Specifying Multimedia Access Control using RDF*

Naren Kodali[2], Csilla Farkas[3,4] and Duminda Wijesekera[1,2]
[1]Center for Secure Information Systems, [2]Dept of Info. and Software Eng.,
George Mason University, Fairfax, VA 22030–4444,
[3]Information Security Laboratory, [4]Dept of Computer Science and Eng.,
University of South Carolina, Columbia, SC-29208,
E-mail: nkodali@gmu.edu, farkas@cse.sc.edu, dwijesek@gmu.edu

Abstract

The *Synchronized Multimedia Integration Language (SMIL)* [Aya01] is an W3C [W3C03] specification for authoring multimedia documents. Although SMIL has XML like syntactic constructs, unlike XML, SMIL compositions have an intended semantics stemming from intuitive notions of playing out many media streams relative to each other. Although there are many excellent models for XML access control [DdVPS00, DdVPS02, BCF01, BCFM00, SF02, KH00, GB02, KW02, GF03], they do not respect the *intended meaning* of multimedia constructs. To remedy this, we propose a model for controlling accesses to SMIL documents by decorating them with appropriate RDF statements that respect these semantics. Using this model, we show how such documents can be fetched by secure runtimes from secure multimedia servers.

1 Introduction

SMIL [Aya01] is an XML-like language for authoring multimedia documents. Unlike XML decorated text data, SMIL constructs have an *intended meaning* that must be enforced by application runtimes. Therefore, any security policy specification has to respect this intended semantics. We propose a framework to do so for a chosen fragment of SMIL consisting of SMIL specifications constructed using sequential (⟨seq⟩) and parallel (⟨par⟩) composition operators.

The current SMIL Metainformation Module [Mic01] does not define or support security and QoS constructs, nor does it provide syntax to define complex relations within multimedia documents. Therefore, we propose a Resource Description Framework (RDF) [KC03, MM03] meta-structure to model security and QoS specifications in SMIL documents. Based on the proposed structure, we define relevant RDF decorations using the XML-RDF [KC03, MM03] syntax, which can be superimposed on SMIL documents in an appropriate form (to be discussed shortly) so that security and QoS specifications can be enforced by security and QoS aware runtimes. We have chosen to represent limited features of access control polices comprising of discretionary, mandatory (also called multilevel secure (MLS)) and role-based access control paradigms.

One of the problems encountered in doing so is that arbitrary SMIL (syntax) trees do not accurately represent their intended semantic hierarchy completely. This is important because security policies specify accesses to objects in a hierarchy and not to one of its syntactic representations. We address this issue by transforming a SMIL document to a specific form, referred to as the *smil normal form (smilNF)*, that reflects the semantic hierarchy and preserves the runtime semantics. As shown momentarily, it is structurally similar to the disjunctive normal form of a formula in propositional

*This work was partially supported by the National Science Foundation under grants CCS-0113515 and IIS-0237782.

logic. Consequently, we provide an algorithm to translate any such fragment to its SMIL normal form.

The rest of the papers is organized as follows. Section 2 describes related work. Section 3 describes the SMIL syntax. Section 4 describes different security paradigms, explains the problem with object identity and provides the generalized algorithm for conversion applicable to all paradigms. Section 5 gives a SMIL fragment in the MLS security paradigm, decorated with custom contracts definable within SMIL to enforce security and Section 6 describes the proposed RDF meta-structure for security and QoS metadata. Section 7 shows the details of decorating a SMIL documents with RDF specifications. Section 8 describes how a secure run-time may communicate to obtain SMIL formatted data from a secure server. Section 9 concludes paper.

2 Related Work

RDF is a W3C standard for representing metadata on the web. RDF has syntax for representing entities, their properties and relationships. RDF Abstraction and Syntax [KC03], and RDF Primer [MM03] specify metainformation representation formats. RDF Schema [BG03] is a general purpose schema language. SMIL has a RDF based metainformation module [Mic01], but is insufficient to specify security policies. In addition, Hayes et al. [Hay03] describes the semantic aspects of RDF. We use the RDF vocabulary defined to specify our metastructure. Independent of SMIL, Quality of Service (QoS) is an integral part of Multimedia. Wijesekera et al. [WS96] specify properties of quality metrics associated with continuous media and Gu et al. [GNY$^+$01] propose *HQML*, a language to negotiate some QoS parameters between multimedia clients and servers.

Rutledge et al. [RHO99] describe some SMIL applications. Several graphical interfaces such as [Bul98] exist to author SMIL documents. In addition, Sampaio et al. [SSC00] propose methodologies to verify the *semantic correctness* of SMIL documents.

We consider DAC(Discretionary), MLS(Multilevel Secure) and RBAC(Role Based) access control security models governing the display and access to SMIL formatted multimedia. DAC is used to control access by restricting a subjects's access to an object. In DAC, there is a direct relation between the subject and the object determined by the access privilege, usually given by access control lists. Sandhu et al [SS96] describe the principles and practices of RBAC systems. A uniform standard for RBAC [SFK00] is in the works and will soon be a standard. In RBAC the *role* of an user determines the subject's access privileges. All permissions granted are based on the role the subject plays with respect to the application. Multilevel security (MLS) [Low99, Osb] has been widely studied to ensure data confidentiality, integrity, and availability. MLS systems provide controlled information flow based on the security classification of the protection objects (e.g., data items) and subjects of the MLS system (e.g., applications running on behalf of a user). To provide information confidentiality, data is allowed to flow only from low security levels to higher security levels.

Damiani et al. [DdVPS00, DdVPS02] propose models for securing textual XML documents. In addition [DdV03] discuss feature protection of XML format images where the primary focus is controlled dissemination of sensitive data within an image. They propose an access control model with complex filtering conditions. This model uses SVG to render the map of a physical facility but does not address operational semantics.

Bertino et al. [BCF01, BCFM00], have developed Author-X, a Java based system to secure XML documents that supports access control policies at various granularities and user credentials. Author-X encodes security policies for a set of XML documents in an XML file referred to as the policy base. They permit both permissions and prohibitions. This feature enables the user to specify exceptions with ease as opposed to creating a set of XML documents and document type definitions (DTD's). There are conflict-resolution and default strategies to address over specification and under specification respectively. With respect to multimedia Bertino at al. [BHAE02] propose a security framework to model access control in video databases. Their objects are sequences of frames or identifiable objects within a frame. Their actions are viewing and editing. However they do not explain how objects with controlled accesses are released to semantics-aware runtimes.

Gabillon et al. [GB02] have suggested an alternative to Damiani et al. [DdVPS00, DdVPS02], where authorization rules related to a specific XML document are first defined on a separate authorization sheet (style sheet), and this sheet is then translated to an (eXtensible Stylesheet Language) XSL sheet. If a user requests access to the XML document then the (XSL Transforms) XSLT [14] processor uses the XSLT sheet to compute the view of the XML document with appropriate rights.

Kudo et al. [KH00]have proposed a methodology based on provisional authorization for document security that has helped in the standardization of XACL. Stoica et al. [SF02] present cover stories in the context of XML. Their aim is to hide the existence of non-permitted data from the naïve user. The motivation of the work is the need to provide secure releases of multilevel XML documents and corresponding DTD files where security sensitivity is not monotonically increasing along all paths originating from the node of the XML document. Authors provide techniques to modify an MLS/XML document to release non-sensitive data in a manner that is semantically correct and inference free. Gowadia et al [GF03] present an access control framework that provides flexible security granularity for XML documents. RDF statements are used to represent security objects and to enforce access control on XML trees and their associations.

The main difference between SMIL and other XML documents are the temporal synchrony and continuity of the latter. The process of retrieval without losing the sense of continuity and synchronization needs better techniques and algorithms which all of the above models do not completely address. Kodali et al. [KW02, KWJ03, KFW03] propose three different models for enforcing different security paradigms. A release control for SMIL formatted multimedia objects for pay-per-view movies on the Internet that enforces DAC is described in [KW02]. The cinematic structure consisting of acts, scenes, frames of an actual movies are written as a SMIL document without losing the sense of a story. Here access is restricted to the granularity of an *act* in a movie. A secure and progressively updatable SMIL document [KWJ03] is used to enforce RBAC and respond to traffic emergencies. In an emergency response situation, different roles played by recipients determine the media clips they receive. In [KFW03] an MLS application for secure surveillance of physical facilities is described, where guards with different security classification in charge of the physical security of the building are provided live feeds matching their level in the security hierarchy. The paper discusses operational semantics for chosen SMIL fragments, the algorithms for conversion into a *SMIL normal form* and their proof of correctness. This paper is an extended version of an earlier paper [KWF03], in which we introduced the concept of normal form and an RDF metastructure for multimedia access control.

3 SMIL: Synchronized Multimedia Integration Language

SMIL [Aya01] is an extension to XML developed by W3C to author multimedia presentations with audio, video, text and images to be integrated and synchronized into one presentation. The distinguishing features of SMIL over XML are the syntactic constructs for timing and synchronizing live and stored media streams with qualitative requirements. In addition, SMIL provides a syntax for spatial layout including non-textual and non-image media and hyperlinks. We do not address the latter aspects of SMIL in this paper, but explain those SMIL constructs that are relevant for our application.

SMIL constructs for synchronizing media are ⟨seq⟩, ⟨excl⟩ and ⟨par⟩. They are used to hierarchically specify synchronized multimedia compositions. The ⟨seq⟩ element plays its children one after another in sequence. ⟨excl⟩ specifies that its children are played one child at a time, without imposing any order. The ⟨par⟩ plays all children elements as a group, allowing parallel play out. For example, the SMIL specification ⟨par⟩ video src=camera1 ⟩ ⟨audio src=microphone1⟩⟨/par⟩ specify that media sources camera1 and microphone1 are played in parallel.

In SMIL, the time period that a media clip is played out is referred to as its *active duration*. For parallel play to be meaningful, both sources must have equal active durations. When clips do not have equal active durations, SMIL provides many constructs to make them equal. Some examples are begin (allows to begin components after a given amount of time), dur (controls the duration), end (specifies the ending time of the component with respect to the whole construct), *repeatCount* (allows a media clip to be repeated a maximum number of times). In addition, attributes such as *syncTolerance* and *syncMaster* controls runtime synchronization, where the former specifies the tolerable mis-synchronization (such as tolerable lip-synchronization delays) and the latter specifies a master-slave relationship between synchronized streams. In this paper we assume that children of ⟨par⟩ have the same active durations, and they begin and therefore end simultaneously.

An important construct that we use is ⟨switch⟩ allowing one to switch among many alternatives compositions listed among its components. These alternatives are chosen based on the values taken by some specified attributes. For example, ⟨switch⟩ ⟩seq ⟨ ⟨audio src="stereo.wav" systemBitrate>25⟩ ⟨audio src="mono.wav" systemBitrate < 25⟩ ⟩ /seq ⟨ ⟨/switch⟩ plays stereo.wav when the SMIL defined attribute systemBitrate is at least 25 and mono.wav otherwise. We use this construct to specify our sample application. In order to do so, we define a `customTest Attribute` that we call `customTestSecurity` taking values ("TS","S","UC"). The attribute indicates the security level of streams that can be top secret, secret or unclassified. We use these attributes in our sample SMIL document for our discussion in section 5. SMIL also requires that every application-defined attribute (custom attribute in SMIL terminology) have a title and a default value. It further has a special flag override that takes the value *hidden* or *visible*. When override takes the value hidden, the player is not allowed to change the value of the custom attributes. That feature is useful in specifying security attributes that are not to be altered by SMIL players.

4 Security Paradigms and Access Control Rules

Most security paradigms specify how *subjects* can access *objects*. The subject may be granted an access permission in DAC, but in MLS and RBAC a such granting is subjected to some constraints, usually expressed in the form of rules. This section formally define the security paradigms we use and the constraints associated with them.

Discretionary Access Control permits an action `a` to be invoked by a subject `s` on an object `o`. This permission is sometimes expressed by constructing an access control list containing appropriate triples `(s,o,±a)`.

The simplest Role-Based Access Control models has three entities (roles, users and privileges) and two associations, (subject-to-role and role-to-privilege assignments) among them. A subject may activate any authorized roles, and by doing so obtains all privileges assigned to the activated role. For each subject `s` let the set of active roles be given by $ActR(s)$, and $AuthR(s)$ be the set of roles permitted to be invoked by `s`. Then, the restriction that a user may activate only authorized roles can be stated as $ActR(s) \subseteq AuthR(s)$. Privileges (access permissions) of each particular role are based on objects defined in the normal form. That is, a given specification S in RBAC normal form is organized in a manner that all objects permitted to a role R_i are represented together. Then, we can define the access permissions of each role r as rToPer(r_i), where rToPer(r_i) consists (object, action) pairs. Then `(s,o,±a)` belongs to the access control matrix iff $ActR(s) \subseteq AuthR(s) \wedge \exists r \in ActR(s)(o,a) \in rToPer(r)$. An RDF structure consisting of the subjects, roles and (object, action) pairs and corresponding user-to-roles and roles-to-permissions is constructed to enforce the stated constraints in later sections.

In Multi Level Security each access permission is determined by the security clearance of the subject and the security classification of the accessed object. Security labels form a lattice structure with a partial order referred to as the *dominance relation* among the labels. Information flow between the security labels is controlled based on the security objectives. Assuming that our access

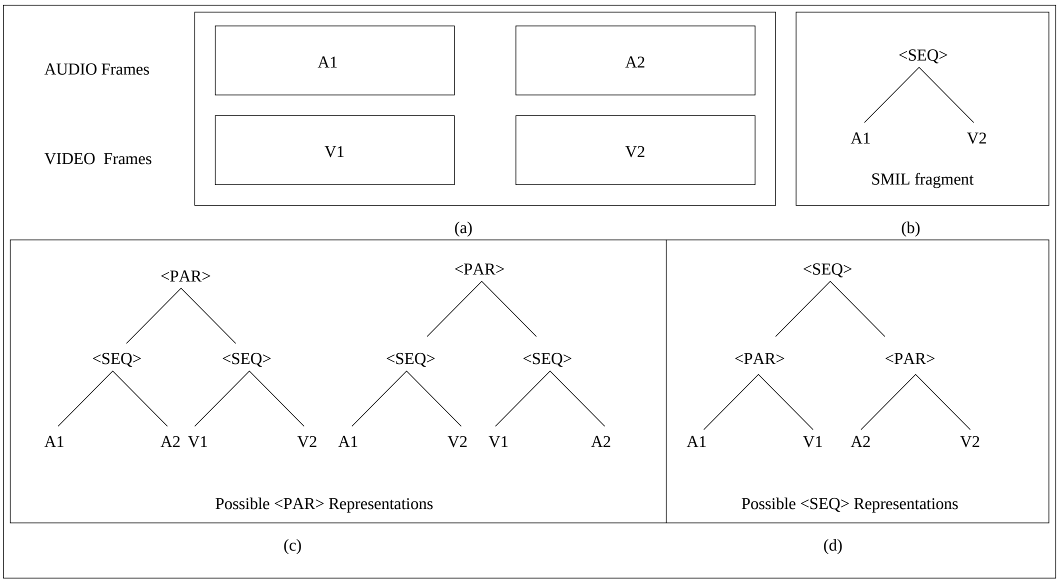

Figure 1: Equivalence Class of the SMIL Constructs

permissions are *read* permissions, the information flows from low security objects to high security objects, therefore a subject is allowed to access an object only if the subject's security clearance dominates the security classification of the object. To model the dominance relation, first we construct the transitive closure of dominance relations. Then, we use this closure to identify the security objects in the *normal form* of a specification S that are dominated by the security clearance of the subject. Let $Class(s)$ denote the classification of subject s. L denotes the lattice structure and binary relation $dominates(l_1, l_2)$, $l_1, l_2 \in L$ denotes that label l_1 dominates label l_2. To generate all labels dominated by the security classification a subject ($Class(s)$), we generate transitive closure of dominance relation as follows:
Let $Dominated(s) = \emptyset$ for all pairs $dominates(l_i, l_j)$, where $l_i = Class(s)$, $Dominated(s) = Dominated(s) \cup l_j$.

4.1 Identifying an Object in SMIL

SMIL uses ⟨par⟩ and the ⟨seq⟩ to specify parallel and sequential playout of multimedia streams. In SMIL, basic units are media intervals and a media interval begins at a specified time, plays out for a specified time and consequently ends at a specified time. That constitutes a rudimentary semantics for media intervals such as (audio) A_1 and (video) V_2 in Figure 1. In this semantics two streams are connected by a ⟨par⟩ if they begin and end playout at the same time. Two streams are connected by a ⟨seq⟩ if the second begins when the first ends. Consider Audio(A_1, A_2) and Video(V_1, V_2) frames as shown in part (a) of Figure 1, can be represented in SMIL in atmost three different ways using the ⟨par⟩ and ⟨ seq ⟩ constructs. Parts (c) and (d) of the Figure 1 represent the possible SMIL representation of the documents.

1. ⟨par⟩⟨seq⟩ A_1, A_2 ⟨/seq⟩ ⟨seq⟩ V_1, V_2 ⟨/seq⟩ ⟨/par⟩

2. ⟨par⟩ ⟨seq⟩A_1, V_2⟨/seq⟩ ⟨seq⟩ A_2, V_1 ⟨/seq⟩ ⟨/par⟩

3. ⟨seq⟩⟨par⟩A_1, V_1⟨/par⟩ ⟨par⟩ A_2, V_2 ⟨/par ⟩ ⟨/seq⟩

4. Because ⟨par⟩ is *commutative* ⟨par⟩ A_1, V_1 ⟨/par⟩ is the same as ⟨par⟩ V_1, A_1 ⟨/par⟩ and ⟨par⟩ A_2, V_2 ⟨/par⟩ is the same as ⟨par⟩ V_2, A_2 ⟨/par⟩.

Now consider the fragment ⟨seq⟩A_1, V_2 ⟨/seq⟩, as shown in part(b) is not a subtree of the given syntactic representations in part(d), but a sub-object of the SMIL tree. It is easy to identify or retrieve a subtree with appropriate queries because a sub-tree is always a single node or a group of nodes in a XML-tree. On the contrary when more than one sub-trees convey similar information, there is a need to capture all such semantically equivalent trees, which collectively form the *protection object*. The identity of the *protection object* therefore is not a node (or a group of connected nodes) in an XML tree, but an equivalence class, represented by it's *normal form*. To enable identifying such a class of semantically equivalent instances, we propose that every SMIL specification be transformed to a sequence of parallel compositions that we call the *smil normal form (smilNF)* given in definition 1 and show that all sub-objects of a SMIL object can be seen as a subtree (created from) of this form [KFW03].

Definition 1 (SMIL Normal Form) *We say that a SMIL specification(s) is in the SMIL Normal Form (smilNF) if it is of the following form ⟨seq⟩ ⟨par⟩ $C_{1,1}$(s) $C_{1,2}$(s) $C_{1,3}$ (s)... $C_{1,n}$(s) ⟨/par⟩ ... $C_{m,1}$(s) $C_{1,2}$(s) $C_{1,3}$ (s)... $C_{m,n}$(s)⟨ /par ⟩ ⟨ /seq ⟩ where $C_{i,j}$ are audio and/or video media intervals.*

We also propose that security and QoS policies be specified on SMIL specifications in smilNF, and not on arbitrary syntax trees - because as shown, syntactic substructure does not coincide with semantic inheritance in SMIL.

4.2 Secure Normal Forms

We allow SMIL documents in smilNF to be decorated to subjects, security levels and roles respectively. Then the final authorization triples `(s,o,±a)` can be derived using appropriate rules. Security decoration on the *protection objects* are defined on the normal form. We allow any node of a SMIL tree in smilNF to be decorated as shown in the Figure 2. Given such a decoration, we can compute a view that is permitted for each subject, security level or a role. They are referred to as *security normal forms*, and are formally stated in the generalized definition 2.

Definition 2 (Generalized Secure Normal Form) *We say that a smilNF specification ($\tilde{s}$) is in the secure normal form if it is of the form ⟨ seq ⟩ ⟨par⟩ $C_1(\tilde{s})$ ⟨/par⟩ ⟨par⟩ $C_2(\tilde{s})$ ⟨/par⟩ ⟨par⟩ C_3 ($\tilde{s}$)... $C_n(\tilde{s})$ ⟨ /par ⟩ ⟨ /seq ⟩ where $C_1, C_2, C_3 \ldots C_n$ are views corresponding to a permission in DAC, a role in RBAC and a security classification in MLS.*

The security normal form is a parallel composition of permitted segments. The smilNF specification is decorated with the metadata related to the particular security paradigm, upon reduction , would group all permitted segments of a particular subject under a single ⟨ par ⟩. Each of the ⟨ par ⟩ construct could be considered as the *view* of the associated subject. A normal form in RBAC is one that is a parallel composition of one or more role specifications that belong to a particular role assignment. Consequently, the generated output would have as many parallel compositions as the number of roles involved. Each subject could be granted access to multiple views, depending on the number of roles it is associated with. A normal Form in MLS is one that is a parallel composition of at most as many instances as the number of security classifications, where each instance belongs to one security class, and these instances are the views corresponding to the respective security classes.

4.3 Reduction to Secure Normal Forms

Algorithm 1 shows how to convert a SMIL specification in smilNF to any of the secure normal forms. The word *subject* is generalized and depends upon the security paradigm. DAC, either allows

Algorithm 1 toNF (A Generalized Algorithm for conversion to Secure Normal Forms)

```
    INPUT : The subjects in smilNF are security decorated, possible subjects $sub_1, sub_2, sub_3, \ldots sub_n$ . The generalized subjects are roles and security classification in RBAC and MLS respectively.
    OUTPUT : Secure Normal Form
Ensure: $(\tilde{s})$ is a smilNF specification (as described in Definition 1 ) with a decorated subject attribute,
    if $(\tilde{s})$ is $\langle$ seq $\rangle$ $s_1 s_2$ $\langle$ /seq $\rangle$ then
        $C_{sub_1}$ $(\tilde{s})$ = $\langle$ seq $\rangle$ $\langle$ par $\rangle$ $C_{sub_1}(s_1)$ $\langle$ /par $\rangle$ $\langle$ par $\rangle$ $C_{sub_1}(s_2)$ $\langle$ /par $\rangle$ $\langle$ /seq $\rangle$
        $C_{sub_2}$ $(\tilde{s})$ = $\langle$ seq $\rangle$ $\langle$ par $\rangle$ $C_{sub_2}(s_1)$ $\langle$ /par $\rangle$ $\langle$ par $\rangle$ $C_{sub_2}(s_2)$ $\langle$ /par $\rangle$ $\langle$ /seq $\rangle$
        $C_{sub_3}$ $(\tilde{s})$ = $\langle$ seq $\rangle$ $\langle$ par $\rangle$ $C_{sub_3}(s_1)$ $\langle$ /par $\rangle$ $\langle$ par $\rangle$ $C_{sub_3}(s_2)$ $\langle$ /par $\rangle$ $\langle$ /seq $\rangle$
        $\ldots C_{sub_n}$ $(\tilde{s})$ = $\langle$ seq $\rangle$ $\langle$ par $\rangle$ $C_{sub_n}(s_1)$ $\langle$ /par $\rangle$ $\langle$ par $\rangle$ $C_{sub_n}(s_2)$ $\langle$ /par $\rangle$ $\langle$ /seq $\rangle$
    else if $(\tilde{s})$ is $\langle$ par $\rangle$ $s_1 s_2$ $\langle$ /par $\rangle$ then
        $C_{sub_1}$ $(\tilde{s})$ = $\langle$ par $\rangle$ $C_{sub_1}(s_1)$ $\langle$ /par $\rangle$ $\langle$ par $\rangle$ $C_{sub_1}(s_2)$ $\langle$ /par $\rangle$
        $C_{sub_2}$ $(\tilde{s})$ = $\langle$ par $\rangle$ $C_{sub_2}(s_1)$ $\langle$ /par $\rangle$ $\langle$ par $\rangle$ $C_{sub_2}(s_2)$ $\langle$ /par $\rangle$
        $C_{sub_3}$ $(\tilde{s})$ = $\langle$ par $\rangle$ $C_{sub_3}(s_1)$ $\langle$ /par $\rangle$ $\langle$ par $\rangle$ $C_{sub_3}(s_2)$ $\langle$ /par $\rangle$
        $\ldots$ $C_{sub_n}$ $(\tilde{s})$ = $\langle$ par $\rangle$ $C_{sub_n}(s_1)$ $\langle$ /par $\rangle$ $\langle$ par $\rangle$ $C_{sub_n}(s_2)$ $\langle$ /par $\rangle$
    end if
    if either of $C_x(s_i)$ are empty for some x $\in\{sub_1$,$sub_2$, $\ldots$ $sub_n\}$ and i $\in\{1,2\}$ then
        $C_x(s_i)$ in the right hand sides above must be substituted by $\phi$ $(s_i)$ where $\phi$ $(s_i)$ is defined as $\langle$ audio/video src = empty$\rangle$
    end if
    The subject attribute could be a permission on the subject in DAC, a security classification if the security paradigm is MAC and a role the subject plays in RBAC.
    If Subject Attribute = $sub_1$, then $C_{sub_1}$ $(\tilde{s})$ = $(\tilde{s})$, $C_{sub_2}(\tilde{s})$= $\phi$, and $C_{sub_3} \ldots C_{sub_n}(\tilde{s})$= $\phi$
    If Subject Attribute = $sub_2$, then $C_{sub_2}$ $(\tilde{s})$ = $(\tilde{s})$, $C_{sub_1}$ $(\tilde{s})$ = $\phi$, and $C_{sub_3} \ldots C_{sub_n}$ $(\tilde{s})$ = $\phi$
    If Subject Attribute = $sub_n$, then $C_{sub_n}$ $(\tilde{s})$= $(\tilde{s})$,$C_{sub_1}$ $(\tilde{s})$ = $\phi$, $C_{sub_2}(\tilde{s})$= $\phi$, $\ldots$, $C_{sub_{n-1}}(\tilde{s})$ = $\phi$
    Then let NF $(\tilde{s})$= $\langle$ seq $\rangle$ $\langle$ par $\rangle$ $C_{sub_1}$ $(\tilde{s})$ $\langle$ /par $\rangle$ $\langle$ par $\rangle$ $C_{sub_2}$ $(\tilde{s})$ $\langle$ /par $\rangle$ $\ldots C_{sub_n}(\tilde{s})$ $\langle$ /par $\rangle$ $\langle$ /seq $\rangle$
```

or disallows access to a subject. In MAC the security classification of a subject should dominate that of the object . In RBAC a subject is granted or denied access depending upon it's role. Additionally, a subject is allowed to play more than one role, in which case it could to be allowed access to more than one view. During the rewrite, some of the nodes are represented as $\langle$ empty $\rangle$ indicating audio or video *silence* (null content). When grouping elements that satisfy a particular access control rule, there is a need to eliminate those that do not qualify. In other words, the disallowed elements should not be a part of generated view and consequently should not display at the client devices. Normally, a silent audio segment or a blank video segment are used to during playout to maintain continuity without losing the sense of security.

The Figure 2 shows the schematic reduction in all three security paradigms after applying Algorithm 1. In smilNF, the security decoration could be at three levels, the primary time container, the nested time container and the frame level. In our DAC example subject sub_1 is permitted access to the whole tree, where as subject sub_2 is granted access only to video frame V_2. The reduction uses $\langle$empty$\rangle$ to denote a contentless element. The views corresponding to sub_1 and sub_2 that when combined form the dacNF after the application of Algorithm 1 is shown on the right hand side. The first two components denotes the *view* of sub_1 and sub_2. In the MLS example the $\langle$par$\rangle$ is classified as Top-Secret and audio frame A_1 is also classified as Top-Secret. The video frame V_2 is classified as secret. The resulting views for Top-Secret and Secret are shown. The resulting mlsNF is a parallel composition of two security classifications, and the Top-Secret (higher classification) is allowed access to the Secret (lower) classification by the virtue its position in the classification hierarchy. Similarly a RBAC decorated smilNF with three roles r_1, r_2, r_3 and its reduced rbacNF is also shown.

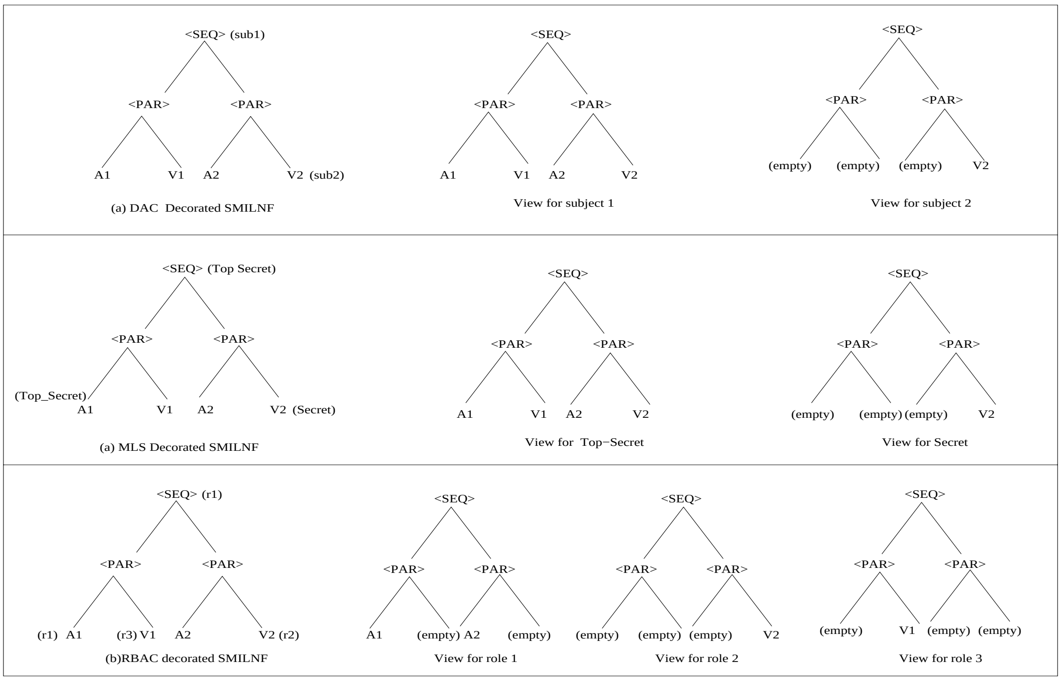

Figure 2: Reduction to dacNF, mlsNF and rbacNF

5 Specifying Security using SMIL Syntax

In this section we apply the concepts introduced in previous sections on a Multi level Secure SMIL fragment using only constructs provided by the SMIL specification [Aya01]. Because, the current specification does not have constructs defined for security and Quality-of-Service, we use Custom attributes to define them.

The SMIL fragment below consists of a switch statement consisting of collection of media streams connected by ⟨par⟩ constructs. In this example we have cameras and a microphones to record audio and video streams . They are named `CameraTS1.rm`, `CameraU1.wav` etc depending on the sensitivity of the information they capture. The security classification of each source is identified by the application defined SMIL attribute `customTestSecurity`. For example, ⟨video src="CameraTS1.rm" channel="video1" customTestSecurity="TS"/⟩ specifies that the video source named CameraTS1.rm has the Top Secret security level.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<customAttributesSecurity>
      <customTestSecurity id="TS" title="Top-Secret"
        defaultState="true" override="hidden"/>
      <customTestSecurity id="S" title="Secret"
        defaultState="true" override="hidden"/>
      <customTestSecurity id="UC"  title="Unclassfied"
        defaultState="true" override="hidden"/>
</customAttributesSecurity>
```

```
<body>
<switch>
<!------Classification is TS(Top-Secret)--->
<video src="CameraTS1.rm" channel="video1" customTestSecurity="TS"/>
<audio src="CameraTS1.wav" customTestSecurity="TS" />
<!------Classification is S(Secret)----->
<video src="CameraS1.rm" channel="video1" customTestSecurity="S"/>
<audio src="CameraS2.wav" customTestSecurity="S"/>
<!-------Classification is U(Unclassified)---->
<video src="CameraU1.rm" channel="video2" customTestSecurity="S"/>
<audio src="CameraU1.wav" customTestSecurity="S" />
</par>
 </body>
</smil>
```

Algorithm 2 TOmlsNF (Conversion to MLS Normal form)

```
INPUT : Security Classification decorated smilNF, possible classifications $l_1 \geq l_2 \ldots \geq l_n$.
OUTPUT : mlsNF
($\tilde{s}$) is a smilNF specification (as described in Definition 1 ) with a possible Security classification
if ($\tilde{s}$) is $\langle$ seq $\rangle$ $s_1 s_2$ $\langle$ /seq $\rangle$ then
    $C_{l_1}(\tilde{s}) = \langle \text{seq} \rangle \langle \text{par} \rangle C_{l_1}(s_1) \langle /\text{par} \rangle \langle \text{par} \rangle C_{l_1}(s_2) \langle /\text{par} \rangle \langle /\text{seq} \rangle$
    $C_{l_2}(\tilde{s}) = \langle \text{seq} \rangle \langle \text{par} \rangle C_{l_2}(s_1) \langle /\text{par} \rangle \langle \text{par} \rangle C_{l_2}(s_2) \langle /\text{par} \rangle \langle /\text{seq} \rangle$
    $\ldots C_{l_n}(\tilde{s}) = \langle \text{seq} \rangle \langle \text{par} \rangle C_{l_n}(s_1) \langle /\text{par} \rangle \langle \text{par} \rangle C_{l_n}(s_2) \langle /\text{par} \rangle \langle /\text{seq} \rangle$
else if ($\tilde{s}$) is $\langle$ par $\rangle$ $s_1 s_2$ $\langle$ /par $\rangle$ then
    $C_{l_1}(\tilde{s}) = \langle \text{par} \rangle C_{l_1}(s_1) \langle /\text{par} \rangle \langle \text{par} \rangle C_{l_1}(s_2) \langle /\text{par} \rangle$
    $C_{l_2}(\tilde{s}) = \langle \text{par} \rangle C_{l_2}(s_1) \langle /\text{par} \rangle \langle \text{par} \rangle C_{l_2}(s_2) \langle /\text{par} \rangle$
    $\ldots C_{l_n}(\tilde{s}) = \langle \text{par} \rangle C_{l_n}(s_1) \langle /\text{par} \rangle \langle \text{par} \rangle C_{l_n}(s_2) \langle /\text{par} \rangle$
end if
if either of $C_x(s_i)$ are empty for some x $\in \{l_1, l_2, \ldots l_n\}$ and i $\in$\{1,2\} then
    $C_x(s_i)$ in the right hand sides above must be substituted by $\phi$ ($s_i$) where $\phi$ ($s_i$) is defined as
    $\langle$ audio or video src = empty $\rangle$
end if
If Security classification =$l_1$, then $C_{l_1}(\tilde{s}) = (\tilde{s})$
If Security classification =$l_2$, then $C_{l_1}(\tilde{s}) = \phi$ ,$C_{l_2}(\tilde{s}) = (\tilde{s})$
. . . If Security classification=$l_n$, then $C_{l_1}(\tilde{s}) = \phi$ , $C_{l_2}(\tilde{s}) = \phi$ , . . . and $C_{l_n}(\tilde{s}) = (\tilde{s})$.
Then let mlsNF ($\tilde{s}$) = $\langle \text{seq} \rangle \langle \text{par} \rangle C_{l_1}(\tilde{s}) \langle /\text{par} \rangle \langle \text{par} \rangle C_{l_2}(\tilde{s}) \langle /\text{par} \rangle \langle \text{par} \rangle C_{l_n}(\tilde{s}) \langle /\text{par} \rangle \langle /\text{seq} \rangle$ .
```

Algorithm 2 shows how to obtain a secure MLS normal form. Although SMIL permits custom attributes, the addition of security of any kind to this form is a non-trivial task, because they are only valid within this fragment and are not generic in nature. Additionally, the regular SMIL interpreter that exists in the display devices of the recipient does not understand a custom security classification or a custom QoS attribute. To provide proper interpretation of these Custom attributes we have to define what they mean and how they are supposed to be understood by the client.

In effect, we would not be able to implement any of the theoretical results we obtained, unless (a) we are able to declare our security and QoS parameters as a integral part of the SMIL formatted document and (b) the recipient display devices are able to to interpret the document in its entirety along with its semantics. The proposed solution for this problem is to create a resource description framework as suggested by SMIL Metainformation Module [Mic01] using RDF [KC03] syntax that allows us to define a namespace with resources and their intended meaning thereby providing greater expressive power to specify security and quality restrictions on SMIL documents.

6 A Metastructure for Secure SMIL

The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. RDF does not stipulate semantics but rather facilitates communities to define metadata elements as needed. The RDF infrastructure enables metadata inter-operability through the design of mechanisms that support semantics, syntax, and structure. RDF uses XML (eXtensible Markup Language) as a common syntax for the exchange and processing of metadata. In RDF, resources have properties (attributes or characteristics). These properties serve both to represent attributes of resources (and in this sense correspond to usual attribute-value pairs) and to represent relationships between resources. Figure 3 represents the class hierarchy of the metadata we define in RDF for specifying security requirements. The words Class Hierarchy and metastructure are used interchangeably within this section.

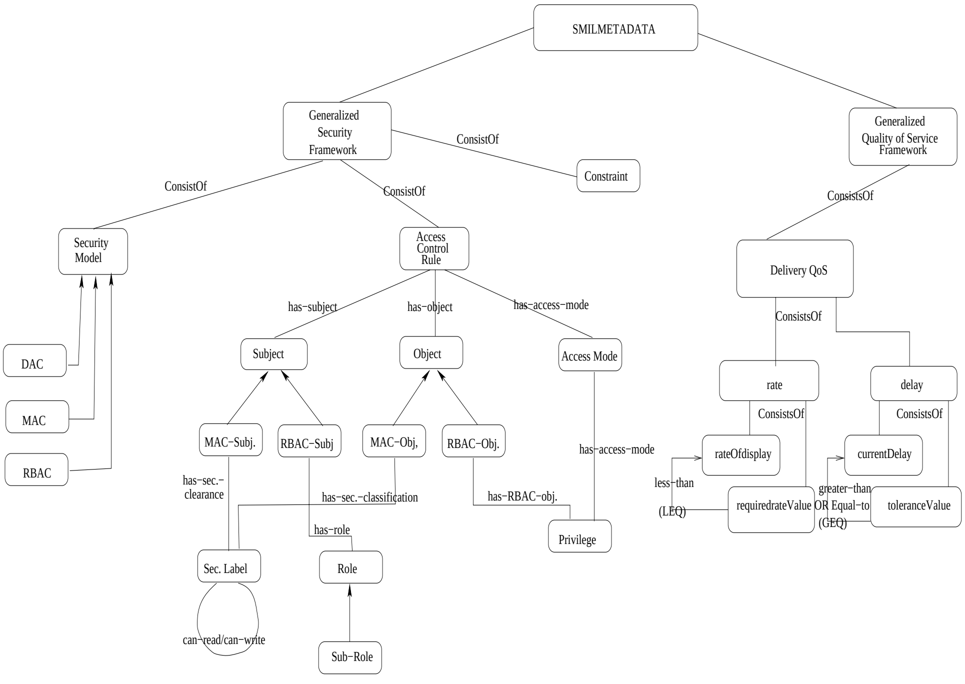

Figure 3: Security and QoS Class Hierarchy of the Proposed RDF-Metastructure

6.1 RDFS VOCABULARY

Generally the use of RDF to create a metastructure and its constituent metadata is application specific and requires the governing metastructure to enable metadata specific to the particular application. We address client-server interaction that transfers multimedia information that is a combination of audio, video and text. The namespace for the metastructure we created for our study is at http://svp.gmu.edu/smil-ns# and is referred to as *smilmetadata* within this paper. The structure represents metadata for both security and QoS. A statement consists of the combination of a Resource, a Property, and a value. These parts are known as the 'subject', 'predicate' and 'object' of a statement. The URI (Uniform Resource Identifiers) with optional fragment identifiers is used to describe subjects objects and predicates in statements and relationships between URI-identifiable entities. In the context of security, as we observe in Figure 3, we define the class hierarchy to represent the entities and the relations between them. The generalized security model consists of particular security model, access control rules and constraints (if any). The children of the security model are the three paradigms DAC, MLS and RBAC. The access control rule has three children subject, object and access mode stated as a `(s,o,±a)` triple as explained in Section 4. The subject and object have children corresponding to each security paradigm and are linked to classifications (MLS) and roles (RBAC) as needed. The access mode is generally +/- and encapsulates the privileges that can be obtained by virtue of association with a particular security paradigm. In the metastructure, Top-Secret, Secret and Unclassified are sub-classes of Class: MLS. The `rdf:subClassOf` property is transitive, implying that resources that are instances of `subClass` are implicit instances of the Class. The `rdf:domain` and `rdf:range` attributes available in RDF are used to define the scope of the members of a container with respect to a property of a class.

6.1.1 Security Metadata

All DAC, MAC, and RBAC models have been used in practice to ensure secure accesses to protected information. Our framework provides a transparent construction of all objects accessible to a subject, regardless of the security framework used. This section contains the metastructure defining security framework for our model and corresponding concepts. We focus on access control and provide interpretation for DAC, MAC, and RBAC models. The generalized security framework consists of description of *access control models*, *access control rules*, and *constraints* that further restrict accesses. The actual representation of metadata in SMIL-RDF is shown as samples below. There is a reference to the location of the namespace of the metastructure, which contains the class hierarchy and validates the correctness of the metadata in use for representation.

```
 xmlns: smilmetadata = http://svp.gmu.edu/AudioVideo/...../smilmetadata#
 ?xml version=1.0?
 <rdf:RDF xml:lang=xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
 xmlns:rdfs=http://www.w3.org/TR/2003/WD-rdf-schema-20030123/#>

<rdfs:Class rdf:ID="AC Models">
<rdfs:subClassOf rdf:resource="#Generalized Sec Structure"/>
<rdfs:subClassOf rdf:resource="#smilmetadata"/>
</rdf:Class>

<rdfs:Class rdf:ID="Disc Access Control">
<rdfs:comment>
Discretionary Access Control
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#AC Models"/>
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
</rdf:Class>
<!----------Similarly MAC and RBAC  ----->
```

Our metastructure is used to enforce access control in security decorated multimedia documents. We require that all objects are in their appropriate security normal forms. As mentioned earlier, our aim is to combine all access permissions for a user into a single `(s,o,±a)` triple. Access control triples are generated from DAC, MAC, and RBAC permissions. For each access control model we define basic concepts such as security label, and their relationships. There are three main components of an access control rule: *subject*, *object*, and *access mode*. Subjects and objects are further divided into MAC and RBAC subclasses, incorporating DAC subjects into the class *Subject* itself. In the current version of our ontology, *access modes* are explicitly defined, e.g., read, write, execute, but can be easily extended or substituted for future versions. For our application domain, we only need the read (retrieve) permission.

```
<!---Security Subjects and Objects----->

<rdf:Class rdf:ID="MAC-Subject">
<rdfs:subClassOf rdf:resource="#Subject"/>
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
</rdf:Class>

<rdf:Class rdf:ID="RBAC-Subject">
<rdfs:subClassOf rdf:resource="#Subject"/>
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
</rdf:Class>

<rdf:Class rdf:ID="MAC-Object">
<rdfs:subClassOf rdf:resource="#Object"/>
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
</rdf:Class>

<rdf:Class rdf:ID="RBAC-Object">
<rdfs:subClassOf rdf:resource="#Object"/>
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
</rdf:Class>
```

The last component of the security framework represent additional restrictions over access control rules and models. These constraints may correspond to well understood restrictions, like time dependent restrictions, separation of duties, session control, but may also represent restrictions on applicable security models, like accesses to highly critical military objects must have MAC classification.

```
<!---------------Access Modes --------->
 <rdf:Class rdf:ID="Access Mode">
 <rdfs:comment>
   Access mode, e.g., read, write, execute,
   defines the mode of access being
   permitted or denied.
</rdfs:comment>

<rdf:oneOf rdf:parseType="rdf:collection">
     <Access Mode rdf:ID="Read permitted"/>
    <Access Mode rdf:ID="Write permitted"/>
    <Access Mode rdf:ID="Execute permitted"/>
    <Access Mode rdf:ID="Read denied"/>
    <Access Mode rdf:ID="Write denied"/>
    <Access Mode rdf:ID="Execute denied"/>
  </rdf:oneOf>
</rdf:Class>
```

6.1.2 QoS Metadata

The DeliveryQoS class and its related sub-classes define metadata pertaining to multimedia QoS form the server to recipient devices. The delivery factors we consider are *rate* and *delay.* In the service level agreement (SLA) between the server and the clients, threshold values for the expected rate and tolerable delay are contracted. The metadata should enable the conformance to such a contract by providing means of enforcement and negotiation. The threshold values are represented by the sub-classes *requiredRateValue* for the rate, and *toleranceValue* for delay. Constraints *greaterTHANORequal* and *lessTHANORequal* are defined to relate the current values to the threshold values and enforce conformance.

```
<rdf:Class rdf:ID="DeliveryQoS">
  <rdf:disjointWith rdf:resource="#SystemQoS"/>
  <rdfs:subClassOf rdf:resource="#QoS Struct"/>
</rdf:Class >

 <rdf:Class rdf:ID="rate" >
  <rdfs:subClassOf rdf:resource="SystemQoS"/>
  <rdfs:subClassOf rdf:resource="#QoS Struct"/>
 </rdf:Class>

 <rdf:Class rdf:ID="delay">
  <rdfs:subClassOf rdf:resource="SystemQoS"/ >
  <rdfs:subClassOf rdf:resource="#QoS Struct"/>
</rdf:Class>
```

7 Specifying Security using RDF-Metadata in SMIL

RDF metadata is needed because of the lack of expressibility using SMIL and its inherent Custom attributes. As explained earlier, metadata that is understood by a SMIL interpreter would enable us do defined subjects objects and their relation in the context of security. The concepts of *Role* , *privilege* , *constraint* are not in the current SMIL specification but can be expressed using our metamodel. This section describes how the designed metastructure is to be used within a SMIL fragment. Assuming that the SMIL document is in the SMIL normal form (smilNF)the *smilmetadata* structure that has been defined earlier is utilized for the RDF metadata for namespace references. The Title, Description, Publisher, Date, Rights and Format are from the Dublin Core URI that identifies these as standard descriptors. The QoS parameters are sometimes categorically stated in the head section. This is to enable these parameters to be negotiated initially with the display device, even before the body of the SMIL document is interpreted so that if they do not evaluate to TRUE, the document is rejected.

```
?xml version="1.0" ? smil xmlns ="http://svp.gmu.edu/SMIL-2.0.dtd"
<head>
    <metadata id="Maabaavamanchivaadunaakumaabaavaantechaalaistam">
    <rdf:RDF>
    xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs = "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#’’
    xmlns:dc = "http://purl.org/metadata/dublin\_core#"
    xmlns: smilmetadata  = "http://aparna.gmu.edu/AudioVideo/.../smil-ns#"

 !-- Metadata about the Media --
```

```
    <rdf:Description about="http://prathibha.gmu.edu/ smilmetadata ">
        dc:Title="A Secure QoS Aware Live Media Feed"
        dc:Description=" Metastructure for QoS aware Secure SMIL"
        dc:Publisher="Thibha"
        dc:Creator="Rajit"
        dc:Date="2004-06-03"
        dc:Rights="Copyright 2003 Bunty"
        dc:Format="text/smil"
    </rdf:Description>

   <smilmetadata :delay>
   <smilmetadata :rateOfDisplay>
  <rdf:Alt ID="Synchronization Parameters.">
  <rdf:li> requiredRateValue = "25Kbps" toleranceValue = "4"</rdf:li>
  <rdf:li> requiredRateValue = "30Kbps" toleranceValue ="3" </rdf:li>
  </rdf:Alt>
   </smilmetadata :rateOfDisplay>
   </smilmetadata :delay>
 </head>
    </metadata>
    </rdf:RDF>
```

The attributes requiredRateValue and toleranceValue have been assigned values in this example to show their usage and for QoS negotiation via challenge response, because the actual transfer is dependent on the result of the negotiation. This mechanism will enable users to enforce QoS restrictions prior to movement of sensitive data.

```
<!-------MLS Security Metadata within a SMIL Document------>
   <body>
     <smilmetadata :MLS>
       <par id="shot3"  smilmetadata : Top-Secret>
          video src="shot3.mpg"
          audio src="shot3.au"
       </par>
     <par id="shot4">
          video src="shot4.mpg"
          audio src="shot4.au" smilmetadata :Unclassified
       </par>
      </smilmetadata :MLS>
    </body>
```

The example above shows an MLS decorated smilNF. The ⟨par⟩ in shot 3 is Top-Secret and the audio frame of shot 4 is Unclassified. The evaluation of the SMIL document in runtime requires a sufficient semantic query model and an efficient interpreter to understand and interpret the RDF metadata used to declare security and QoS.

```
<!-------RBAC Security Metadata within a SMIL Document------>
   <body>
     <smilmetadata :RBAC>
        <par id="shot1">
            video src="shot1.mpg"  smilmetadata: role_1
            audio src="shot1.au"
        </par>
        <par id="shot4"  smilmetadata:role_3>
            video src="shot4.mpg"
            audio src="shot4.au"
        </par>
    <smilmetadata :RBAC>
  </body>
```

The security decoration in the SMIL fragment shown above belongs to the RBAC security paradigm. The video frame of shot 1 is allowed for $role_1$ and the entire parallel composition in shot 4 is allowed for $role_3$.

8 Runtime Operations

Our metastructure can be used by a multimedia client that seeks SMIL documents with proposed RDF decorations. Our client must use an RDF based query system for this purpose to generate views for DAC, MLS and RBAC. The RDF Query [MS98] uses a declarative syntax for selecting RDF resources that meet specified criteria. For example, for RBAC retrieval, we show how to construct a RDF query to retrieve the view for a given role. Similarly, we show an example query to retrieve all objects corresponding to particular security classification. An RDF-Interpreter is necessary to understand and assemble a SMIL view from a RDF decorated SMIL document that is to be interpreted by a SMIL player at the client. Although we do not provide such an interpreter, our client needs to have two interacting interpreters, where the SMIL-Interpreter calls the RDF-Interpreter.

As stated in Section 4, all DAC, MLS and RBAC can be reduced to the access control rule stated as a simple (s: subject, o: object, a:access). Therefore the access control rule is defined as a 4 tuple (c,o,d,a) where C is a condition expressed in RDF Query and is representative of the generalized subject, o is the security object (Normal Form), d is the decision to grant or deny and a is the type of access. An example of RDF Query [MS98] for the RBAC and MLS security paradigms are discussed in Section 8.1 and 8.2. The conditions use SQL keywords such as *select*, *from* etc. Complex and nested queries could be formulated with the use of boolean expressions.

8.1 An RBAC Query

This query represented below retrieves the view pertaining to a single role ($role_1$) from the rbacNF. The scope of the RBAC query is the RBAC Normal form. The structure of rbacNF guarantees that media components associated with the particular role is grouped together, and hence the retrieval could be based on the metadata used to define the particular role assignment. The RBAC query in section 8.1 would `select` components associated with *smilmetadata*: $role_1$ `from` the specified URI for the location of the rbacNF.

```
<rdfq:rdfquery>
<rdfq:From eachResource="http://svp.gmu.edu/AudioVideo/smil-ns #rbacNF">
    <rdfq:Select>
        rdfq:Propertyname="role_1"
    </rdfq:Select>
</rdfq:From>
</rdfq:rdfquery>
```

8.2 MLS Query

The query below retrieves the view pertaining to a specified security classification within a MLS Normal Form. The scope of the MLS query is the mlsNF represented by the appropriate URI. The MLS query in section 8.2 would *select* components associated with *smilmetadata* :Top-Secret *from* the specified URI that denotes the location of the mlsNF.

```
<rdfq:rdfquery>
<rdfq:From eachResource="http://svp.gmu.edu/AudioVideo/smil-ns#mlsNF">
    <rdfq:Select>
        <rdf:ID> Top-Secret </rdf:ID>
    </rdfq:Select>
</rdfq:From>
</rdfq:rdfquery>
```

Negotiating QoS parameters is the first step of the run-time operation. Unavailability of required QoS or non-conformance to the (SLA) would result in the terminating the media transfer. Once the query answer is obtained, the access control policy is evaluated and if access is granted the associated action (grant/deny) is initiated. Views could be encrypted to enforce integrity and unwanted mediate stream acquisition and guarantee unforgability. Several encryption techniques, such as the ones suggested in [KWJ03, KW02] can be used.

9 Conclusions

We presented a RDF metastructure to specify access control policies for multimedia documents. Our metastructure can enforce discretionary, mandatory and role based access control paradigms while respecting continuity and synchronization semantics of multimedia. We proposed a run-time that uses RDF and SMIL queries to securely retrieve documents decorated as specified by us. We are building our way through the upper layers of the Semantic Web including DAML+OIL [CHH01], OWL [DC03], RuleML [BTW01] and the most recent SWRL [HST^{+}03] to enforce security in SMIL-formatted multimedia documents.

References

[Aya01] Jeff Ayars. *Synchronized Multimedia Integration Language.* W3C Recommendation, 2001. http://www.w3.org/TR/2001/REC-smil20-20010807.

[BCF01] Elisa Bertino, Silvana Castano, and Elena Ferrari. Securing xml documents with author-x. *IEEE Internet Computing*, 5(3):21–31, 2001.

[BCFM00] Elisa Bertino, Silvana Castano, Elena Ferrari, and Marco Mesiti. Specifying and enforcing access control policies for xml document sources. *World Wide Web*, 3(3):139–151, 2000.

[BG03] Dan Brickley and R.V. Guha. *RDF Vocabulary Description Language 1.0:RDF Schema.* W3C Working Draft, January 23 2003. http://www.w3.org/TR/2003/WD-rdf-schema-20030123.

[BHAE02] Elisa Bertino, Moustafa Hammad, Walid Aref, and Ahmed Elmagarmid. An access control model for video database systems. In *Conferece on Information and Knowledge Management*, 2002.

[BTW01] Harold Boley, Said Tabet, and Gerd Wagner. Design rationale of ruleml: A markup language for semantic web rules. In *SWWS, Stanford*, 2001.

[Bul98] David Bulterman. Grins: A graphical interface for creating and playing smil documents. In *Proc. of Seventh Int'l World Wide Web Conf. (WWW7)*. Elsevier Science, New York, April 1998.

[CHH01] Dan Connoly, Frank Harmelen, and Ian Horrocks. *DAML+OIL Reference Description.* W3C Note, 2001. http://www.w3.org/TR/daml+oil-reference.

[DC03] Mike Dean and Dan Connolly. *OWL Web Ontology Language Overview*, 31st March 2003.

[DdV03] Ernesto Damiani and Sabrina De Capitani di Vimercati. Securing xml based multimedia content. In *18th IFIP International Information Security Conference*, 2003.

[DdVPS00] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. Securing XML documents. *Lecture Notes in Computer Science*, 1777:121–122, 2000.

[DdVPS02] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. A fine grained access control system for xml documents. *ACM Transactions on Information and System Security*, 5, 2002.

[GB02] Alban Gabillon and Emmanuel Bruno. Regulating access to xml documents. In *Proceedings of the fifteenth annual working conference on Database and application security*, pages 299–314. Kluwer Academic Publishers, 2002.

[GF03] Vaibhav Gowadia and Csilla Farkas. Rdf metadata for xml access control. In *Proceedings of the 2003 ACM workshop on XML security*, pages 39–48. ACM Press, 2003.

[GNY$^+$01] Xiaohui Gu, Klara Nahrstedt, Wanghong Yuan, Duangdao Wichadakul, and Dongyan Xu. An xml-based quality of service enabling language for the web, 2001.

[Hay03] Patrick Hayes. *RDF Semantics.* W3C Working Draft, January 23 2003. http://www.w3.org/TR/2003/WD-rdf-mt-20030123.

[HST$^+$03] Ian Horrocks, Patel Schneider, Said Tabet, Benjamin Grossof, Harold Boley, and Mike Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* W3C Working Draft, January 23 2003. http://www.w3.org/TR/2003/WD-rdf-mt-20030123.

[KC03] Graham Klyne and Jeremy Carroll. *Resource Description Framework(RDF) Concepts and Abstract Syntax.* W3C Working Draft, January 23 2003. http://www.w3.org/TR/2003/WD-rdf-concepts-20030123.

[KFW03] Naren Kodali, Csilla Farkas, and Duminda Wijesekera. Enforcing integrity in multimedia surveillance. In *IFIP 11.5 Working Conference on Integrity and Internal Control in Information Systems*, 2003.

[KH00] Michiharu Kudo and Satoshi Hada. Xml document security based on provisional authorization. In *Proceedings of the 7th ACM conference on Computer and communications security*, pages 87–96. ACM Press, 2000.

[KW02] Naren Kodali and Duminda Wijesekera. Regulating access to smil formatted pay-per-view movies. In *Proceedings of the 2002 ACM workshop on XML security*, pages 53–60. ACM Press, 2002.

[KWF03] Naren Kodali, Duminda Wijesekera, and Csilla Farkas. Multimedia access control using rdf metadata. In *Workshop on Metadata Security*, 2003.

[KWJ03] Naren Kodali, Duminda Wijesekera, and J.B.Michael. Sputers: A secure traffic surveillance and emergency response architecture. In *In submission to the Journal of Intelligent Transportaion Systems*, 2003.

[Low99] Gavin Lowe. Defining information flow, 1999.

[Mic01] Thierry Michel. *The SMIL 2.0 MetaInformation Module*. W3C Recommendation, 2001. http://www.w3.org/TR/2003/WD-rdf-mt-20030123.

[MM03] Frank Manola and Eric Miller. *RDF Primer*. W3C Working Draft, January 23 2003. http://www.w3.org/TR/2003/WD-rdf-primer-20030123.

[MS98] Ashok Malhotra and Neel Sundaresan. *RDF Query Specification*. W3C Specification, December 03 1998. http://www.w3.org/TR/2003/WD-rdf-primer-20030123.

[Osb] Sylvia Osborn. Mandatory access control and role-based access control revisited. pages 31–40.

[RHO99] L. Rutledge, L. Hardman, and J. Ossenbruggen. The use of smil: Multimedia research currently applied on a global scale, 1999.

[SF02] Andrei Stoica and Csilla Farkas. Secure xml views. In *Proc IFIP 11.3 Working Conference on Database Security*, 2002.

[SFK00] Ravi Sandhu, David Ferraiolo, and Richard Kuhn. The NISI model for role-based access control: Towards a unified standard. In *ACM RBAC 2000*, pages 47–64, 2000.

[SS96] Ravi Sandhu and Pierangela Samarati. Access control: Principles and practices. *IEEE Communications*, 29(2):38–47, 1996.

[SSC00] Paulo Nazareno Maia Sampaio, C. A. S. Santos, and Jean-Pierre Courtiat. About the semantic verification of SMIL documents. In *IEEE International Conference on Multimedia and Expo (III)*, pages 1675–1678, 2000.

[W3C03] *World-Wide-Web Consortium*, 31st July 2003.

[WS96] Duminda Wijesekera and Jaideep Srivastava. Quality of service (qos) metrics for continuous media. *Multimedia Tools and Applications*, 3(2):127–166, 1996.