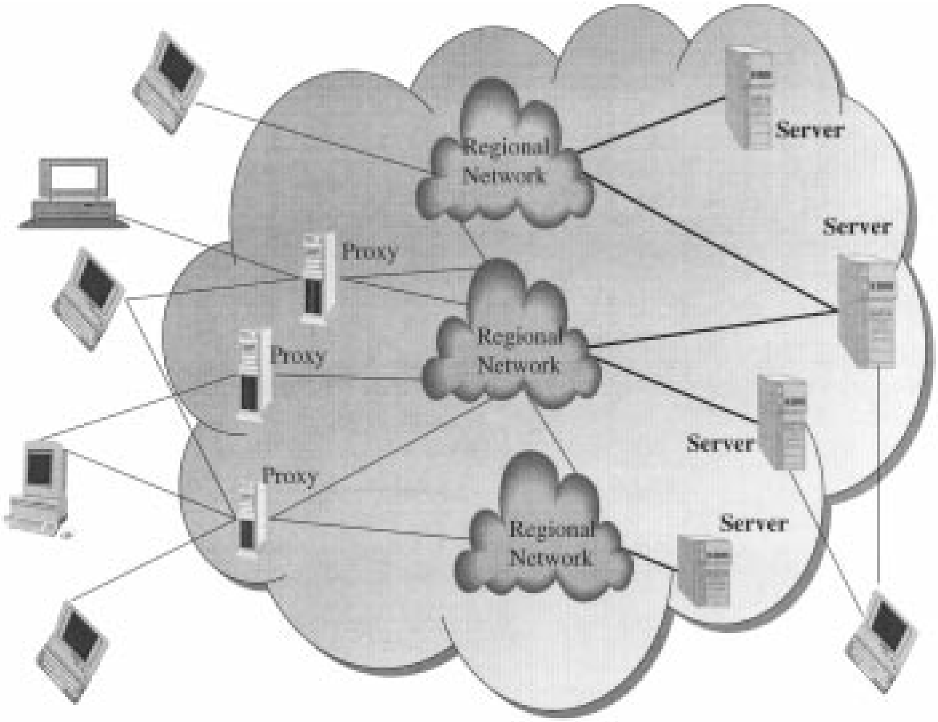


Fig. 1. Distributed, heterogeneous, multimedia database system.

rity domains and hierarchical classification policies, thus supporting specification of complex documents with multilevel objects maintained by the underlying heterogeneous databases. The model constitutes a significant extension to the existing *object-composition Petri-net model (OCPN)* [29], [30] by integrating security requirements for multimedia documents. In addition, the proposed model facilitates the specification of nondeterministic temporal relationships and incomplete temporal information for media objects. It allows asymmetric relationships to exist among distributed objects of a document. Provision for nondeterministic temporal relationships and incomplete temporal information is an important issue in a distributed multimedia environment with random network delays. Based on the proposed model, a system architecture is presented that supports authoring, querying, storage, and content-based retrieval of distributed multimedia documents in a secure environment.

In literature, few researchers have addressed the issue of access control for multimedia documents [13], [14]. Although there is a growing realization to address this issue, the existing work either presents a preliminary discussion on the requirements for security in multimedia systems [13], or focuses on authorization models for mono-medium databases [4]. As mentioned earlier, the model proposed in this paper provides an integrated framework to address both the application level access control needs and media synchronization requirements for composing distributed multimedia documents. Existing multimedia synchronization models such as OCPN [29], XML [41], and time-flow graph [26], [27] address only deterministic temporal relationships where durations among objects are known *a priori*.

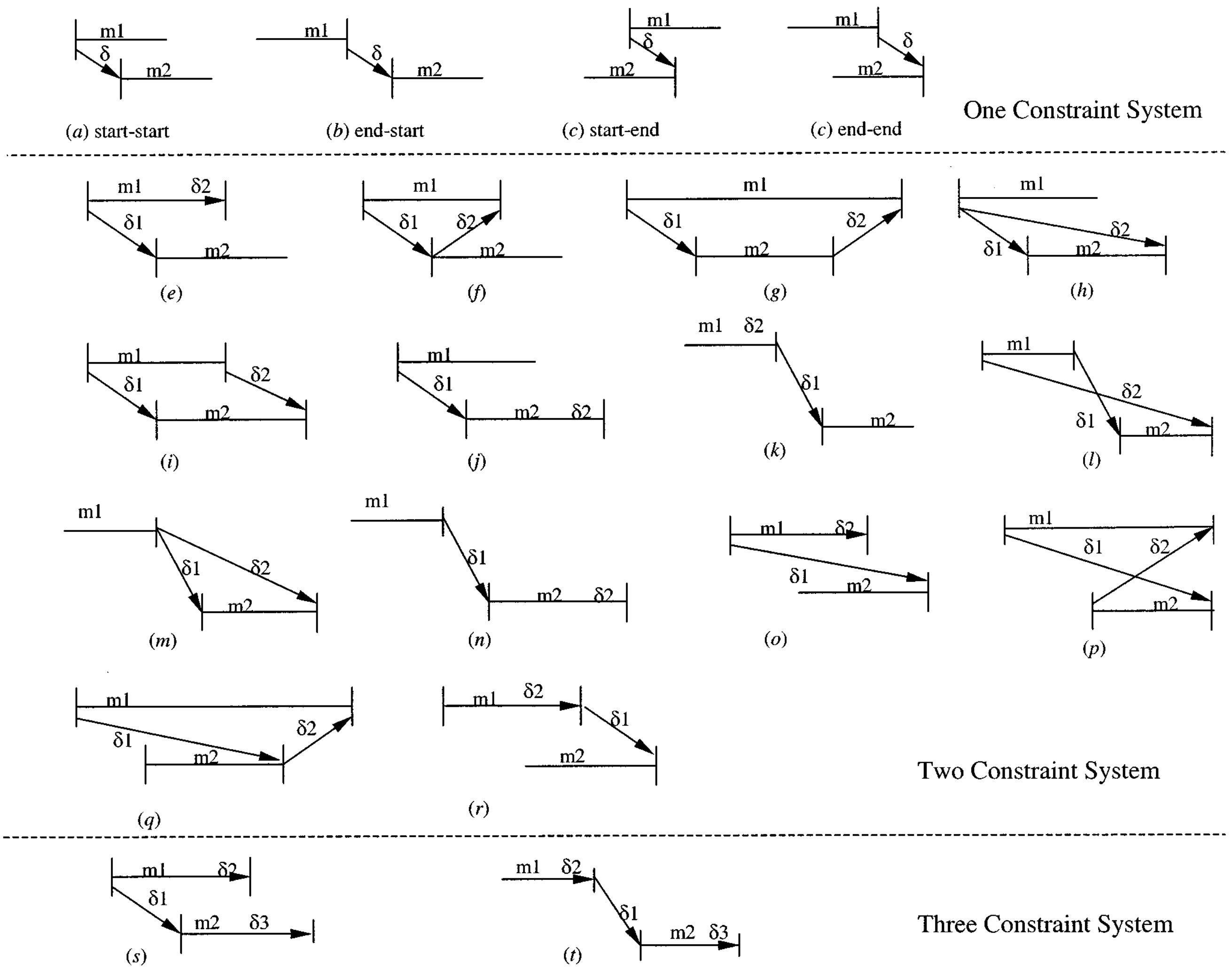
The paper is organized as follows. In Section II, we present the generalized OCPN (GOCPN) model for specifying deterministic and nondeterministic temporal relations. In Section III, we present a *secure*-GOCPN model, which uses GOCPN as the base model, and show how the extended model allows MLS for multimedia document databases. In Section IV, we present an architecture of a secure distributed multimedia document database system. In Section V, we present related work. Section VI provides a conclusion of this paper.

II. Multimedia Document Model for Synchronization and Indexing

Temporal modeling is a requirement for ensuring synchronization of distributed media objects to compose multimedia documents. In this section, we first present a temporal synchronization scheme called *temporal constraint set (TCS)* to enumerate all the possible temporal relationships that may exist between any two media streams represented as temporal intervals. TCS describes the binary temporal relations between any two intervals and captures both deterministic and nondeterministic temporal relations.

A. Temporal Constraint Set (TCS)

Any two time-intervals with deterministic duration can be synchronized in 13 different ways [1]. For interactive multimedia, duration of media objects may be unknown or unpredictable prior to presentation of documents to the end users. To represent temporal relations among the intervals with unknown durations, the proposed TCS is based on time instants and intervals. A time instant is a point on the time axis that can be represented by a nonnegative number. An interval starts at a time instant and ends at another time instant as defined below.



(*a*) start-start (*b*) end-start (*c*) start-end (*c*) end-end

One Constraint System

(*e*) (*f*) (*g*) (*h*) (*i*) (*j*) (*k*) (*l*) (*m*) (*n*) (*o*) (*p*) (*q*) (*r*)

Two Constraint System

(*s*) (*t*)

Three Constraint System

Fig. 2. Temporal constraint set.

Definition 1 (Interval) [30]: Let $[R, =]$ be a partially ordered set, and let a, b be any two elements of R such that $a = b$. The set $\{x \mid a = x = b\}$ is called an interval of R denoted by $[a, b]$.

Let a and b be the *start* and *end*-points of an interval and $b - a$ its duration. End points of an event are nondeterministic when they are influenced by an occurrence of another event or are unknown beforehand. The former case represents a causal relation. The end points of an interval can perform an action on an end point of another interval. An action aimed at the beginning of an interval corresponds to *start* or *activation*, while an action aimed to end an interval corresponds to a stop or termination. Such an action is represented in a graph as a constraint arrow from one end point to another end point, as depicted in Fig. 2. For any two intervals $m_1(a_1, b_1)$ and $m_2(a_2, b_2)$, there can be at most four possible constraints between the end points b_1 and b_2. Fig. 2 shows the four relations. For a causal system, the delay is always nonnegative.

In *start-start* relation of Fig. 2(a), the start point of interval m_2 occurs after the start point of interval m_1 with a delay of δ time units. In this case, the constraint between the start points a_1 and a_2 of m_1 and m_2, respectively, is given by $a_2 - a_1 = \delta$. The end points of these two intervals are set to be free and remain unspecified. Similarly, the cases (*b*) *end-start*, (*c*) *start-end*, and (*d*) *end-end* represent constraints between two intervals for different pairs of end points. These constraints are $a_2 - b_1 = \delta, b_2 - a_1 = \delta$ and $b_2 - b_1 = \delta$, respectively. Since the delay δ is nonnegative, the arrow for specifying a delay does not point backward. The inverse relation is obtained by simply switching the intervals m_1 and m_2.

By adding one more constraint to the one constraint system in Fig. 2, we can define eight double-constraints. Two such relations are *over-constrained* where both *start* and *end* points of m_1 have a constraint on the same end point of m_2 and can be either the *start* or the *end* of m_2. Therefore, six out of the eight relations are valid double constraints. These include cases (*e*), (*f*), (*g*), (*h*), (*i*), and (*j*) in Fig. 2. Similarly, we can enumerate all other binary constraints. Fig. 2 illustrates all the 14 possible temporal relationships for the two-constraint system. We have excluded over-constraint relations.

The 14 constraints of two constraint systems are not mutually independent. For example, constraint (*e*) can represent temporal information contained in (*f*). In other words, constraint (*e*) is at least as powerful as (*f*) when we consider the event ordering of the two intervals. If $\delta 2$ in (*e*) is specified to be the sum of $\delta 1$ and $\delta 2$ in (*f*), the end point position of interval m_1 is determined to be the same instant in the time axis for both (*e*) and (*f*).

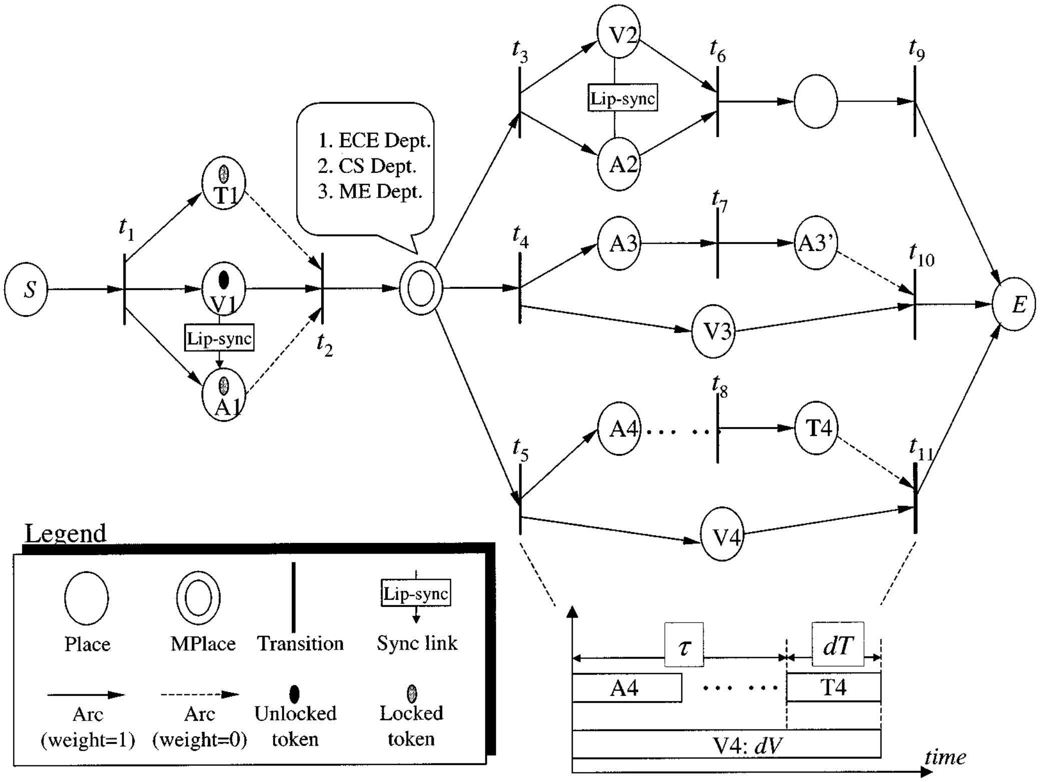


Fig. 3. Recursive firing algorithm.

However, (*f*) cannot represent (*e*) if $\delta 2$ in (*e*) is smaller than $\delta 1$ in (*f*). Similarly, constraint (*j*), (*n*), and (*o*) have more temporal descriptive power than (*h*), (*m*), and (*q*), respectively. Consequently, the number of temporal relations is reduced to nine by dropping (*f*), (*h*), (*m*), and (*q*).

The three-constraint set can be generated from the two-constraint set. As most three constraints introduce conflicts, only two temporal relations, (*s*) and (*t*), shown in Fig. 2 are meaningful. The final TCS is the set of the following 16 temporal relations depicted in Fig. 2:

$$\{(a),(b),(c),(d),(e),(g),(i),(j),(k),(l),(n),(o),\\ (p),(r),(s),(t)\}.$$

B. The Generalized Object Composition Petri-Net (GOCPN) Model

The proposed TCS provides a powerful and flexible model that unifies time instants and intervals through constraints. It can describe Allen's 13 temporal relations and overcome its limitation by capturing nondeterministic relations. The model can be used to specify arbitrary temporal requirement for composing and presenting multimedia documents. For this purpose, we introduce the GOCPN model that augments the Petri-net structure and the execution rules to represent multimedia synchronization. Based on this model, one can precisely describe temporal and spatial information, user interaction, browsing semantics, media contents, and quality of presentation (QoP) of component media involved in a multimedia document. This is achieved by associating time duration modeled by TCS constraints to places in GOCPN, assigning weights to arcs, specifying transition types, and modifying firing rules for the GOCPN. Formally, a GOCPN model is defined as follows.

Definition 2 (GOCPN): A GOCPN is a 7-tuple $G = \{P, T, A, \mathrm{AW}, \mathrm{PT}, \mathrm{SL}, \mathrm{ObjA}\}$, where

1) (P, T, A) is a Petri-net; (i) P is a finite set of places, (ii) T is a finite set of transitions such that $P \cap T = \emptyset$, and (iii) $A \subseteq \{P \times T\} \cup \{T \times P\}$ is a set of arcs;
2) AW: $A \to B, B = \{0, 1\}$ is a weight function of arcs;
3) PT: $P \to \{\text{regular, decision}\}$ identifies a place as either regular or decision place (regular place has only one output transition whereas a decision place has multiple output transitions);
4) ObjA: $P \to \{C \times Q \times D \times S \times \mathrm{Op}\}$ is a mapping of places to the content set C, QoP set Q, integer set D representing durations, spatial information S, and an ordered set of media operations Op (if $\mathrm{Op} = \{\mathrm{op}_n, \mathrm{op}_{n-1}, \ldots, \mathrm{op}_2, \mathrm{op}_1\}$ then operation op_i is applied on the media content before operation op_{i+1});
5) SL: $\{P \times P\} \to I$ represents *lip-sync* link between two places, where I is an integer set representing maximum skew allowed between two media objects measured by discrete time units.

The dynamic behavior of GOCPN can be described in terms of markings that represent the distribution of tokens in the net. An *initial state* is represented by a token in the *source place*. The presentation of a multimedia document starts by firing the output transition of the *source place*. The marking of a GOCPN is changed according to the following transition firing rules.

1) A transition t is enabled if each of its active input place p is marked with at least $\mathrm{AW}(p, t)$ unlocked tokens, where $\mathrm{AW}(p, t)$ is the weight of the arc from p to t.

```
Algorithm Recursive-Firing($t_i$)
  T = ∅;
  For each $p_j \in P_{in}(t_i)$ no tokens
      Insert $t_{in}(p_j)$ into transition set T
  For each transition $t_k \in$ T
      Firing ($t_k$)
  Remove a token from each $p_j \in P_{in}(t_i)$
  Add a token to each place $p_l \in P_{out}(t_i)$

/* $P_{in}(t_i)$ represents the input places of $t_i$ */
```

Fig. 4. GOCPN representation of TCS.

2) Firing of an enabled transition t takes place as follows: (a) if t contains an input arc with $\mathrm{AW} = 0$, then the recursive firing algorithm shown in Fig. 3 is used; (b) if the input place of t is of *regular* type, t fires immediately; (c) if its input place is of *decision* type, t can fire only when triggered by an input. The algorithm of Fig. 3 recursively fires every input transition of those places that appear before the enabled transition. The algorithm strictly follows the temporal ordering in GOCPN in the sense that a transition is fired only if all the transitions ahead of it in the GOCPN have already been fired.
3) Firing of a transition results in removing a token from each of its input places, and placing a token in each of its output places.
4) Upon receiving a token, a place starts the process of displaying the associated media object. We assume that the QoP requirements and other specification regarding duration, spatial and synchronization constraint are managed by the underlying system architecture. An active process locks the token. The token is unlocked whenever either of the following events happens: (i) the specified duration d of the place expires, or (ii) if the duration is unspecified, the token is unlocked by the process.
5) If place p_1 has a *lip_sync* link to another place p_2, synchronization between the presentation processes in those two places is enforced by ensuring that the skew between the two media streams is less than the specified value of $\mathrm{SL}(p_1, p_2)$.
6) The contents element of an *Mplace* provides description of information about all the selectable paths. Each description is associated with a logical button that can be mapped to some area of the screen. Each logical button in turn is associated with an output transition of the *Mplace*. When a path is selected, the corresponding transition is triggered to fire.

Fig. 4 shows the GOCPN representation of all the constraints of TCS.

Fig. 5 shows an example of a GOCPN document. The number of tokens deposited at a given place p is determined by the weight of the arc leading to p. An arc with weight 1 (i.e., $\mathrm{AW} = 1$), drawn as a solid line, is *active*. The set of active input places for transition t is denoted by $P_{\text{active}}(t) = \{p \,|\, (p,t) \in A \text{ and } \mathrm{AW}(p,t) = 1\}$. Similarly, the set of inactive input places for transition t is denoted by $P_{\text{inactive}}(t) = \{p|(p,t) \in A \text{ and } \mathrm{AW}(p,t) = 0\}$.

In GOCPN, a place may have multiple output transitions that represent multiple browsing paths in a document. Such a place represents a *decision* place where a user can choose one of its output transitions to fire. Such a place is called *Mplace* and is denoted as a double circle in the GOCPN model, as shown in Fig. 5. The function PT is used to differentiate such a place from a *regular* place. When a token arrives at an *Mplace*, a text is mapped to the screen to give brief information about the contents of each path, for which a button or selectable item is displayed. In Fig. 5, transitions t_3 and t_4 are related to "*ECE Dept.*" and "*CS Dept.*," respectively. The user can choose a path to proceed with the presentation process.

QoP parameters are supplied by the user in the form of perceived quality for the presentation of a document. Generally, each media type has its own unique QoP requirements. For instance, text may allow some delays but it should be free from error or missing information. Similarly, in most of the applications, video is sensitive to delay but may tolerate dropping of some frames. In some applications, a finer level of synchronization of two related media streams, such as *lip-sync* between video and audio, must be enforced in order to provide a meaningful presentation. In GOCPN, *lip-sync* links are introduced to specify the maximum tolerable skew between two media streams. The direction of a synchronization link reflects the asymmetric relation in inter-media synchronization. For example, the *lip-sync* direction from V1 (video) to A1 (audio), in Fig. 5 implies that A1 should catch up with V1 during presentation. In this case, more importance is attached to object V1. In an asymmetric relation between two objects, one object is identified as the master object and the other as the slave object. For example, in Fig. 5, V1 has the major content for the presentation and can be considered as the master object, whereas, A1 and T1 can be considered as slave objects. A1 and T1 end as soon as V1 ends upon firing of t_2. In some cases, it may be necessary to leave the duration of an object unspecified. For example, as shown in Fig. 5, the text object T4 may need to be displayed when video V4 is being presented after A4 has been played out. In a deterministic model, duration dT associated with T4 needs to be determined based on τ and dV. However, finding τ may be difficult in case there is a complex substructure between T4 and A4. GOCPN allows such incomplete timing information between A4 and T4 by allowing inactive arc as shown in Fig. 5.

III. Extended GOCPN for Specifying Access Control in Multimedia Document Database

Media contents in a multimedia document may have various levels of sensititvity or classification that depend upon the nature and/or the quality of information they contain. For example, a video sequence depicting the use of firearms may need to be classified as "*adult viewing only*" to prevent children from learning how to use firearms. Similarly, a person who pays a higher rate of subscription may be allowed to view more expensive information with detailed charts and diagrams of up-to-the-minute stock market analysis. Such content-based classification may also depend on the context of information. For example, a publicly available news clip may be classified as top-secret when it is augmented with some annotation that identifies a person in the clip as a secret agent. In following, we introduce *secure*-GOCPN, which augments the GOCPN model to allow such content-based classifications on subjects and objects so

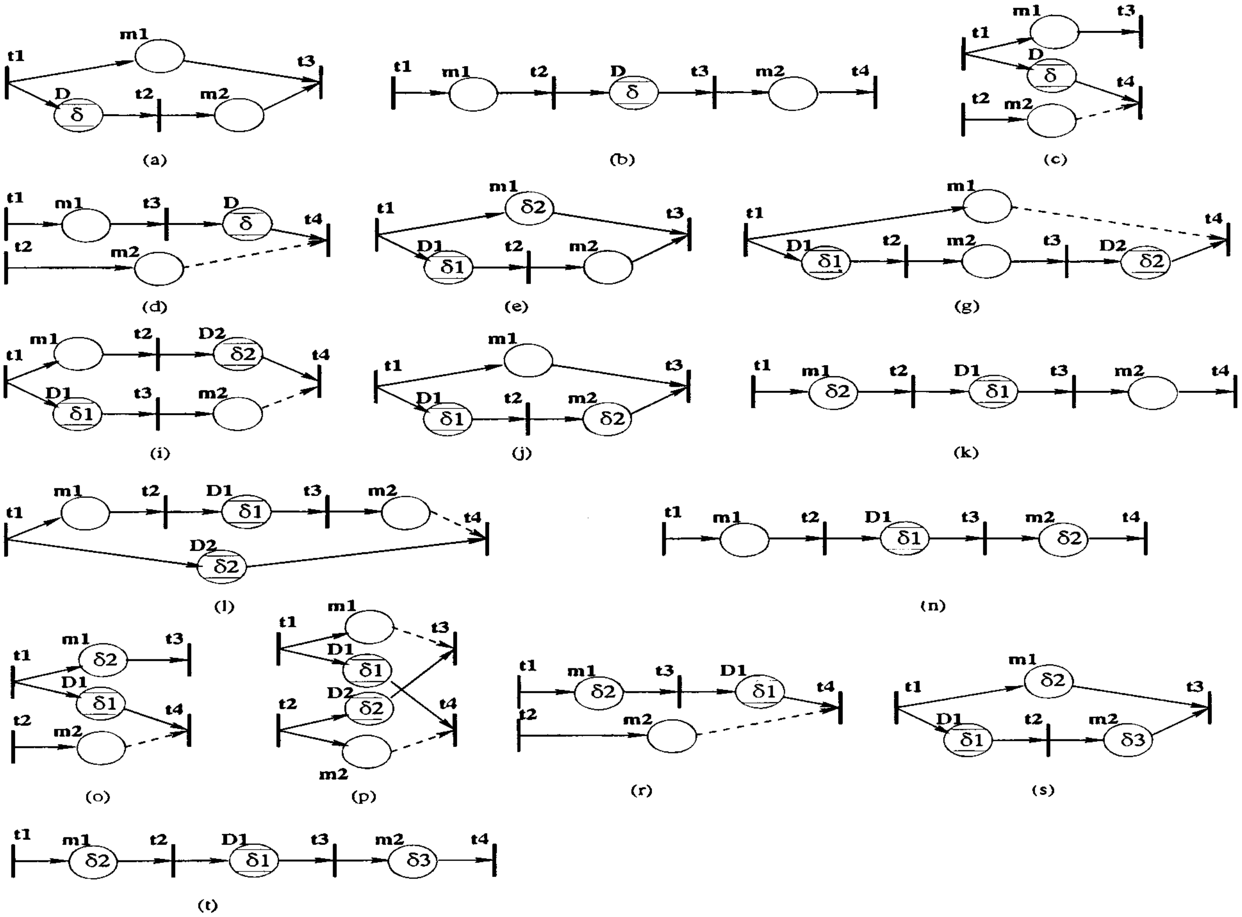


Fig. 5. GOCPN document.

that appropriate class of media objects within a document can be selected for presentation to individual users.

A. Multilevel Security (MLS)

MLS has been proven to be a practical model for many military and commercial applications where some form of natural hierarchy or compartmentalization exists [25]. Distribution of information in such organizations is controlled by some measure of *sensitivity* or on a *need-to-know* basis. The *sensitivity* of information is determined by its content, context, aggregation, or time [2]. To ensure secure access, users should be allowed to access only those information for which they have the required clearances. In an MLS model, a system consists of a set S of subjects, also known as *active entity*, a set O of objects, also called *passive entities*, and a set L of security levels with a partial ordering relation. A level l_i is said to be *dominated* by l_j if $l_i \leq l_j$, and *strictly dominated* by l_j if $l_i \leq l_j$ and $l_i \neq l_j$. If neither $l_i \leq l_j$ nor $l_i \geq l_j$, then l_i and l_j are said to be *incomparable*. In an MLS scheme, these security labels are used as object classification as well as subject clearance levels that are assigned by the administrator. The *dominated* relation between the classification level of an object and the clearance level of a subject is used to determine whether the subject is to be allowed to access the object. The well-known Bell-LaPadula (BLP) [3] restrictions for an MLS system are as follows.

1) *Simple security property*: a subject s is allowed to read an object o iff $l_s \geq l_o$ (*no-read-up property*).

2) $*$*property*: a subject s is allowed to write an object o iff $l_o \geq l_s$ (*no-write-down property*).

In this paper, we focus on the issue of the availability of the presentation documents to the authorized users. For simplifying classification mechanism, we use the notion of views as used in [2] and [7]. We next illustrate such a mechanism with the multiview model of MLS proposed in [2], [8], and [11].

1) Multiview Model of MLS: The multiview model decomposes a database that uses a n-level classification scheme into n views. A view is maintained for a given level of classification and provides access to all the data that has classification level less than or equal to that of the view. The assignment of classification levels in the multiview model is done at the object level (instances). Fig. 6 illustrates an example of MLS. Here, we assume each object is classified as either classified (C) or unclassified (U).

A creator A with a clearance for *unclassified* level creates an object O_1 and assigns to attribute `Name` the classification U and value "Ralph," and to attributes `Age` and `Country` the classification C. O_1 is given the sensitivity level U. As `Country` is classified at C, he may put the value "Canada" as a "*cover story*" for it. In the *classified* database (DB), pointers to the corresponding object state in *unclassified* DB are created as shown by an arrow in Fig. 6(b) from an object instance in *classified* DB to the object instance in the *unclassified* DB. For an attribute classified as C, the actual value is copied. After the creation, the *classified* DB and *unclassified* DB are as shown in

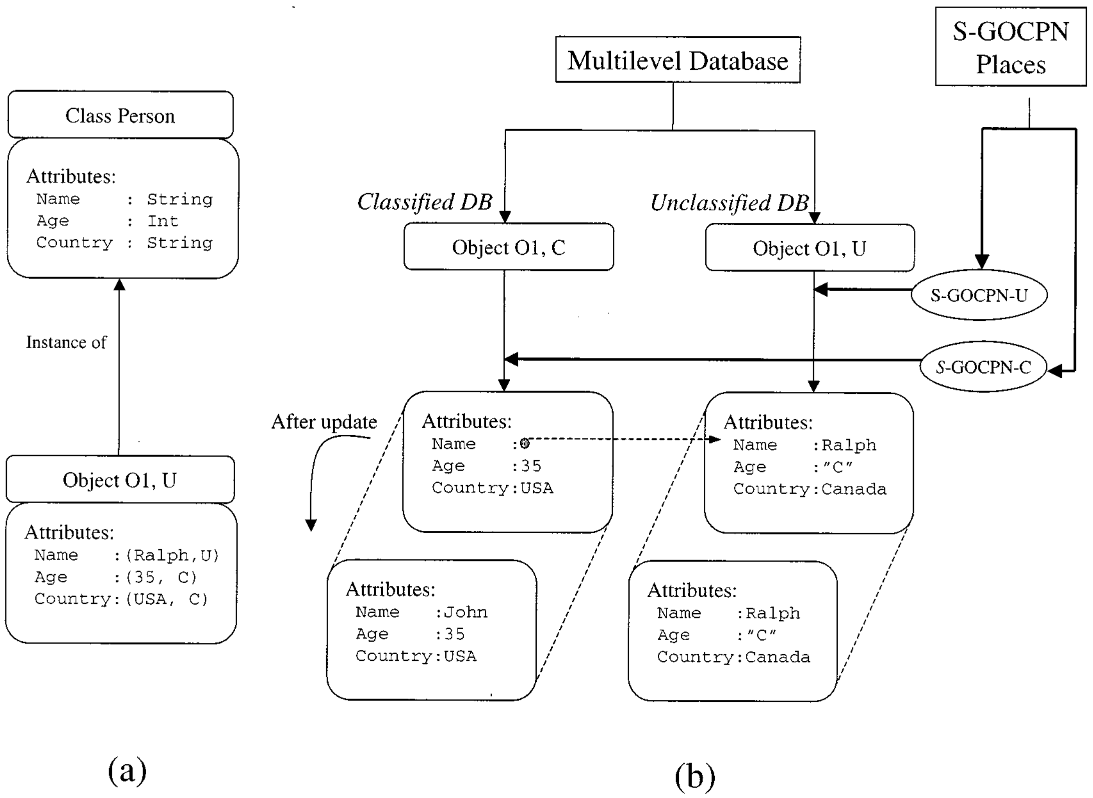


Fig. 6. (a) Multiview model. (b) One view for each level.

Fig. 6(b). Now, a user with clearance U can view the *unclassified* DB. For the user, attribute `Age` is hidden, while attributes `Name` and `Country` are visible. However, the user is unaware of the fact that the value for *country* is a "*cover story*" and thinks `Country` is an *unclassified* attribute and thus, thinks that "Ralph" is from "Canada."

User A can later update the classified information using his clearance C and only updates the object state, which is stored in the *classified* DB. For example, using his clearance C, he can change the value of `Name` to "John" in the *classified* DB. After this update, the link from *classified* DB's `Name` attribute to the *unclassified* DB's `Name` attribute is removed. At this time, for a user with clearance U, the name will appear as "Ralph" and the user will not know that the actual name is "John." The value "Ralph" in the *unclassified* DB now becomes a "*cover story*."

B. The Secure-GOCPN Model

To allow specification of multilevel classified multimedia documents and an MLS policy for multimedia documents, we extend the GOCPN model using a colored Petri-net model [20]. The model allows creation and retrieval of multimedia presentations that require accessing independent media objects belonging to different security domains. Each *security domain* consists of a set of users/subjects, objects and a classification scheme. To accommodate multiple security policies, we first define an SP for a domain as follows.

Definition 3 (Security Policy): An SP is a tuple $\{S, O, L, \leq, \lambda_p\}$, where

1) S is a set of subjects;
2) O is a set of objects;
3) $\{L, \leq\}$ is a lattice with a set of security levels L;
4) λ: $S \cup O \rightarrow L$ is a classification function.

Furthermore, the BLP restrictions are applied to control the flow of information.

Thus, a security domain represents the scope of an SP over a set of subjects and objects [21]. For a multidomain environment, let the ith SP be referred to as $\mathrm{SP}_i = \{S_{\mathrm{sid\_}i}, O_{\mathrm{oid\_}i}, L_i, \leq, \lambda_i\}$. Furthermore, let S_{sid} and O_{oid} denote the set of subjects and objects, respectively, in all the domains of a system, i.e., $S_{\mathrm{sid}} = \bigcup_i S_{\mathrm{sid\_}i}$ and $O_{\mathrm{oid}} = \bigcup_i O_{\mathrm{oid\_}i}$. For a multidomain environment containing n independent domains, we have a set of policies $\mathrm{SP}^n = \{\mathrm{SP}_1, \mathrm{SP}_2, \ldots, \mathrm{SP}_n\}$. Formally, we define a secure-GOCPN as follows.

Definition 4 (Secure-GOCPN): A *secure*-GOPCN *is a tuple* $\{G, \mathrm{SP}^n, \Sigma, C_p, G_t, E_a\}$, where:

1) G is a GOCPN defined earlier with the following changes to P and T: $P = \{$regular, decision, SPlace, EPlace, AMPlace, INPlace$\}$ and $T = \{$regular, gate-transtion, interoperation-transition$\}$.
2) SP^n is a finite set of security policies with $n \geq 1$.
3) $\Sigma = \{S_D, S_{\mathrm{AR}}, S_A, S_M, S_I, S1, S2\}$, where $S_D, S_{\mathrm{AR}}, S_A, S_I$, S1, *and* $S2$ are *types* or *color* sets. $S_D = \{\mathrm{sid} \,|\, \mathrm{sid} \in S_{\mathrm{sid}},\}$ is the set of *default tokens* (default color set), $S_{\mathrm{AR}} = \bigcup_i S_{\mathrm{AR-i}}$ *with* $S_{\mathrm{AR-i}} = \{(\mathrm{sid}, \mathrm{oid}) \,|\, \mathrm{sid} \in S_{\mathrm{sid\_}i}, \mathrm{oid} \in S_{\mathrm{oid\_}i}\}$ is the set of *authorization request tokens*, $S_A = \bigcup_i S_{A-i}$ *with* $S_{A-i} = \{(\mathrm{sid}, \mathrm{oid}, c) \,|\, \mathrm{sid} \in S_{\mathrm{sid\_}i}, \mathrm{oid} \in S_{\mathrm{oid\_}I}$ and c is the clearance sid has for oid in the ith SP $\mathrm{SP}_i\}$ is the set of *authorization tokens*, $S_M = \{(a_i, a_j) \,|\, a_i \in L_i$ and $a_j \in L_j\}$ is the set of mappings from level a_i domain SP_i

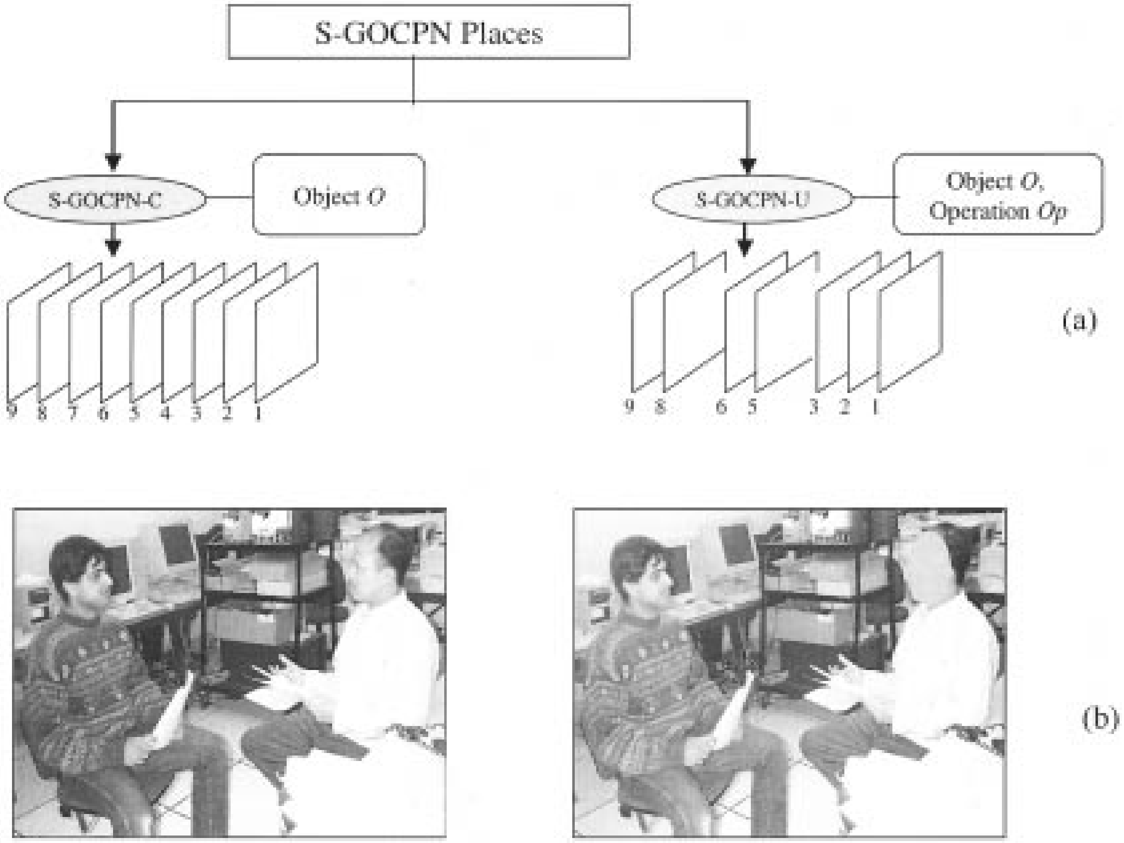


Fig. 7. Multiple views over a media object. (a) Dropping frames. (b) Hiding a portion.

to level a_j domain $\mathrm{SP}_j, S_I = \{(\mathrm{sid}, \mathrm{oid}, \mathrm{In}) \,|\, \mathrm{sid} \in S_{\mathrm{sid\_}i}, \mathrm{oid} \in S_{\mathrm{oid\_}i}, \mathrm{In} \in S_M\}$ *is the set of* interoperation token, $S1 = S_D \cup S_{\mathrm{AR}} \cup S_R$ and $S2 = S_M \cup S_A \cup S_I$.

4) C_p: $P \rightarrow \Sigma$ is a color function that assigns each place a color set from Σ; a place can acquire only a token of the color that is defined for the place. A *colored* token carries information about the identity of the subject, object, and/or the subject's security clearance.
5) G_t: $T \rightarrow B_E$ is a guard function defined from T into Boolean expressions B_E such that $\forall_t \in T$:$[\mathrm{Type}(G_t(t)) = \mathrm{Boolean} \wedge \mathrm{Type}(\mathrm{Var}(G_t(t))) \subseteq \sigma]$, where *Var* refers to variables in a guard expression and *Type* refers to the type of the variables.
6) E_a: is an arc expression from A to expressions such that Σ: $\forall_a \in A$: $[\mathrm{Type}(E_a(a)) = C_p(p(a)) \wedge \mathrm{Type}(\mathrm{Var}(E_a(a))) \subseteq \Sigma]$, where *Var* and *Type* are the same as above.

In the following, we present the representation of multilevel objects in *secure*-GOCPN and then describe how security specification in a single domain as well as across multiple policy domains can be achieved by using *secure*-GOCPN.

1) Multilevel Objects in Secure-GOCPN: We can elaborate the following two scenarios in which: 1) the component objects are managed by a database that supports a multilevel protection and 2) the component objects are independent files or objects without any multilevel decomposition, such as a movie file, an image file, etc.

In the first example, to classify multilevel objects, places can be used. For example, places S-GOCPN-C and S-GOCPN-U in Fig. 6(b) refer to the objects in *classified* and *unclassified* DB.

In the second example, a multilevel classification scheme is needed for the objects in a document. For example, a movie file may contain scenes that display some "*violence*" and is not suitable for a child viewing. In order to allow a child to view the movie, a child's view must be created for the file where violent scenes are removed. Storing a separate copy of the movie with the scenes removed is a costly choice. A better approach is to provide a real-time video operation such as dropping the frames that contain "*violence*" scenes. Fig. 7(a) shows two views of a video sequence; in the *unclassified* view, the fourth and the seventh frames are dropped during presentation, whereas in the *classfied* view, all frames are presented. Fig. 7(b) depicts two views of an image. In the *unclassified* view, the face of a person has been covered to protect the identity of the person. Hence, a view of a media object such as a video sequence can be represented by specifying such a media operation on the object. It can be noticed that the structure of GOCPN supports views for objects by allowing filtering operations to be specified for each place using the place function ObjA (*Definition 2*).

In *secure*-GOCPN, we represent a multilevel object with n possible classifications by n places, where each place represents the view of the object at a given classification level. We add an extra place, called *default place*, for each multilevel object and to represent the "*access denied*" path during firing of the *secure*-GOCPN. In other words, if a subject is not cleared authorized for an object, this special place is selected. This place can be used to retain any synchronization constraint introduced

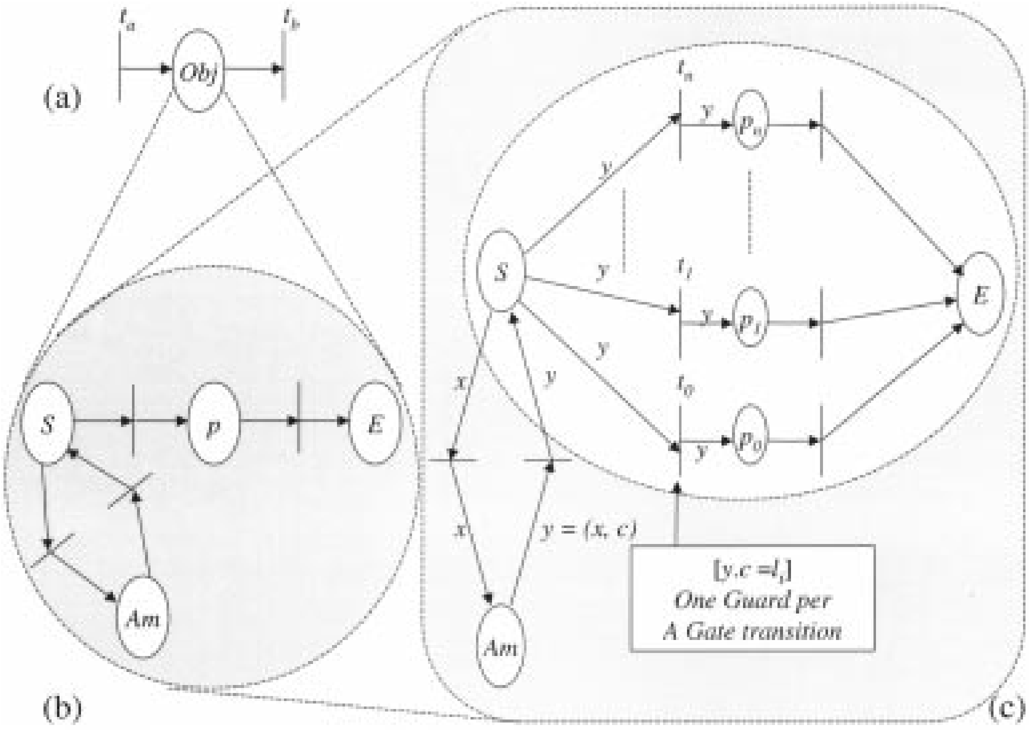


Fig. 8. Representation of multiple views of a multilevel object.

by the multilevel object. In case there is no need for any extra functionality, we can assume that the *default place* merely forward the token without consuming any time. Thus, each multilevel object is represented in *secure*-GOCPN by $n + 1$ places, as shown in Fig. 8. Firing of a transition t_i implies that the ith level object, i.e., p_i, is presented.

2) Security Boundary and Access Control: A *secure*-GOCPN can handle multiple predefined security policies that provide classification schemes for the media objects in the multidomain environment. For using *secure*-GOCPN model, a media object needs to be associated with the proper security domain that provides access rules for the objects. In order to provide this functionality, we introduce the three special places of *secure*-GOCPN, namely, *SPlace*, *AMPlace*, and *EPlace*.

An *SPlace* is used before each multilevel object, as shown in Fig. 8. An *SPlace* is labeled with the symbol "S." For example, in Fig. 8, the secure object *Obj* in (a) is an abstract view of the *secure*-GOCPN net in (b) which in turn is an abstract view of the *secure*-GOCPN net in (c). Fig. 8(b) shows that document *Obj* that contains a multilevel object, represented by p, has different views depending upon the classification scheme used, as shown in Fig. 8(c). An *SPlace* is connected to an *AMPlace*. Tokens assigned to an *SPlace* is of color *SS*; that is, $C_p(\text{SPlace}) = \text{SS}$. Thus, an *SPlace* can contain a default token of color set S_D that it receives from other *Eplaces*. Alternatively, it can contain an authorization request token of color set S_{AR}, or an authorization token of color set S_A.

An *AMPlace* specifies an authorization module that is maintained by each security domain. It accepts authorization requests and generates authorizations. The arcs labeled x between places S and Am indicate that only tokens of color x can flow on these arcs, where x is an authorization request token. In other words, $x = (\text{sid}, \text{oid}) \in S_{\text{AR}}$. Upon receiving this token, Am generates an authorization token $y = (x, c)$, where c is the clearance the user sid has for accessing object oid, which in this case is p. For an authorization request that cannot be granted, $c = \texttt{NULL}$.

Once place S receives a y token from Am, one of the transitions $t_0, t_1, \ldots, t_n$ is enabled. Each of these transitions has a condition, called *transition guards*, which if satisfied, enables the transition. For the ith transition, the guard is "$y \cdot c = l_i$"; ($y \cdot c$ means clearance c in token y); that is, the ith transition is enabled if the clearance level c in the authorization token y matches classification level l_i of object p. Place p_i represents the view of object p at level l_i. The enabled transition fires immediately indicating that access has been granted for the particular view of the object. In this regard, these transitions act as access gates. Accordingly, such transitions can be referred to as *gate-transitions*. The outputs of *gate-transitions* are x tokens.

An *EPlace*, labeled as "E," appear at the output transitions of places $p_0, p_1, \ldots, p_n$, as shown in Fig. 8. *EPlace* represents exit from the multilevel object. From each place $p_0, p_1, \ldots, p_n$ up to the *EPlace*, only default tokens flow, and hence $C_p(\text{EPlace}) = S_D$. All arcs that are not labeled in the figures are assumed to carry default tokens. Although, no particular semantics is associated with an *EPlace*, it provides a modular representation of each protected object. *SPlace* and *EPlace* pair constitutes the security boundary of the associated multilevel object. Furthermore, an *EPlace* can be augmented with post-processing requirements such as security auditing.

We assume that each protected object in the system is represented in the Petri-net form shown in Fig. 8(c). When composing GOCPN documents, we use the abstraction places shown in Fig. 8(a). However, if a higher-level document has many multilevel subdocuments and if no authorization is required at the

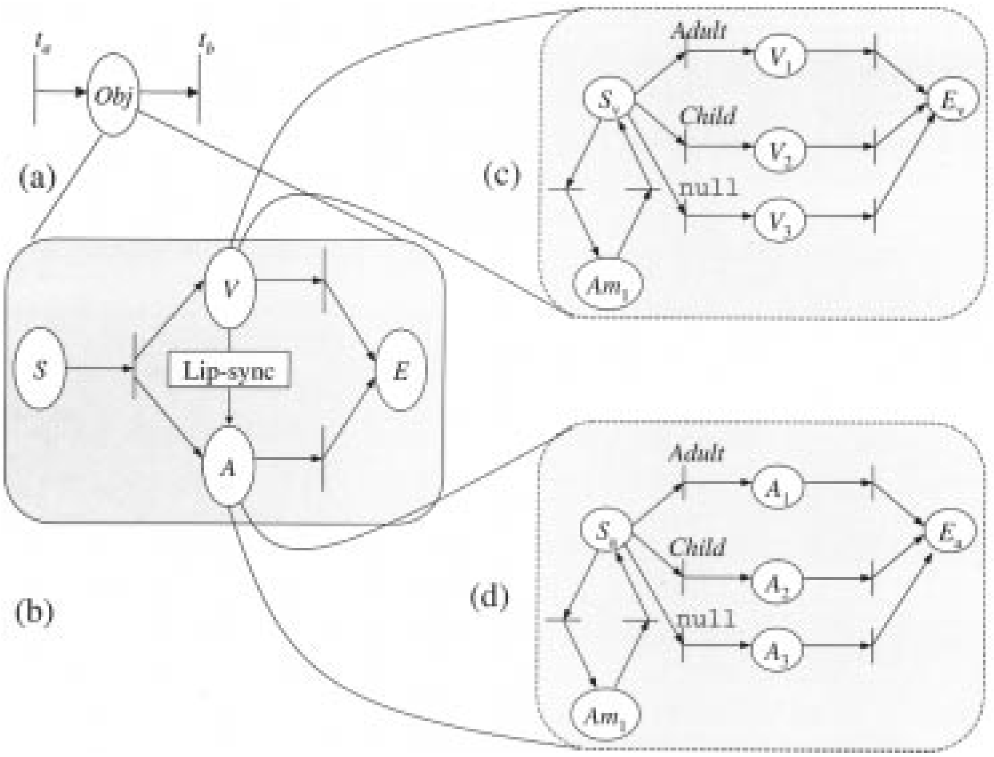


Fig. 9. Sample *secure*-GOCPN document containing multilevel objects.

highest level, then *SPlace* is not connected to any *AMPlace*. For example, in Fig. 9, a document is composed of two protected multilevel objects V and A. Place S needs not to send out authorization request to Am as the combination of V and A at this level is not classified. The clearance and authorization check is carried out at each multilevel subdocument V and A. Fig. 9(a) is the highest level of abstraction of the document and it can be used as a component in other documents. Fig. 9(b) shows an arrangement for subdocuments. Fig. 9(c) and Fig.9(d) show the detailed representation of the subdocuments V and A, respectively.

To make the model general and flexible, places $p_0, p_1, \ldots, p_n$ that represent the views of a place p in the protected document *Obj*, may not be restricted to the same multimedia object. Each p_i can reference a different GOCPN document. The user can apply application specific semantics to such classification. An example is a Web-site presentation for which p_{i+1} can be a document presented to "*privileged*" users and p_i can be a document presented to "*general*" public. Information viewed by a *privileged* user can be vastly different from the information viewed by a *general* user. In some cases, the information viewed by the *privileged* user can be a superset of the information viewed by the *general* user.

A place p_i that represents a view for a place p as in Fig. 8, is not necessarily attached to the same classification scheme at all the times. The document represented as p_i may appear as a view of another object within a different classification scheme in a different document. For example, if p_i represents an image, in one scheme it may appear as "*adult*" view of an object, where the classification scheme uses "*adult*" and "*child*" to classify objects based on the adult material present. However, in another scheme, p_i be used as a particular class of art when some artistic categorization is used. We note that there is no semantics attached to a place p_i in Fig. 8. The classification semantics is provided by *SPlace*, *AMPlace*, and the *gate-transitions*.

3) Hierarchical Protection: The proposed GOCPN model can be applied to achieve hierarchical classification on media objects [25]. We illustrate this concept with an example, depicted in Figs. 10 and 11. According to *Definition 2*, it can be noticed that $\mathrm{ObjA}: P \rightarrow \{C \times Q \times D \times S \times \mathrm{Op}\}$ where C represents a content set, Q represents the set of QoP parameters, integer set D represents durations, S represents spatial information, and *Op* represents an ordered set of media operations. Also, in the set of operation *Op*, operation op_i is applied on the media content before operation op_{i+1}. We show that by properly defining a set of operations we can apply hierarchical classification on a media object. To simplify our discussion below, we only consider sets C and *Op*; that is, $\mathrm{ObjA}: P \rightarrow \{C \times \mathrm{Op}\}$. In Figs. 10 and 11, we have a base, unclassified object represented as p, which is a high-quality movie file.

In Step 1(a) of Fig. 10, a classification scheme is applied that uses "*adult*" and "*child*" as classification labels. The result is a multilevel document O_1 that is shown in Fig. 10. Am_1 represents the authorization module associated with this scheme. The "*adult*" view of p is p_1 and constitutes playing out the original movie file p, whereas the "*child*" view of p is p_2, which constitutes playing out the actual movie without frames that contain adult material and are enumerated in set F. Hence, p_2 represents an application of an operation $\mathtt{drop}(F)$ on p. In other words, $\mathrm{ObjA}(p_2) = (p, \{\mathtt{drop}(F)\})$. To simplify the figures we have omitted the "*access denied*" path (indicated by `null`) in Fig. 10. Note that while places p and p_2 refer to the same base media

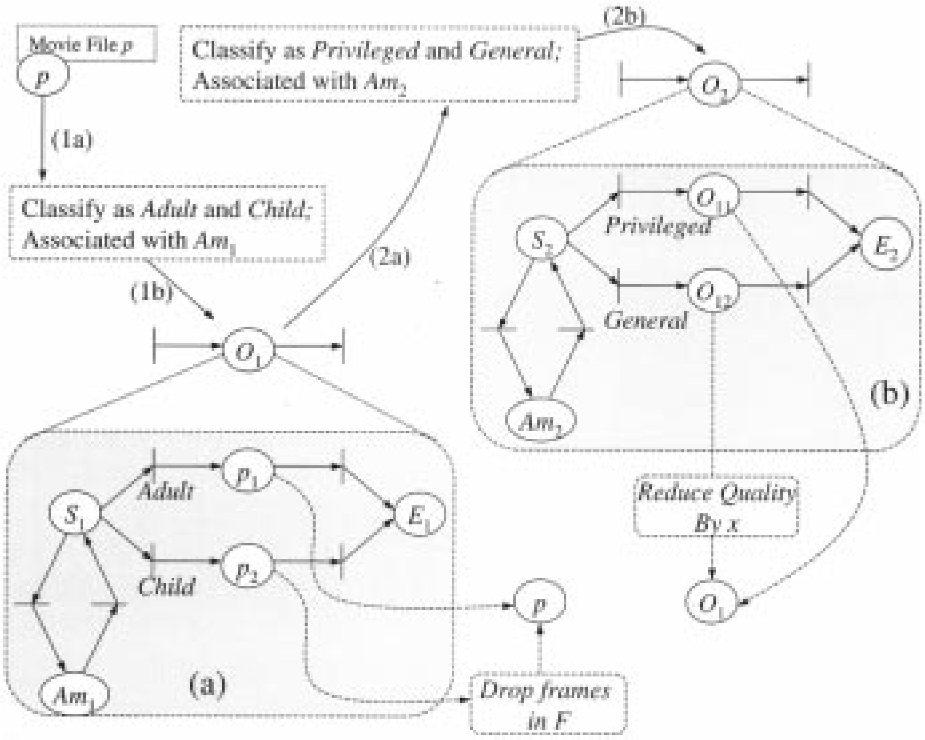


Fig. 10. A hierarchical classification scheme for multimedia objects.

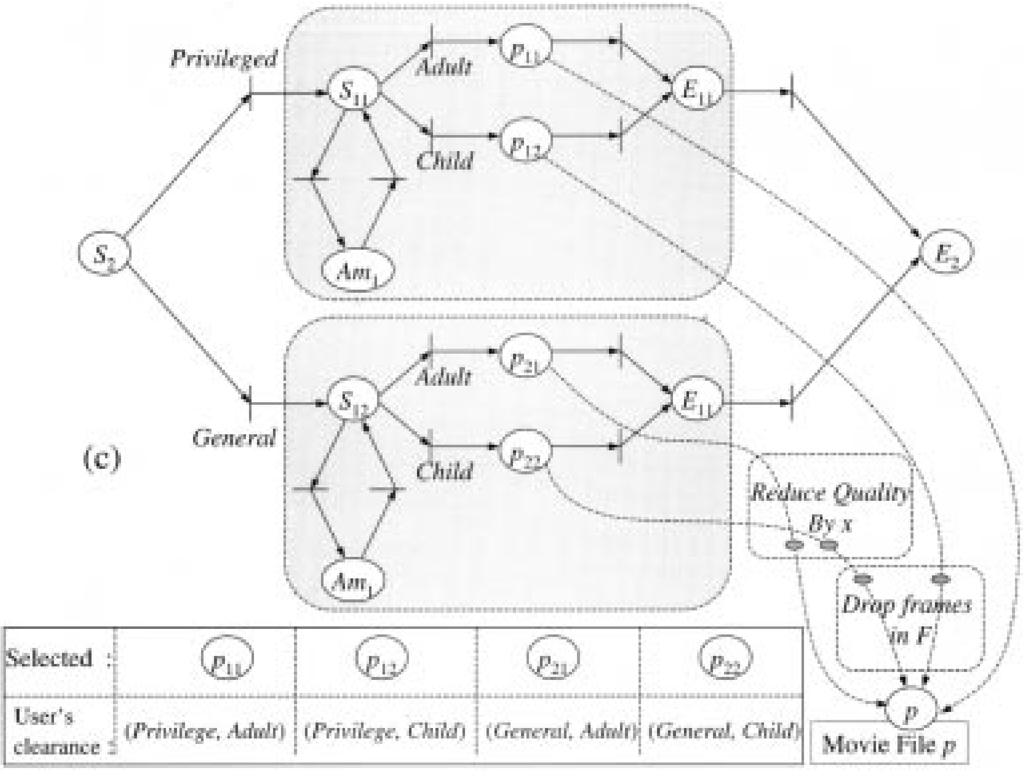


Fig. 11. Detailed view of the net in Fig. 10.

content, place p refers to the content as it is available in storage, whereas p_2 refers to a modified version of the base media content.

In step 2(a), another classification on object O_1 is carried out to get object O_2. The second classification uses "*privileged*" and "*general*" classes. A "*privileged*" user gets a high-quality presentation, whereas a "*general*" user gets a low-quality presentation as shown in Fig. 10(b). The place O_{11} represents the view of object O_2 at the "*privileged*" level, whereas O_{12} represents the view of object O_1 at the "*general*" level. The "*general*" level can be represented as $(O_1, \{\texttt{reduce\_quality}(x)\})$ as

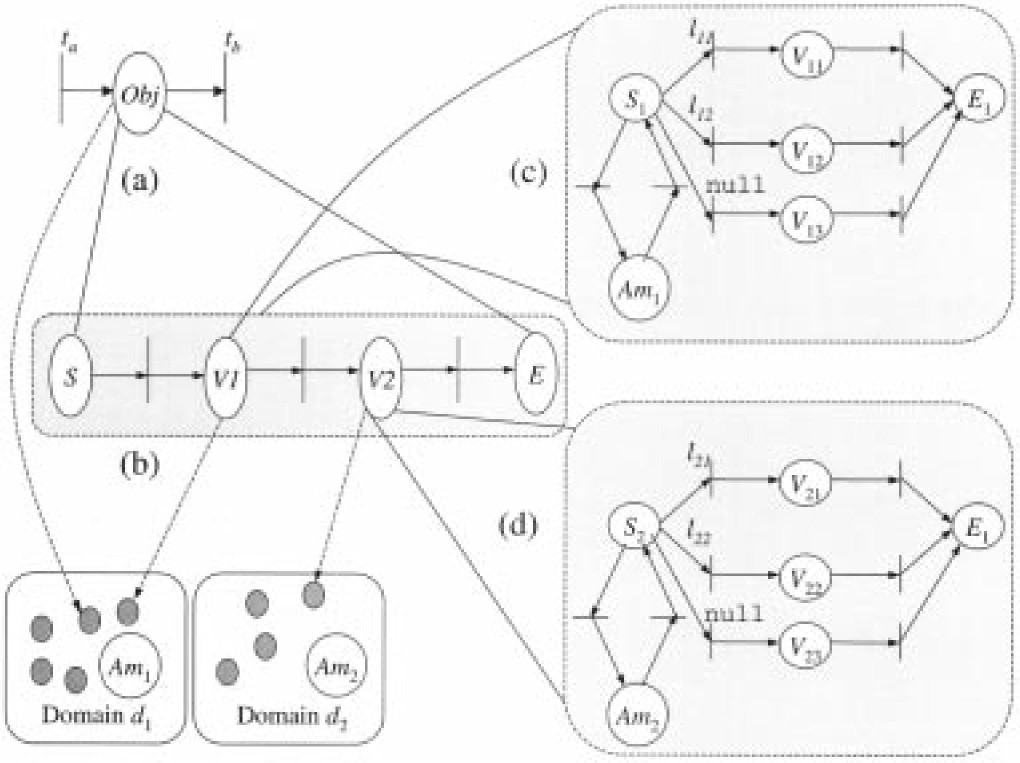


Fig. 12. Example of a composite document containing multiple documents from different domains.

it involves applying operation $\texttt{reduce\_quality}(x)$, where x indicates the reduction in quality on the content of O_1, which is the content of p with frames from the set F dropped. Fig. 11 shows the expanded view. Note that p_{11} represents the view of the file for an "*adult*" user having a "*privileged*" status and constitutes the presentation of the movie at its original form. Place p_{12} represents the view of the file for a "*child*" user having a "*privileged*" status and constitutes the presentation of the original movie with adult-oriented frames dropped, which are represented by a small oval on the dotted line from p_{12} to p. Similarly, p_{21} represents the view of the file for an "*adult*" user having a "*general*" status and constitutes a reduced quality movie. Similarly, p_{22} represents the view of the file for a "*child*" user having a "*general*" status, which constitutes the movie with reduced quality as well as adult-oriented frames dropped. Hence, we can capture each view p_{ij} of p using the ObjA function in terms of the base object p as follows: $\mathrm{ObjA}(p_{11}) = (p, \texttt{null}), \mathrm{ObjA}(p_{12}) = (p, \{\texttt{drop}(F)\}), \mathrm{ObjA}(p_{21}) = (p, \{\texttt{reduce\_quality}(x)\})$ and $\mathrm{ObjA}(p_{22}) = (, p2, \{\texttt{reduce\_quality}(x)\}) = (p, \{\texttt{reduce\_quality}(x), \texttt{drop}(F)\})$. This process allows a hierarchical classification scheme. The process can be specified on media objects by defining an ordered set of operations on an object.

4) Interdomain Access: In numerous applications, a user may retrieve several independent documents from different security domains and develop a composite presentation. In such cases, multidomain access policies become a limiting factor in composing a multimedia presentation. In Fig. 9, each subobject belongs to the same domain (both use Am_1) and hence the same classification scheme applies to the subobjects. Fig. 12 shows a document composed from two documents that belong to two different domains d_1 and d_2. V1 belongs to d_1 and V2 belongs to d_2. The composed document Obj itself belongs to d_1. In this case, if a user has some clearance in each domain, then he is presented the views of objects V1 and V2 depending upon his clearance.

It is possible, however, that a user has clearance to only one domain, in which case he can view only one of the subdocuments, say V1 and V2. However, the system can be configured to provide a mapping between the classification lattices of the two domains to allow a user having a clearance in one domain to allow some level of clearance in the other domain [6], [12]. For example, in Fig. 13, domain d_1 employs four classification levels and domain d_2 employs five classification levels. The mapping a shown as a directed arc indicates that anyone cleared for level l_{22} in d_2 is allowed to view level l_{12} objects in d_1. Similarly, mapping b indicates that anyone cleared for level l_{13} in d_1 is cleared for level l_{23} objects in d_2. If mapping c is also included, it can create a violation within d_2. That is, a user cleared for l_{21} in d_2 gets clearance for l_{14} in d_1 and as l_{13} is mapped to l_{23}, he also gets clearance for l_{23} in d_2, thus allowing a user with only l_{21} clearance to view an object at a higher classification level l_{23}. Mappings a and b define the interdomain access allowed in a multidomain environment. Care must be taken to ensure that such mappings are consistent [17] so that they do not create security violation. In essence, such secure interoperation in a multidomain environment must ensure the *principles of autonomy* and *security* [17]. The principle of autonomy requires that whatever accesses are allowed within a single domain should still be allowed when an interdomain access is permitted between this domain and other domains through such mappings. The *principle*

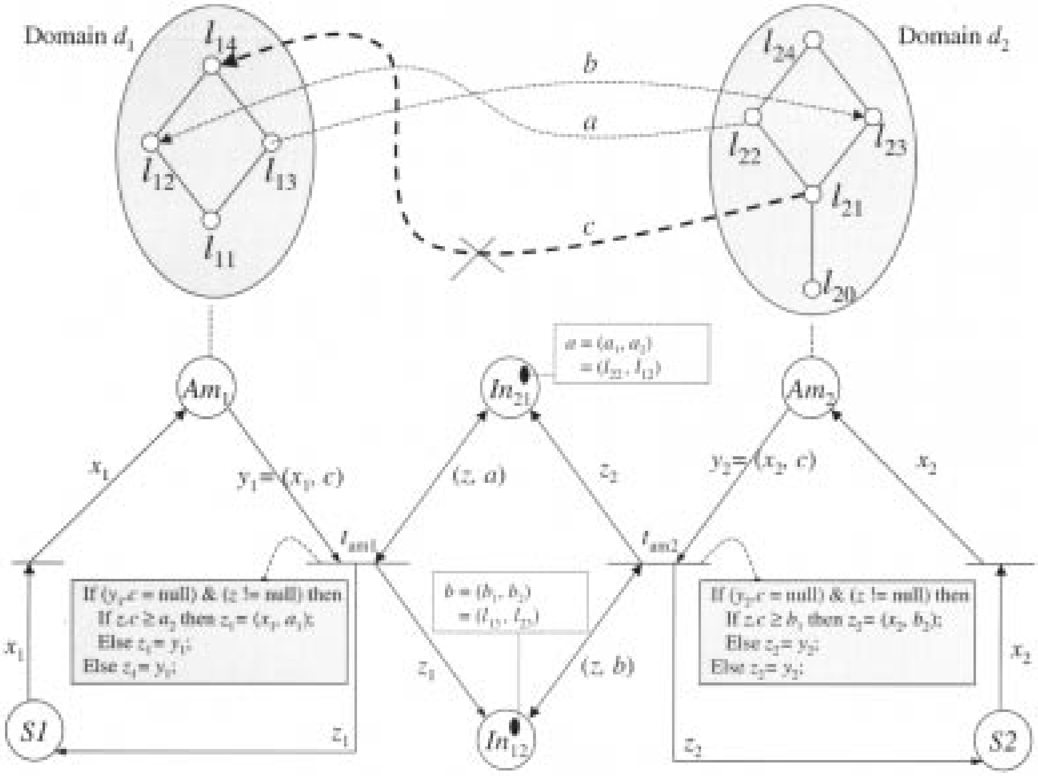


Fig. 13. Mapping between two domains using interoperation tokens.

of security requires that, if an access is not permitted in a domain, that access should not be allowed through interdomain accesses. Addition of arc c violates the principle of security. Here we assume that mappings between security lattices of different domains have been correctly established. For discussions that follow, we assume that mapping c does not exist.

To model such mappings between the levels of different domains, we introduce a place to represent each one directional set of mappings. For example, for a mapping b from d_1 to d_2, we use place In_{12} (an *INPlace*), which always contain a token that contains the mapping information. For example, In_{12} contains token (b_1, b_2), where b_1 represents the level of d_1 mapped to level b_2 of domain d_2. In_{12} accepts an authorization token from the associated *AMPlace*, in this case Am_1. Note that $C_p(\mathrm{INPlace}) = S_M \cup S_I \cup S_A$. As mentioned earlier, the authorization token is used to store the information as to what clearance a user has. In_{12} outputs a token (z, b) to $t_{\mathrm{am}2}$ (an *interoperation* transition), which fires upon receiving a token each from In_{12} and its associated Am_2. The firing condition for $t_{\mathrm{am}2}$ is that the user has a clearance in d_2. If the user has such a clearance then the c value in the token received from Am_2 is nonnull. However, if the user does not have the required clearance, the mapping information is used. Similarly, an interoperation place In_{21} is created for d_2 to d_1 mapping. For each Am_i, we have a t_{am_i} and $\mathrm{In}_{ij} \in t_{\mathrm{am}\_i}$.

IV. An Architecture for Multimedia Information System (MMIS)

In this section, we present an architecture for an MMIS that allows creation, storage, retrieval, and presentation of information related to individual media as well as complex multimedia documents. It uses *secure*-GOCPN to model secure multimedia documents. The overall architecture of the system is illustrated in Fig. 14. The highest-level module consists of a *multimedia user environment (MUE)* through which an application developer can compose, query, manage authorization, and display multimedia documents and objects. It provides an interface between a naive user and complex multimedia services given by the underlying systems.

The *script translator and deadline generator (STDG)* processes a *secure*-GOCPN document and produces an instance of GOCPN document based on user clearances, during which it interacts with the *authorization manager (AM)* to determine the clearances a user has for different views of each multilevel object that appears in the *secure*-GOCPN document. The AM maintains the information related to each security domain and redirects authorization requests sent to it. The resulting GOCPN script is then used to construct a hierarchical structure of the document suitable for database search and retrieval. A multimedia document playback deadline table is generated from hierarchy. This table is used by the *multimedia presentation manager (MPM)* to schedule real-time presentation of the document.

At the time of saving a multimedia document, *multimedia query processor (MQP)* accepts the inputs from STDG and multimedia authoring tool (MAT), and stores the document script (or *secure*-GOCPN graph) into databases along with other information. For querying and retrieval purposes, MQP first translates user-level query specifications into *common multimedia query language (CMQL)* supported by the underlying DBMS (which is Postgres in this case) and then communicates with DBMS to get back searching results.

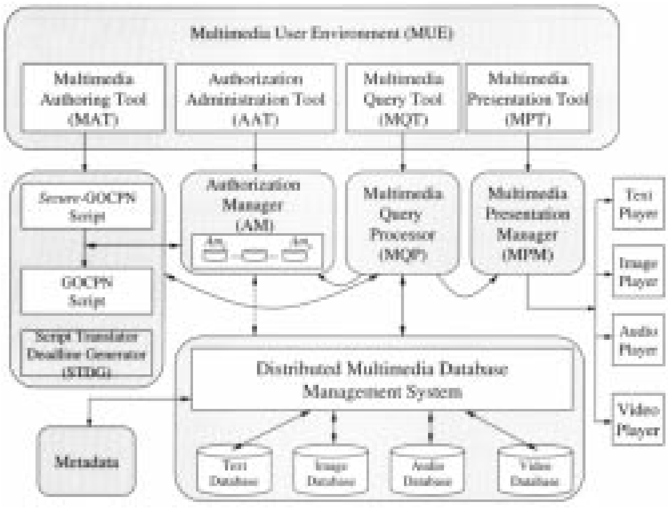

Multimedia User Environment (MUE)
Multimedia Authoring Tool (MAT)
Authorization Administration Tool (AAT)
Multimedia Query Tool (MQT)
Multimedia Presentation Tool (MPT)
Secure-GOCPN Script
Authorization Manager (AM)
Am
Am
Multimedia Query Processor (MQP)
Multimedia Presentation Manager (MPM)
Text Player
Image Player
Audio Player
Video Player
GOCPN Script
Script Translator Deadline Generator (STDG)
Distributed Multimedia Database Management System
Metadata
Text Database
Image Database
Audio Database
Video Database

Fig. 14. Architecture for MMIS.

The MPM is responsible for playing back multimedia documents in real time. It needs to simultaneously control multiple output devices, fetch media data from their sources to output devices and enforce both inter-media and intra-media synchronization constraints. Dynamic control and QoP adaptation for presentation are also necessary to accommodate user interactions and system's bottlenecks. In the following subsections, we discuss these components in detail.

A. The Multimedia User Environment (MUE)

In contrast to traditional textual data, multimedia presentation can be highly interactive. This requires proper design of easy-to-use *graphical user interface (GUI)* that interacts directly with the end users and supports services related to the underlying database. Therefore, MUE, a GUI tool based on X/Motif, is designed and implemented to author, query and display multimedia documents and objects. MUE communicates with layers underneath and requests services from them on behalf of a user's request. There are four basic tools in MUE: 1) *MAT*; 2) *authorization administration tool (AAT)*; 3) *multimedia query tool (MQT)*; and 4) *multimedia presentation tool (MPT)*.

MAT provides facilities that allow users to interactively compose, store, and modify multimedia documents. After the *secure*-GOCPN-based composition of a document is completed, the script generator interprets the Petri-net graph, creates a *secure*-GOCPN script by extracting its structural information. This script can be saved in a file or database for later reference. In order to locate and retrieve multimedia data from databases, MQT provides ad hoc query interfaces that allow a user to formulate queries in the form of scripts, tables, graphs, or their combinations. It also communicates with MQP for query execution. AAT allows a security administrator to define new classification policies that can be applied during the composition of a document, assign user clearances, and specify interdomain accesses. MPT is designed for controlling multimedia presentation interactively in real time. It accepts inputs from users and requests corresponding services from presentation manager. Subsequently, MPT allows users to play back selected multimedia documents with VCR-type functionalities.

B. The Script Translator and Deadline Generator (STDG) Module

The *secure*-GOCPN scripts generated by MAT consist of a transition script followed by a place script. In the transition scripts, the total number of transitions is recorded along with information about individual transition such as its index, indexes of its input places and output places. When a user wants to view a *secure*-GOCPN document, a GOCPN instance is created based on his clearances to component objects.

The GOCPN instance is represented in a GOCPN script, as shown in Fig. 16. The place script contains information about individual multimedia objects, such as object type, duration, object filename, display position, size on the screen, etc. The type of object is represented by a number, e.g., 1 for image, 2 for text, etc. An example of a GOCPN script file has a format as shown in Fig. 16.

The GOCPN script generated is interpreted by the script translator, which converts it into a hierarchical structure by using the temporal operator described in [28]. In addition, all the information related to data places is also retrieved from the script. The *deadline generator* processes the hierarchy and determines the play-out deadline for each data place in a given GOCPN instance, using the deadline algorithm in Fig. 15.

C. The Authorization Manager (AM)

The AM maintains information about various security domains and makes authorization decisions. This module is responsible for processing authorization requests from user and uses algorithm `secure-GOCPN-to-GOCPN( )` of Fig. 15. When an associated input transition to an *AMPlace* from a *SPlace* is fired, the translator algorithm (see Fig. 15) sends a message to the authorization agent based on the authorization request token along with the identity of the policy depending upon the *AMPlace*.

The AM maintains a set of tables for each policy and a policy-mapping table for interdomain accesses, as shown in Fig. 18. Associated with each policy is a table that lists the objects it governs. We note that this table may be empty, as the policy may be an application level policy that is applied on objects belonging to other domains. For example, in Fig. 10, the base object is the movie file p, which actually resides in the underlying database. When the object O_1 is composed, we do not need to explicitly consider O_1 as an object belonging to a policy domain of Am_1, although it is implicit. All it is saying is that only "*privileged*" or "*general*" classes of users as defined by Am_1's policy can be cleared for whatever subobjects that this document is composed of. Thus, we need not make an explicit entry in the object table for O_1.

D. Object-Oriented Multimedia Data Model

In this subsection, we discuss our design of object-oriented data model for multimedia information and corresponding database schema for indexing purposes. The proposed data

```
Algorithm Secure-GOCPN-to-GOCPN(oid, uid)
// Input: secure-GOCPN document oid, user's id uid.
// Output: GOCPN document instance for uid.
Let $S, E$ be the SPlace and Eplace of oid;
Let $T_{old}$ be the transition set document oid;
$Source = S$; $End = E$;
Place $token_d = (uid, oid)$ and on $S$
$P_{new} = P_{new} \cup \{S\}$;
Let $T = T_{old}(S)$;
$PVisited = \{S\}$;
$EnabledT$ = computeEnabled ($T$);
For each $t \in EnabledT$ do
    If $\exists p \in P_{in}(t)$ s.t. $(PT(p) \;!= AMPlace)$ then
        $T_{new} = T_{new} \cup \{t\}$; $PVisited = PVisited \cup$ fire($t$);
For each $p \in PVisited$ do
    $P_{new} = P_{new} \cup \{p\}$;
    $F_{new} = F_{new} \cup \{(t, p)\}$;
    $PVisited = Pvisited - \{p\}$;
    $T = T - \{t\}$;
    $EnabledT$ = computeEnabled ($T$);
// Remove unwanted SPlaces and EPlaces
For each $p \in P_{new}$ s. t. $p$ is SPlace or Eplace do
    If ($p$ != End and p != Source) then
        $T_{new} = T_{new} - = T_{in}(p)$; $P_{new} = P_{new} - \{p\}$
        For each pair $(p', t)$ s.t. $p' \in P_{in}(T_{in}(p))$ & $t \in T_{out}(p)$ do
            $F_{new} = F_{new} \cup \{(p', t)\}$;
    Else
        If ($p$ == End) then
            Rename $p$ as End and $(t, p)$ to $(t, End)$;
        If ($p$ == Source) then
        Rename p to Source and $(t, p)$ to $(t, Source)$;
```

```
Algorithm generate-deadline(GOCPN) I
// auxiliary variables for firing time
$st'[i] = -1$ for $0 \leq i < n$
// source transition firing time
$st[0] = st'[0] = 0$
$sp[j] = st[0]$ for all $j$ s.t. $p_j \in P_{out}(t[0])$
While(!done)
    /*minimum firing time of unfired ransitions*/
    $min\_st = \infty$
    /* index of transition with min firing time*/
    $min\_index = -1$
    $done = TRUE$
    for each $t[i']$ s.t. $st[i]<0$ and $sp[j] \geq 0$, $d[j] \geq 0$ for all $j$ s.t. $p[j] \in P_{active}(t[i])$
        $st'[i]$ = the $tw(t[i])$'s smallest value in $\{sp[j] + d[j] \mid \forall p[j] \in P_{active}(t[i])\}$
        if $st'[i] < min\_st$ then
            $min\_st = st'[i]$
            $min\_index = i$
        if $min\_index \geq 0$ then $st[min\_index]$ = min_st
            fire($min\_index$)
            $done = FALSE$
```

```
Algorithm fire (i)
$Tlist = \emptyset$   /* transition index set
for each $p[j] \in P_{in}(t[I])$
    if $sp[j] \geq 0$ and ($d[j] > st[i] - sp[j]$ or $d[j] < 0$) then
        $d[j] = st[i] - st[j]$
    if $st[j] < 0$ and $t[k] = T_{in}(p[j])$ then
        $st[k] = st[i]$
        insert $k$ into Tlist  /* (no duplication)*/ end
for each $k \in Tlist$
    fire(k)
    for each $j$ s.t. $p[j] \in P_{out}(t[I])$
    $sp[i] = st[i]$
```

Fig. 15. Algorithms Secure-GOCPN-to-GOCPN( ), Generate-Deadline( ), and Fire( ).

Transition Script		**Place Script**	
*Transition num	6	*Object num	8
Tans index	1	*Obj index	1
Input place num	0	Obj type	1
Input places		Obj input trans #	1
Output place	4	Obj output trans #	2
num	1 4 5 7	Obj filename	object
Output places		Method	rotate
	2	Obj duration	25.00
Trans index	1	Obj size-in-bytes	253200
Input places	1	Obj WinPosition	100
num	1	X	100
Input places	2	Obj WinPosition	400
Output place		Y	400
num		Obj WinSize X	
Output places		Obj WinSize Y	
.		.	
.		.	
.		.	

Fig. 16. GOCPN format.

```
Class MMobject  {
    String oid;
    String author;
    String objectPointer;
    String subject;
    String title;
    String description;
    Int size;
    Date date;
   String format;
}

Class MMdocument :
   Class MMobject { }

Class Image : Class GraphicalObject
{
   Int pixelDepth;
   String colorModel;
   String iconPtr;
}
```

```
Class Text : Class Mmobject {
   String language;
}
Class GraphicalObject : Class MMobject {
   Int width;
   Int height;
}
Class Video : Class GraphicalObject {
   Int pixelDepth;
   String colorModel;
   Int duration;
   Int rate;
   String iconPtr;
}
Class Audio : Class MMobject {
   Int duration;
   Int rate;
   Int bitsPerSample;
   Int nChannnels;
}
```

Fig. 17. Object classes.

schema is stored in database and facilitates users to do temporal and spatial searches as well as traditional content-based searches. Object-oriented technology is effective in modeling and manipulating multimedia data [15], [32], [39]. It provides an abstraction by integrating data, attributes, and method in a single construct that is well suited for multimedia applications.

E. Generic Multimedia Objects

There are some common features and properties shared by all objects in the multimedia object hierarchy. They constitute the generic class of multimedia objects that is the super class of all multimedia objects. This generic multimedia object class is defined as Class MMobject, as shown in Fig. 17. Fig. 17 also

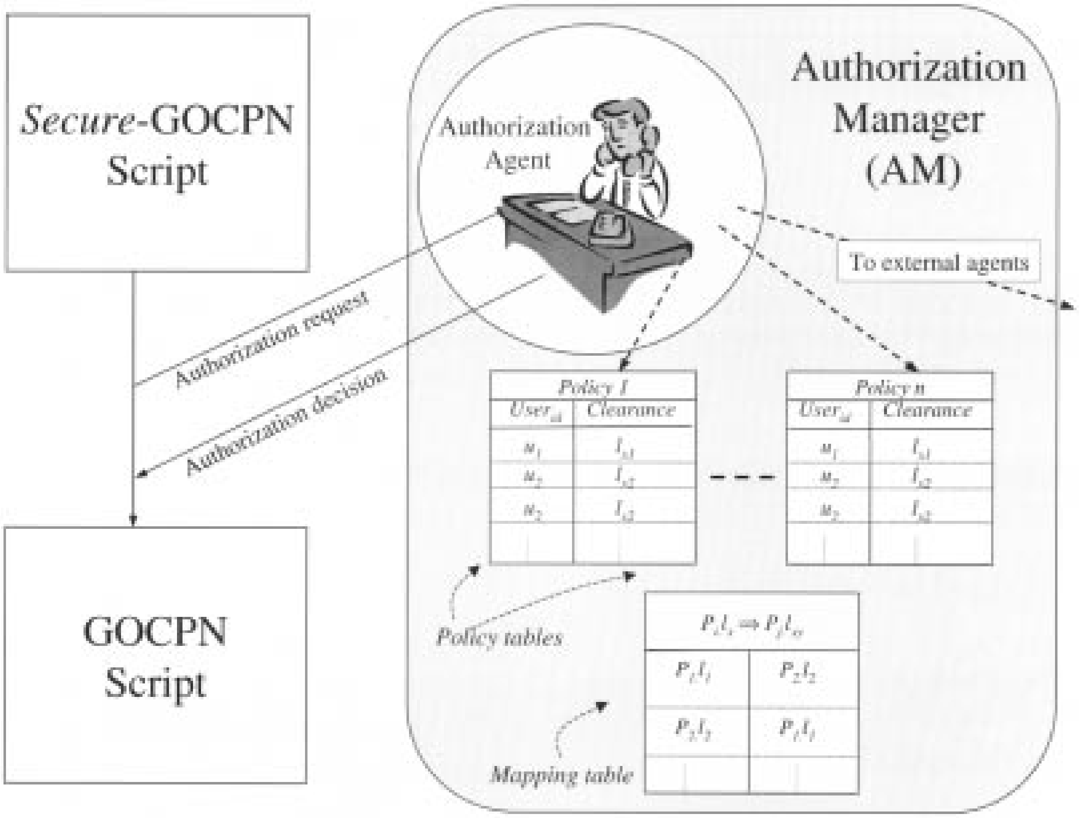


Fig. 18. Authorization manager.

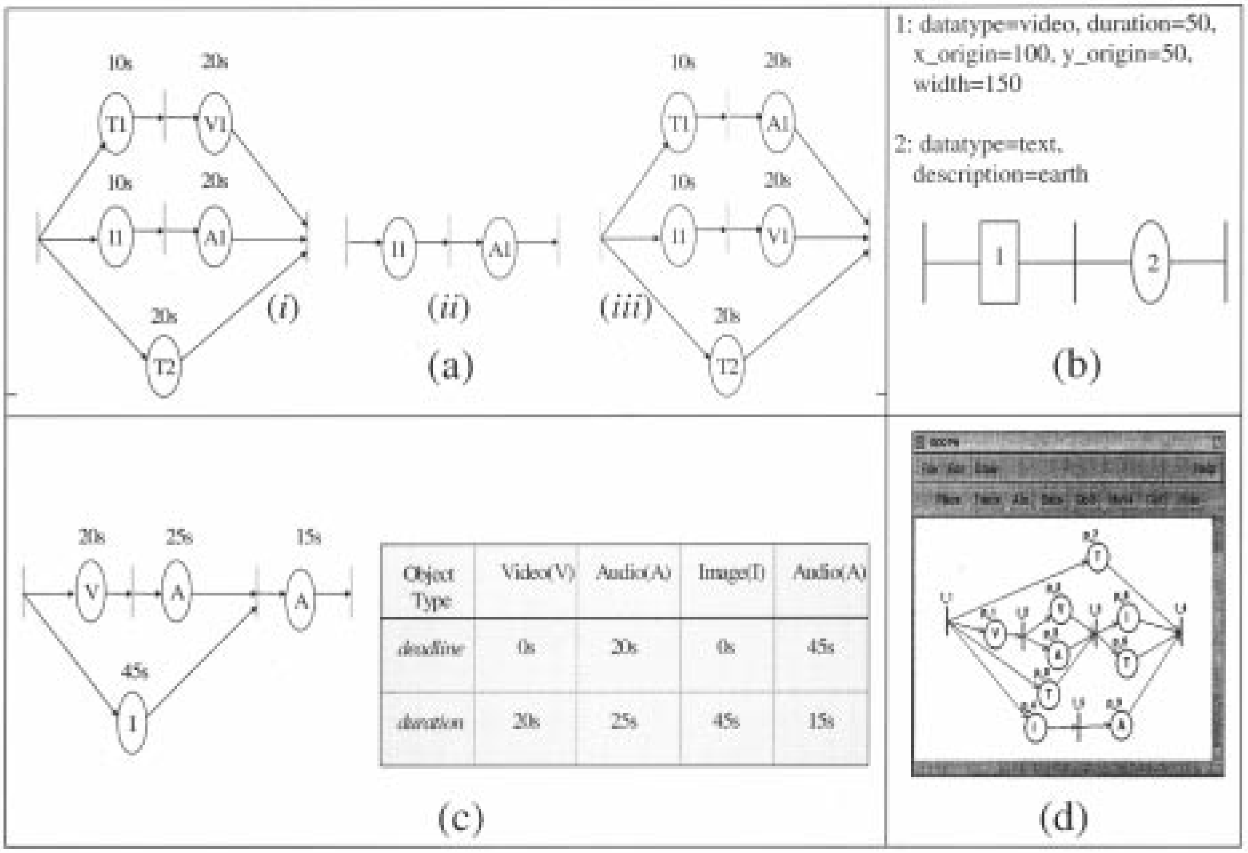
defines classes for each object type. Each media object requires a unique identifier, oid, that distinguishes itself from all other objects throughout its lifetime within the system. An oid can be made unique by combining several of the following components: *network domain name, server ID, time stamp, object class,* and *object instance count*. An *objectPointer* points to the actual source of the media data providing a fast, direct access to data, and can be determined by a combination of the hostname or Internet address and a filename or device name.

Appropriate metadata is essential for displaying, indexing and processing multimedia data. The *title*, *author,* and *date* attributes are used for bibliography information. Bibliography data alone may not be sufficient for supporting content-based search on multimedia objects. Multimedia information usually has complex semantics and is extremely difficult for computer interpretation. Thus, it is necessary to derive metadata from the content of multimedia manually or semiautomatically. The *description* attribute gives details of media content and a key word search can be performed on the value of this attribute. Moreover, each multimedia object has a *subject* attribute that classifies the object into various categories. Different media have different display characteristics. The basic information of a multimedia object should be known before it can be interpreted and displayed correctly. The attribute *format* is provided to identify the representation scheme of associated media data. Sometimes, it is also necessary to know the data volume of multimedia object when considering the capacity of network, storage device, and other aspects of the system. Accordingly, we use *size* to specify the data volume of media object.

Texts can be either plain or formatted. Tex, Latex, and ODA are some of the examples of document formats. In our system, we use *format* and *language* attributes to differentiate various text formats and languages used. A graphical object is a subclass of the generic multimedia object with *width* and *height attributes* added that indicate the horizontal and vertical resolutions of an image measured in pixels. Image class inherits all the properties from *graphical object* and adds *pixelDepth* and *colorModel* for interpretation of pixel values and specifying the model of the color used. An *iconPtr* is used to point to reduced quality image to support fast browsing and previewing of images. A video objects is a subclass of a *graphical object* that adds time-related attributes, *duration* and *rate*, as shown in Fig. 16. Video data also uses attributes *pixelDepth* and *colorModel*. Rate specifies how many frames should be delivered per unit time. Since a video object demands huge storage and transmission bandwidth, it is also desirable to have a small icon for users to scan it very rapidly by displaying key frames only or video with reduced resolution. The location of such a video icon is referenced by *iconPtr*. Audio objects include *duration, rate* (samples per second), *bitsPerSample* (usually it is 8-bit for telephony or 16-bit for CD quality) and *NChannel* (number of channels used for audio playback) attributes.

F. Multimedia Query Processing

Users can formulate multimedia queries in the form of table, graph, script, or their combinations. MQT described earlier provides the necessary GUI tools for formulating a query in



1: datatype=video, duration=50,
x_origin=100, y_origin=50,
width=150

2: datatype=text,
description=earth

Object Type	Video(V)	Audio(A)	Image(I)	Audio(A)
deadline	0s	20s	0s	45s
duration	20s	25s	45s	15s

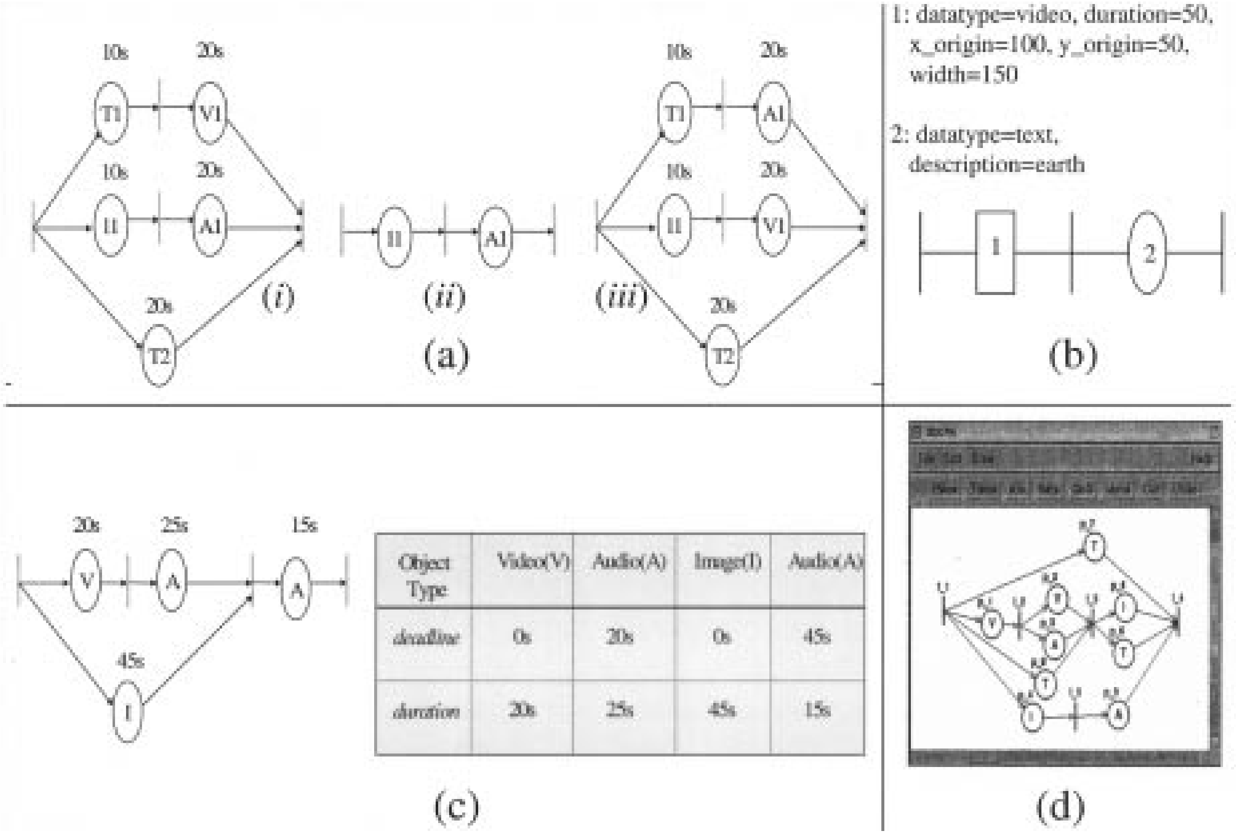
Fig. 19. Sample queries.

the content, temporal and spatial domains. This application level query cannot be interpreted directly by underlying DBMSs. MQP translates the user query into CMQL that is accepted by DBMS as well as some other functions, such as: 1) accepting queries in various forms from MQE and obtaining corresponding query data structures; 2) analyzing the query information and performing necessary processing; 3) setting up connections to databases, execute queries and get back results from databases; and 4) passing the query results to MQE for display. Another task of the query processor is to actually store the user created multimedia documents into the database for later retrieval and query. After a user composes a multimedia document through MAT, a command can be issued to the system to save the document. The STDG module then constructs the corresponding database tables and requests the query processor to insert tables into the underlying database. These tables form a template for a GOCPN instance created from secure-GOCPN document after user clearances are determined.

1) Content and Bibliographic Queries: Content-based query is perhaps the most common query in traditional information system. Bibliography query may include information such as *title, author, and creation date of multimedia objects*. A typical multimedia query is "*Find all pictures (paintings) about women made by Picasso before year 1930*." Upon receiving a query request from MQE, the MQP creates a SQL template and all necessary variables for the query. Then appropriate search-conditions are constructed and inserted into the SQL statements. The SQL codes generated is the QUERY1 in Fig. 19. The subject and text descriptions of multimedia contents described in the previous subsection are used for content-based retrieval. If the subject is specified, the corresponding field of multimedia objects should be successfully matched. On the other hand, if key words are used for search purposes, the multimedia objects with a text description containing these key words will be returned to the user. In this example, the image object *id, location pointer,* and *icon pointer* are returned.

2) Spatio-Temporal Queries: MMISs should provide necessary facilities to accommodate temporal and spatial queries as well. A multimedia document usually contains temporally, spatially and logically related multiple media objects. A user may ask for documents with certain structural information. For instance, a typical query can be "*Find a document which displays a video clip about Yellowstone National Park accompanied by an English text caption*." This query inherently has a temporal constraint, e.g., video and text should be displayed simultaneously.

Problems arise for such a temporal query when the representations of temporal relationship is not unique. For example, the two GOCPN structures shown in (i) and (iii) in Fig. 20(a) represent the same temporal relation. However, their net structures are different. When querying for GOCPN subnet shown in (ii) in Fig. 20(a), the direct matching will fail if the document is constructed as the GOCPN graph shown in (iii) in Fig. 20(a). In order to overcome this difficulty a universal canonical representation scheme (UCRS), which is unique and deterministic



1: datatype=video, duration=50,
x_origin=100, y_origin=50,
width=150

2: datatype=text,
description=earth

Object Type	Video(V)	Audio(A)	Image(I)	Audio(A)
deadline	0s	20s	0s	45s
duration	20s	25s	45s	15s

Fig. 20. (a) Different representation of a GOCPN. (b) Comprehensive query example. (c) Temporal query example; its deadline table. (d) GOCPN query result.

to a given temporal relation is needed. By translating GOCPN into this universal representation scheme, the matching can be carried out without ambiguity. Another reason for UCRS is to cope with the heterogeneity problems in synchronization specification schemes. In other words, the temporal and spatial relationships among media objects may be represented by various synchronization models.

For multimedia documents, the play-out deadline for each component media is calculated and stored in multiple deadline tables together with spatial information and other media specific metadata. The deadline-based representation of temporal relation is unique for each document. It is employed by MQP to calculate relative time ordering information and perform appropriate query processing on database. Fig. 20(c)–(d) shows an example of such a temporal query in GOCPN form. This temporal query is first translated into multimedia deadline tables with corresponding attributes instantiated by the STDG. The result is shown in the table in Fig. 20(c). The table is only for an illustration purpose, the actual deadline table contains more information than is shown here. From the deadline tables, the MQP constructs the CMQL (SQL) shown in QUERY2 in Fig. 19 and sent to the DBMS for execution.

3) Querying With Partial Knowledge: This kind of query is to find the multimedia document according to the partial knowledge of the document. For example, a user may want to read a multimedia document which has a video clip of Michael Jordan playing basketball. In this example, it is possible that the document description and other metadata for document do not contain such information. However, we may find it by looking into the metadata of the component media object within the multimedia document. This query can be specified by a GOCPN graph containing single image object with a content description of Michael Jordan. The resulting SQL query is QUERY3 in Fig. 19.

4) Comprehensive Query: In the proposed MMIS, a user can formulate complex queries by combining content and structural search. The content part is specified in table form, as described in the previous subsection, while structural part can be constructed through GOCPN authoring tools. Fig. 20(b) is a query "*Find an OCPN document having such a structural information with a subject about the earth*."

When the query processor accepts the query data structure from MQE, it is also informed by MQE of the search types. In this case, both content and structural searches are needed. Therefore, query processor uses this information and processes the user's structural constraints. It finally generates QUERY4 following SQL statements that combine both aspects of the query.

V. Related Work

Multimedia synchronization has been the focus of research in multimedia database community for the last several years [29], [24], [10]. Many temporal modeling schemes have been developed to describe multimedia synchronization requirements [19], [26]. Synchronization based on time line [19] is one of the early works in synchronization modeling. In this approach, all objects are linked to a reference time line and decoupled from each other. This allows removal or addition of presentation objects without affecting the synchronization of other objects. By introducing object time, a modified version discussed in [16] allows each object to be scaled, translated and inverted on the time line.

As the synchronization can only be defined based on fixed points in time, the models [19], [16] lack the ability to specify multimedia presentation with nondeterministic temporal information. The OCPN model was proposed in [29], [30], [28]. The authors show that each of Allen's 13 basic temporal relationships [1] can be expressed by a Petri-net. In this model, a place represents an object play-out duration while a transition represents synchronization point. A firing rule is used to specify the synchronization conditions. However, OCPN and its enhanced version also rely on the duration information of each media object and does not allow handling of the asymmetric relations involved in multimedia presentation. Li *et al.* have presented time-flow graph as a fuzzy temporal modeling scheme in [26], [27]. In their model, three additional temporal relationships, namely, shortest, longest and chosen, and two Boolean margin functions were introduced to enhance Allen's representation scheme. This approach allows nondeterministic temporal relationships to be represented to some extent. Unfortunately, some shortcomings of the OCPN model were still present in their scheme, because it is an interval-based model and mainly relies on Allen's 13 relations. In addition, the newly added three relationships assume that the composite object's starting time and ending time can only be determined by one of the component objects. Their model cannot handle general situations, as in the case where the starting time and ending time are determined by different objects. XML is also emerging as a powerful tool for multimedia document composition and presentation [41].

In hierarchical synchronization specification, a complex multimedia object is organized in a tree structure consisting of nodes, which denotes serial or parallel presentation of the outgoing subtrees [5]. Haindl [18] presented a model, which captures various level of multimedia synchronization granularity. Each data object is recursively decomposed into an ordered set of data subsets. The level of such decomposition depends on applications and the lower level subsets reflect the finer synchronization. This model lacks the capability to manage nondeterministic temporal relationships and incomplete timing.

Several research works can be found in the literature related to MLS, security in object-oriented systems, and distributed systems. Some work in security related to multimedia and hypertext systems can be found in [13], [14], [23], [36], and [38].

In [13] and [14], researchers discuss security and access control issues related to multimedia systems. A simple access control mechanism and verification for hypertext documents using Petri-nets has been discussed in [38]. It uses basic access control based on Petri-net constructs such as "*repeat* n *times*," "*forever repeat*," etc., but does not address MLS. In [23] and [36], authorization models for distributed systems focused toward WWW are proposed, both of which address the DAC mechanism for the systems like hypertext and WWW. To the best of our knowledge, the work presented here constitutes first research attempt in integrating modeling and specification of media synchronization as well as MLS of multimedia documents in a distributed, heterogeneous, multidomain environment.

VI. Conclusion

In this paper, we have presented an integrated Petri-net-based synchronization model that allows specification of application level, user-specific MLS. The model allows specifying interdomain access restrictions when subject/objects are governed by different classification policies in a multidomain environment. This facilitates composition of complex documents that have multilevel objects belonging to different domains and or classified using some hierarchical set of classification schemes. Such a capability allows composing a single document that constitutes different set of information for different users. The *secure*-GOCPN documents maintain such multilevel classified information throughout the life-cycle of the documents. More complex documents can be composed using such *secure*-GOCPN documents without any security concerns for multilevel objects comprising each such documents. In addition, *secure*-GOCPN has capabilities to model various multimedia synchronization requirements, such as deterministic and nondeterministic temporal relations, incomplete timing, asymmetric relations, user interactions, and *lip-sync* synchronization. We have also proposed the temporal constraint system that generalizes both deterministic and nondeterministic temporal relations for which *a priori* knowledge about interval durations and/or interval end points may not be available.

References

[1] J. F. Allen, "Maintaining knowledge about temporal interval," *Commun. ACM*, vol. 26, no. 11, pp. 832–843, Nov. 1983.

[2] A. Baraani-Dastjerdi, J. Pieprzyk, and R. Safavi-Naini, "A multi-level view model of secure object-oriented databases," *Data Knowl. Eng.*, vol. 23, pp. 97–117, 1997.

[3] D. E. Bell and L. J. LaPadula, "Secure computer system: unified exposition and multics interpretation," MITRE Corporation, Bedford, MA, MTR-2997, Mar. 1976. Available as NTIS AD A023 588.

[4] E. Bertino, M. A. Hammad, W. G. Aref, and A. K. Elmagarmid, "An access control model for video database systems," Dept. Computer Sciences, Purdue University, West Lafayette, IN, Tech. Rep. CERIAS TR 2000-04, 2000.

[5] G. Blakowski and R. Steinmetz, "Media synchronization survey: Reference model, specification, and case studies," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 36–51, Jan. 1996.

[6] P. Bonatti, M. L. Sapino, and V. S. Subrahmanian, "Merging heterogeneous security orderings," presented at the Proc. Computer Security—Esorics, LNCS no. 1146, E. Bertino, H. Kurth, G. Martella, and E. Montolivo, Eds., 1996.

[7] N. Boulahia-Cuppens, F. Cuppens, A. Gabillon, and K. Yazdanian, "Multilevel security in object-oriented data-bases," presented at the Proc. Workshop Computing—Security for Object-Oriented Systems, B. Thuraisingham, R. Sandhu, and T. C. Ting, Eds., Washington, DC, 1993.

[8] ——, "Decomposition of multilevel objects in an object-oriented database," presented at the Eur. Symp. Research in Computer Security, Brighton, U.K., 1994.

[9] J. F. K. Buford, *Multimedia Systems*. New York: ACM Press, 1994.

[10] J.-P. Courtiat, L. F. R. da Costa Carmo, L. F. Rust, R. Cruz de Oliveira, and R. Crus, "General-purpose multimedia synchronization mechanism based on causal relations," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 185–195, Jan. 1996.

[11] F. Cuppens and A. Gabillon, "A logical approach to model a multilevel object-oriented database," in *Database Security X: Status and Prospects*, P. Samarati and R. S. Sandhu, Eds. London, U.K.: Chapman & Hall, 1996.

[12] S. Dawson, S. Qian, and P. Samarati, "Providing security and interoperation of heterogeneous systems," *Distrib. Parallel Databases*, vol. 8, pp. 119–145, Jan. 2000.

[13] E. B. Fernandez, K. R. Nair, M. M. Larrondo-Petrie, and Y. Xu, "High-level security issues in multimedia/hypertext system," presented at the IFIP TC6/TC11 Int. Conf. Communications and Multimedia Security, Essen, Germany, Sept. 23–24, 1996.

[14] S. M. Funnell, N. J. Salmons, P. W. Sanders, C. T. Stocket, and M. J. Warren, "Approaches to security in healthcare multimedia systems," presented at the IFIP TC6/TC11 Int. Conf. Communications and Multimedia Security, Essen, Germany, Sept. 23–24, 1996.

[15] S. Gibbs, L. Dami, and D. Tsichritzis, "An object-oriented framework for multimedia composition and synchronization," in *Proc. Multimedia: Systems, Interaction, and Applications*, L. Kjelldahl, Ed., Stockholm, Sweden, Apr. 1991, pp. 101–111. Eurographics.

[16] S. Gibbs, "Composite multimedia and active objects," presented at the OOPSLA, 1991.

[17] L. Gong and X. Qian, "Computational issues in secure interoperation," *IEEE Trans. Software Eng.*, vol. 22, Jan. 1996.

[18] M. Haindl, "New multimedia synchronization model," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 73–83, Jan. 1996.

[19] M. E. Hodges, R. M. Sasnett, and M. S. Ackerman, "Athena muse: A construction set for multimedia applications," *IEEE Softw.*, pp. 37–43, Jan. 1989.

[20] K. Jensen, *Colored Petri-Nets—Basic Concepts, Analysis Methods, and Practical Use*, 2nd ed. New York: Springer-Verlag, 1996, vol. 1.

[21] J. Joshi, A. Ghafoor, W. Aref, and E. H. Spafford, "Digital government security infrastructure design challenges," *IEEE Comput.*, vol. 34, Feb. 2001.

[22] J. Joshi, W. G. Aref, A. Ghafoor, and E. H. Spafford, "Security models for web-based applications," *Commun. ACM*, vol. 44, pp. 38–72, Feb. 2001.

[23] J. Kahan-Oblatt. A distributed authorization model for WWW. presented at Proc. NET. [Online]. Available: http://inet.nttam.com

[24] L. Lamont, L. Li, R. Brimont, and N. D. Georganas, "Synchronization of multimedia data for a multimedia news-on-demand application," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 264–278, Jan. 1996.

[25] C. E. Landwehr, "Formal models of computer security," *ACM Comput. Surv.*, Sept. 1981.

[26] L. Li, A. Karmouch, and N. D. Georganas, "Multimedia teleorchestra with independent sources: Part I—Temporal modeling of collaborative multimedia scenarios," *ACM Multimedia Syst.*, vol. 2, no. 1, pp. 143–153, 1994.

[27] ——, "Multimedia teleorchestra with independent sources: Part II—Synchronization algorithms," *ACM Multimedia Syst.*, vol. 2, no. 1, pp. 154–165, 1994.

[28] M. Iino, Y. F. Day, and A. Ghafoor, "An object-oriented model for spacial-temporal synchronization of multimedia information," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*. Los Alamitos, CA, May 1994, pp. 110–119.

[29] T. D. C. Little and A. Ghafoor, "Synchronization and storage models for multimedia objects," *IEEE J. Select. Areas Commun.*, vol. 8, no. 3, pp. 443–427, 1990.

[30] ——, "Interval-based conceptual models for time dependent multimedia data," *IEEE Trans. Knowl. Data Eng.*, vol. 5, pp. 551–563, Aug. 1993.

[31] T. D. C. Little and A. Ghafoor, "Multimedia synchronization protocols for broad-band integrated services," *IEEE Trans. Commun.*, vol. 9, pp. 1368–1382, Dec. 1991.

[32] Y. Masunaga, "An object-oriented approach to temporal multimedia data modeling," *IEICE Trans. Inform. Syst.*, vol. E78-D, pp. 1477–1487, Nov. 1995.

[33] D. Minoli and R. Keinath, *Distributed Multimedia Through Broadband Communication*. Norwood, MA: Artech House, 1993.

[34] J. Peterson, "Petri-nets," *Comput. Surv.*, vol. 9, pp. 223–252, 1977.

[35] R. Steinmetz and K. Nahrstedt, *Multimedia: Computing, Communications and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[36] P. Samarati, E. Bertino, and S. Jajodia, "An authorization model for a distributed hypertext system," *IEEE Trans. Knowl. Data Eng.*, vol. 8, pp. 555–562, Aug. 1996.

[37] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: Toward a unified standard," in *Proc. 5th ACM Workshop Role-Based Access Control*, Berlin, Germany, July 26–27, 2000, pp. 47–63.

[38] P. D. Stotts and R. Furuta, "Access control and verification in Petri-net-based hyperdocuments," in *Proc. COMPASS*, 1989, pp. 49–55.

[39] M. Vazirigiannis and C. Mourlas, "Object-oriented model for interactive multimedia presentations," *Comput. J.*, vol. 36, no. , pp. 78–86, 1993.

[40] H. Fahmi, M. Latif, S. Sedigh-Ali, A. Ghafoor, P. Liu, and L. Hsu, "Proxy servers for scalable interactive video support," *IEEE Comput.*, vol. 34, pp. 54–60, Sept. 2001.

[41] Extensible Markup Language (XML) 1.0 (Second Edition) (2000, Oct. 6). [Online]. Available: http://www.w3.org/TR/REC-xml

James B. D. Joshi (M'00) received the B.E. degree in computer science and engineering from Motilal Nehru Regional Engineering College, Allahabad, India, in 1993, and the M.S. degree in computer science from Purdue University, West Lafayette, IN, in 1998. He is currently pursuing the Ph.D. degree in the School of Electrical and Computer Engineering at Purdue University.

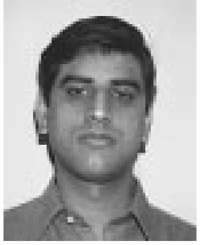
From 1993 to 1996, he held the Lecturer of Computer Science and Engineering position at Kathmandu University, Nepal. His research interests are computer security and distributed systems.

Dr. Joshi is a student member of the ACM.

Zhaohui Kevin Li received the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 1998.

From 1998 to 2001, he was with Cisco Systems. He is currently with Abeona Networks. His research interests include multimedia databases, document models, and networking.

Husni Fahmi received the B.S. degree in computer and electrical engineering in 1995, the M.S. degree in electrical engineering and 1997, and the Ph.D. degree from the School of Electrical and Computer Engineering in 2002, all from Purdue University, West Lafayette, IN. His research interests include MMIS, quality-of-service routing, and streaming multimedia.



Basit Shafiq received the B.S. degree in electronics engineering from GIK Institute of Engineering Sciences and Technology, Pakistan, in 1998, and the M.S. degree in electrical engineering from Purdue University, West Lafayette, IN, in 2001. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering at Purdue University.

Arif Ghafoor (S'83–M'83–SM'89–F'99) received the B.Sc. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 1976, and the M.S., M.Phil., and Ph.D. degrees in electrical engineering from Columbia University, New York, in 1977, 1980, and 1984, repectively.

Currently, he is Professor at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, and the Director of Distributed Multimedia Systems Laboratory. He has been actively engaged in research areas related to database security, parallel and distributed computing, and MMIS. He has published numerous technical papers in leading journals and conferences. He has served on the editorial boards of various journals including *ACM/Springer Multimedia Systems Journal*, *Journal of Parallel and Distributed Databases*, and *International Journal on Computer Networks*. He has served as a Guest/Co-Guest Editor for various special issues of numerous journals including *ACM/Springer Multimedia Systems Journal*, *Journal of Parallel and Distributed Computing*, and I*nternational Journal on Multimedia Tools and Applications*. He co-edited a book entitled *Multimedia Document Systems in Perspectives* and coauthored a book entitled *Semantic Models for Multimedia Database Searching and Browsing* (Norwell, MA: Kluwer, 2000).

Dr. Ghafoor served as a Guest/Co-Guest Editor for IEEE JOURNAL ON THE SELECTED AREAS IN COMMUNICATIONS and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. He received the IEEE Computer Society 2000 Technical Achievement Award.