International Journal on Digital Libraries manuscript No.
(will be inserted by the editor)

An Authorization Model for Multimedia Digital Libraries*

Naren Kodali[1], Csilla Farkas[3,4] and Duminda Wijesekera[1,2]

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

The date of receipt and acceptance will be inserted by the editor

Abstract. In this paper we present a *generalized authorization model* for digital libraries. Our aim is to support the enforcement of access control requirements of the original data sources without without the need to create a new, unified model for the digital library. We integrate the three most widely used access control models (i.e., Mandatory, Discretionary, and Role-Based Access Control) within a single framework, allowing seamless accesses to data protected by these security models. In particular, we address the access control needs of *continuous multimedia data*, while supporting Quality of Service (QoS) requirements, and preserving operational semantics.The technical core of the paper focuses on the development of *metadata* and corresponding *metastructure* to represent authorization policies and QoS requirements, and show their applicabilty to continuous multimedia. More specifically, we define our security objects based on the Synchronized Multimedia Integration Language (SMIL), that controls multimedia presentation. Following the synchronization constructs <par> and <seq> of SMIL, we define a normal form for multimedia streams, called *SMIL Normal Form*. SMIL Normal Form provides a syntax independent representation of semantically equivalent multimedia data. SMIL Normal Form compositions are extended (decorated) with RDF statements, representing security and QoS metadata. Interpretation of these statements, and therefore the authorization and QoS requirements of the decorated multimedia object, is defined by the metastructure, represented as a DAML+OIL ontology. We propose the concept of *generalized subject* that encompasses all access permissions of a given user, regardless of the multiple permissions in different access control models. Finally, we develop methods to generate secure views for each generalized subject and retrieve them using a secure multimedia server.

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

1 Introduction

Digital libraries support a wide variety of applications, ranging from educational and research activities to government and private sector applications. Digital library research focuses mainly on digital library design and efficient data manipulation for providing library services, with minimal or no consideration of data security. While a fundamental aspect of digital library design is ensuring open access [MG01], the increased dependence of a variety of applications on digital libraries, and privacy and copyright requirements raise the need to develop security models.

Existing works for digital library access control presented in [Lag95, Arm00, BHAE02, BFP02, AABF02] assume the existence of a uniform authorization model for the digital library. For example in [AABF02] Adam et al. discuss a content-based authorization model for digital libraries, where access decisions are evaluated based on the users' credentials, and security object identities and their content. In [BFP02] Bertino et al. describe MaX, an access control system based on user credentials and data content. They define a set of privileges, i.e., browsing and authoring. Access control policies are defined by a tuple over credential specification, entity specification (content), privilege, and a sign, to represent permission or denial. While these models seem expressive to incorporate the most widely used access control models, i.e., discretionary, mandatory, and role-based access control, they require the construction of a uniform access control model. To satisfy the original access control needs, it is crucial that each original access control model is mapped accurately to the unified model. Due to the large number of data stored in digital libraries with heterogeneous

access control requirements, this approach is error prone and cumbersome.

In this paper, we present a different approach, allowing each data object, to remain under the protection of the original access control. Our main aim is, to guarantee the original security needs of the data object, while allow transparent accesses to them, regardless of the applied security model. We develop a generalized security framework to represent Discretionary (DAC), Mandatory (MAC), and Role-Based Access Control (RBAC) models. We propose the concept of *generalized subject*, that is the combined capabilities of a user from different access control models (i.e., DAC, MAC, or RBAC). Our framework provides transparent retrieval of data by transforming all capabilities of a subject into *(s, o, a)* triples, where s is the subject (acting on behalf of the user), o is the security object, and a is the access permitted on object o to subject s. In this paper we focus on multimedia retrieval only.

We also develop a security ontology (metastructure). This ontology provides interpretation of the metadata attributes of the data objects and enables the enforcement of heterogeneous access control requirements. We use DAML+OIL (Darpa Agent Markup Language + Ontology Inference Layer) to represent our metastructure.

We show the applicability of our framework for multimedia digital libraries. Digital libraries for multimedia data, including continuous data, is a rapidly emerging field with critical applications like surveillance and remote video-audio conferencing. In addition to the security requirements, multimedia digital libraries also require the preservation of operational semantics and Quality of Service (QoS) requirements for continuous multimedia data. We use SMIL [Aya01], an XML-like language for authoring multimedia documents, to define protection objects, and to represent access control and QoS requirements. SMIL composition operators ⟨seq⟩ (sequential) and ⟨ par⟩ (parallel) define a rudimentary semantics of multimedia documents by controlling the presentation timing of multimedia frames. Our goal is to satisfy these timing constraints, thus preserve the operational semantics of multimedia presentation, while enforcing access control requirements. Due to the properties of ⟨seq⟩ and ⟨par⟩, several syntactic representations are possible for the same operational semantics (see Figure 2). Access control models that are based on the syntactic structure of the protected objects, like the current XML access control models [DdVPS00, DdVPS02, DdV03, BHAE02, SF02], are not sufficient to resolve this conflict of multiple syntactic representation of semantically equivalent object.

We address this problem by developing techniques to transform any SMIL document into a specific syntactic form, called SMIL normal form, while preserving run time semantics. *SMIL normal form (smilNF)* allows to represent SMIL documents in a syntactically consistent manner, any two, semantically equivalent SMIL representation will result in syntactically same smilNF. We use the *smilmetadata* attribute to represent security and QoS restrictions over smilNF. Similarly to the interpretation of the security metadata, we develop an ontology to describe the QoS metadata. We generate security views of these transformed documents.

Finally, we address the need of developing retrieval methods for this new multimedia data representation. Our aim is to provide service that satisfies the access control and QoS requirements, while preserves operational semantics. We build upon existing query languages for multimedia retrieval, while using the metadata and metastructure information to enforce the requirements.

The rest of the paper is organized as follows. In Section 2 we give a brief overview of related work. Section 3 gives a brief introduction of SMIL. Section 4 discusses the problem of object identity in SMIL and the normal forms. Section 4.2 presents the generalized subject and Section 5 defines the secure normal forms of different security paradigms. The security and the QoS metastructures are defines in Section 6, and Section 7 gives an example of the usage of the metadata in SMIL. Section 8 contains runtime operation and the query language to manipulate the multimedia data. We conclude and recommend future research in Section 9

2 Related Work

2.1 Secure Digital Libraries

Research and applications for digital libraries have increased during the last years. The main issues of these works to provide standards and technologies for digital library development and to support library services (see [Arm00] for an overview). Recently, the need to develop security models and mechanisms, that are applicable to digital libraries has emerged [Lag95, Arm00, BHAE02, BFP02, AABF02]. These works address storage media security, development of authorization framework and enforcement mechanisms. However, they assume the existence of a uniform access control model for the digital library. However, this requirement not only increases the risk of misclassification of data but also costly and time consuming. Also, to our best knowledge, none of these works address the security needs and semantic requirements of continuous multimedia data.

2.2 Access Control Models

The three most widely implemented access control models are Discretionary (DAC), Mandatory (MAC), and Role-Based Access Control (RBAC) [SS96, SFK00]. DAC allows database owners (or security administrators) to define access permissions on a per user-based manner. That is, there is a direct relation between the subjects

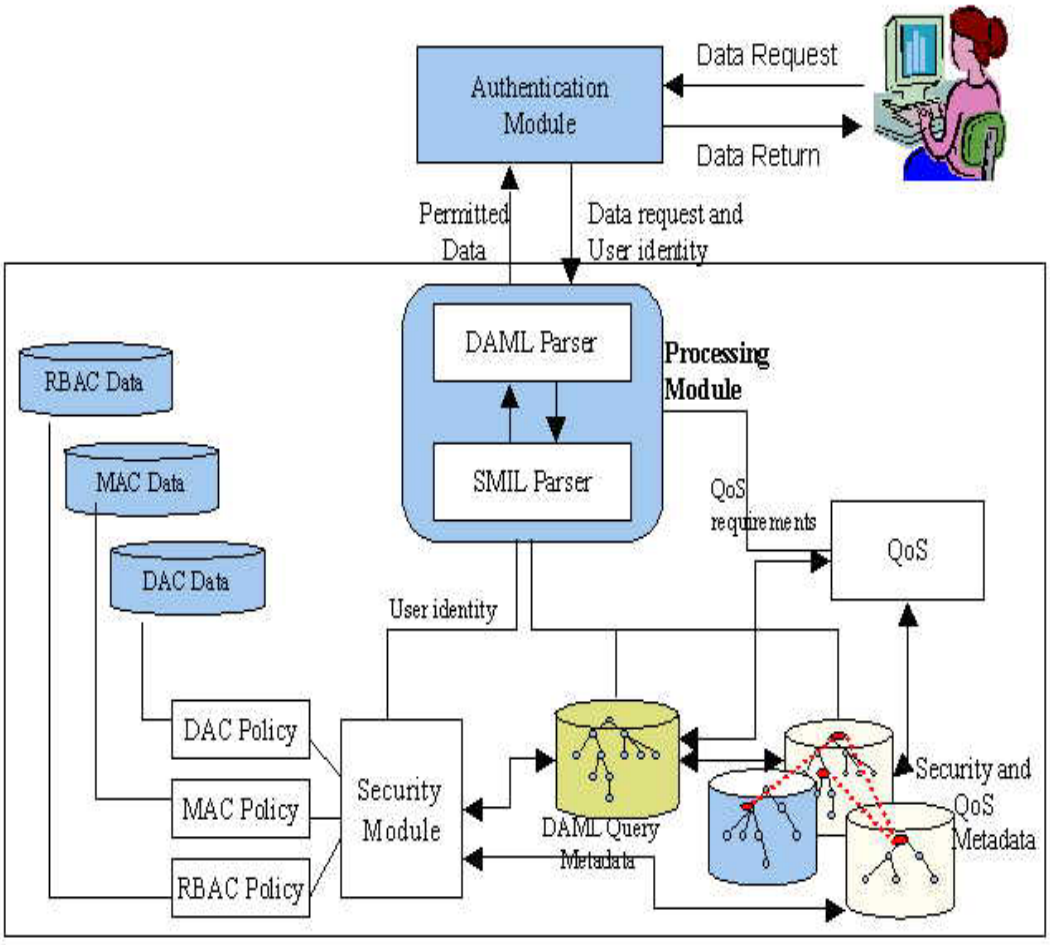

Fig. 1. Architecture of the Authorization Model for Multimedia Digital Libraries

(users) and the objects (data items) determined by the access privilege, like given by an access control lists. MAC policies provide controlled information flow between security layers based on the security labels of the protection objects and subjects. Security labels form a lattice structure with the dominance relation between the labels. To provide information confidentiality, data is permitted to flow only from a dominated security label to a dominating security label. In RBAC the *role* that a user plays within the organization determines his/her access privileges. Privileges are assigned to roles, and users are assigned to roles. A user is allowed to activate any assigned roles and the corresponding privileges within a session.

2.3 Semantic Web Technologies

Our authorization model is built upon existing technologies to provide data representation and integration on the Semantic Web [BLHL01]. In particular, we use XML, RDF, SMIL, and DAMl+OIL technologies. The eXtensible Markup Language (XML), a subset of Standard Generalized Markup Language (SGML) has become widely used to provide efficient way for information exchange. Built upon the XML syntax, Resource Description Framework (RDF) [KC03,MM03] provides enhanced semantic context by allowing to define entities, their properties and relationships. RDF Schema [BG03] is a general purpose schema language. Hayes et al. [Hay03] and Patel-Schneider et al. [PSS02] describes semantical aspects of RDF. Kodali et al. [KFW03c] use RDF vocabulary to specify a secure metastructure for multimedia libraries in different security paradigms.

The developments of ontologies provide interoperation among different applications. Current research on DARPA Agent Markup Language (DAML) aims to link information on the Web to domain ontologies. Similarly, Ontology Inference Layer (OIL) integrates ontologies with Web-standards like RDF and XML. The latest release of DAML [DAML+OIL] provides constructs to define and integrate ontologies with information resources. The specification of DAML with respect to semantics is explained by Horrocks [Hor02] , and Ankolekar et al. [AHS02]. The aspects of security in DAML have been discussed by Denker [Den02] with respect to DAML Web services described in DAML-S [Mar03]. A query specification for DAML represented ontologies has been proposed by Anyanwu et al. [AS03]. The DAML Query

Language [FHH03, FHH02] is a formal language that enables query-answer interaction between the server and the client agents. A DQL query is a query pattern written in DAML+OIL. Literals in the patterns are specified as unbound variables. The answer to the query is constructed by finding mappings to these variables such that the query is satisfied.

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. 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 spatially laying out media, including non-textual and non-image media and hyperlinks. We do not address latter aspects of SMIL in this paper. Consequently we explain those SMIL constructs that are relevant for our current application.

SMIL constructs for synchronizing media are ⟨seq⟩, and ⟨par⟩. They are used to hierarchically specify synchronized multimedia compositions. The ⟨seq⟩ element plays its children one after another in sequence. 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 equate them. 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 active durations.

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, KFW03a] 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 recipients of the live feeds have to be discriminated to people playing different roles. The paper describes a mechanism to enforce RBAC policies. [KFW03a] describes an MLS application for secure surveillance of physical facilities where guards with different security classification in charge of the physical security of the building are provided live feeds matching their level in the MLS subject hierarchy.

4 Generalized Access Control Framework

In this paper we present a general access control framework that accommodates discretionary, mandatory, and role-based access control models. Figure 1 shows our access control architecture and its components. We present transformations that enable to compute all permitted accesses to a user, regardless of the access control models used to define these requirements. For this, we need to specify the *objects* (media intervals) and *subjects* (users). We permit that a user may have multiple security clearances, however, security objects are classified according to a single security model. However, the digital library contains a collection of objects where different objects may be guarded by different access control models. The following sections define our security objects and subject, and the access control granularity.

4.1 Security Object Identity in SMIL

Unlike XML for textual documents, SMIL constructs have *intended meanings* (i.e., presentation restrictions) that must be enforced at run time. SMIL uses ⟨ par⟩ and the ⟨ seq⟩ to specify parallel and sequential presentation of multimedia streams. In SMIL, the basic objects are media intervals constructed from frames and timing constraints. A media interval begins at a specified time, plays for a specified duration, and ends at a specified time. This time-dependent presentation of the media intervals constitutes a rudimentary semantics. Consider the audio A_1, A_2, and video V_1,V_2 intervals with equal duration shown in Figure 2.a. SMIL constructs ⟨ par⟩ and ⟨ seq⟩ can be used to ensure that A_1 and V_1 and, similarly, A_2 and V_2 are played together and presentation of A_2, V_2 follows the presentation of A_1, V_1. A possible representation of this requirement using SMIL is ⟨

par⟩ ⟨ seq⟩ A_1, A_2 ⟨ /seq⟩ ⟨ seq⟩ V_1, V_2 ⟨ /seq⟩ ⟨ /par⟩ (see left tree of Figure 2.c. Assume that this syntactic form is used to enforce presentation constraints on the multimedia document.

However, several different syntactic representations of audio(A_1, A_2) and video(V_1, V_2) intervals is possible that satisfy the timing constraints (see Figure 2c, d.)

1. ⟨par⟩ ⟨seq⟩ A_1, A_2 ⟨/seq⟩ ⟨seq⟩ V_1, V_2 ⟨/seq⟩ ⟨/par⟩
2. ⟨par⟩ ⟨seq⟩A_1, V_2 ⟨/seq⟩ ⟨seq⟩ V_1, A_2 ⟨/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 that the system administrator, aware of the presentation requirement, used the right tree of Figure 2.c to define ⟨par⟩ ⟨seq⟩ A_1, V_2 ⟨/seq⟩ ⟨/par⟩ as a disallowed object. Unfortunately, this object is not contained in the tree used for presentation, thus the security requirement cannot be enforced.

Security objects need to be defined in a clear and unambiguous manner. Syntax-dependent representations, like the models for XML formatted textual documents [DdVPS00, DdVPS02, SF02] where the *protection objects* are nodes of the XML tree, do not capture data semantics, needed for multimedia. We have shown in Figure 2 that this approach may lead to incorrect security enforcement. We propose a new approach by defining a normal form for SMIL documents, representing the identity of the *protection object*. This object is not a node in the XML tree, but an equivalence class. The definition of the normal form is given in Definition 1.

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⟩ $C_{1,1}(s)$ $C_{1,2}(s)$ ... $C_{1,n}(s)$ ⟨/par⟩ ... ⟨par⟩ $C_{m,1}(s)$ $C_{m,2}(s)$ ... $C_{m,l}(s)$ ⟨/par ⟩ ⟨/seq ⟩ where $C_{i,j}$ are audio or video intervals.

SmilNF is the basic data object our security model. It provides a syntax-independent and presentation semantics aware representation format. SmilNF's are further decorated with access control and QoS metadata. For this, the *attribute* that are interpreted to generate secure views of the document.

Security classification for smilNF can be defined at ⟨seq ⟩, ⟨par⟩, or leaves (A_1, A_2, V_1, V_2) levels. Figure 3 shows an example of smilNF, decorated with RBAC metadata. A decoration represents the actual classification of the corresponding node and its subtree. We define classification propagation as follows:

Definition 2 (Classification Propagation)

1. *If the node ⟨seq ⟩ is decorated with security metadata m, then all its descendent nodes are also decorated with m and all frames must be played according to the presentation requirements.*
2. *If any of the nodes ⟨par⟩, is decorated with security metadata m, then all of its descendent nodes are also decorated with m and all frames under this ⟨par⟩ must be presented together.*
3. *If any of the nodes, representing an audio or video frame is decorated with security metadata m then this node can be released to a user with comparable security clearance.*
4. *If a node have two inconsistent security metadata decorations, i.e., two different security labels of MAC, then the dominating security label is considered active.* [1]

Using Definition 2, a consistent, and systematic labeling of the semantic object components is possible. The granularity of our access control model is frame level, however, the minimal presentation granularity is smilNF. Further, our model also support association-based constraints, forming a second layer of access control.

4.2 Generalized Security Subject

In this section we present a method to integrate access permissions of different models to define all permitted accesses of a subject. We call this subject the *generalized subject*.

4.2.1 DAC (Discretionary Access Control)

In Discretionary Access Control permits an action `a` to be performed by a subject `s` on an object `o`. This permission is expressed by constructing an access control triple `(s,o,a)`. Let T denote all access control triples of the form (s, o, a), where o is the object, and a is the permitted access on o to s.

4.2.2 RBAC (Role Based Access Control)

Role-Based Access Control models has three entities roles, users, privileges, and two associations,
subject-to-role and role-to-privilege assignments among them. A subject may activate any set of authorized roles within a session to obtain all privileges assigned to the activated role. Each session is associated with a single user, but a user may have several sessions active at a time.

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 $ActR(s) \subseteq AuthR(s)$.

[1] Note, that for DAC or RBAC this is not a problem, since the same document may be accessible to several different roles or users. However, an object can belong only to one of the security classes.

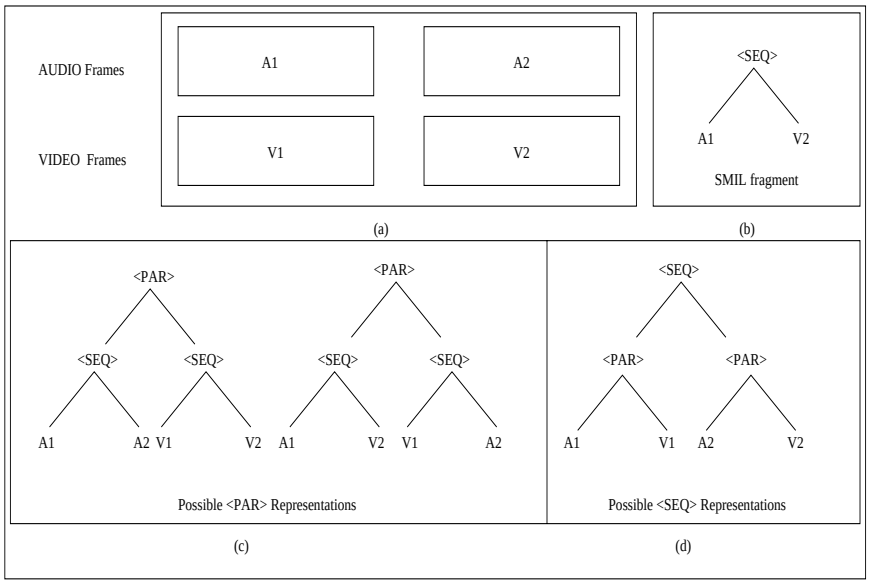

Fig. 2. Equivalence Class of the SMIL Constructs

Access permissions of each role r_i are denoted as rToPer(r_i), where rToPer(r_i) consists (object, action) pairs permitted to role r_i. Then (s,o,a) can be computed as follows:

1. Let s denote the subject and $T = \emptyset$ denote all access control triples of the form (s, o, a), where o is the object, and a is the permitted access on o to s
2. Generate active roles of s as the set $ActR(s) \subseteq AuthR(s)$
3. For each role $r \in ActR(s)$ and subrole r' of r
   (a) For each pair (o, a) in $rToPer(r)$ or in $rToPer(r')$
   (b) $T = T \cup (s, o, a)$

Authorized roles of a subject, and permission assignments can be generated using security ontology and following the "has-role" and "has-permission" properties.

4.2.3 MLS (Multi Level Security)

In Multi Level Security each access permission is guided by the security clearance of the subject and the security classification of the accessed object. Security labels form a lattice structure with the dominance relation among the labels. Information flow between the security labels is controlled based on the security objectives. In this paper we allow information flow from dominated object to a dominating object. Assuming that our access permissions are "read" permissions, it means that 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. We use this closure to identify the security objects that are permitted to be retrieved by the subject.

Let $SecLabel(s)$ denote the classification of subject s. L denotes the lattice structure and $can{-}read(l_1, l_2)$, $l_1, l_2 \in L$ a binary relation, i.e., information can flow from l_2 to l_1. The can-read property corresponds to the dominance relation, that is, $can - read(l_1, l_2)$ means that label l_1 dominates label l_2. The following procedure generates all (s, o, a) for a subject s with security clearance $SecLabel(s)$:

1. Let s denote the subject and $T = \emptyset$ denote all access control triples of the form (s, o, a), where o is the object, and a is the permitted access on o to s
   Generate transitive closure of *can-read*
2. Let $Dominated(s) = \emptyset$
3. For all pairs $can{-}read(l_i, l_j)$, where $l_i = Sec.Label(s)$, $Dominated(s) = Dominated(s) \cup l_j$
4. If $SecLabel(o) \in Dominated(s)$ then (s, o, a)

That is, a subject is granted the access a to an object o if the security clearance of the subject dominates the security classification of the object. Hence MAC could be stated as an (s,o,a) triple. In effect, the generalized access control rule in all three domains could be declared as a (s,o,a) triple.

A generalized subject s acting in behalf of a user is computed based on the unique identifies of the user. Our aim is to provide a seamless representation of users without going through several rounds of authentication and identification. For a give user u and the subject s running in behalf of u, we collect the set of DAC restrictions, permitted security clearance, and assigned roles. These data, along with the ontology is used to transform all allowed accesses into (s, o, a) triplets. All access permissions of a generalized subject is given by $T_{DAC} \cup T_{MAC} \cup T_{RBAC}$.

5 Secure Normal Forms

In the previous section we defined our security objects based on operational semantics of continuous multimedia. Each object is guarded by a particular access control model, however, the collection of all objects is guarded by heterogeneous access control. In this section, we present a method of preprocessing this heterogeneously classified collection to increase the performance of the retrieval. Given a smilNF multimedia document with security metadata attributes, we compute a view that is permitted for each subject, security level, or a role. They are referred to as *security normal forms*. Security normal forms are formally defined in Definitions 3, 4, 5 respectively.

5.1 Normal Form for DAC

The DAC normal form is a parallel composition of permitted segments. The smilNF specification is decorated with the DAC metadata, and upon reduction, would group all permitted segments of a particular subject under a single ⟨par⟩ construct. Each of these ⟨par⟩ construct is the *view* of the associated subject.

Definition 3 (DAC Normal Form) *We say that a smilNF specification ($\tilde{s}$) is in the DAC Normal Form (dacNF) if it is of the form ⟨ seq ⟩ ⟨par⟩ $C_1(\tilde{s})$ ⟨/par⟩ ⟨par⟩ $C_2(\tilde{s})$ ⟨/par⟩ ... ⟨par⟩$C_n(\tilde{s})$ ⟨/par ⟩ ⟨/seq ⟩ where $C_1, C_2, \ldots C_n$ are media intervals permitted to be accessible to subjects $s_1, s_2, \ldots, s_n$, respectively.*

5.2 Normal Form for MLS

Definition 4 (MLS Normal Form) *We say that a smilNF specification ($\tilde{s}$) is in the mlsNF(MLS Normal Form) if it is of the form ⟨seq⟩ ⟨par ⟩ $C_{l_1}(\tilde{s})$ ⟨/par⟩ ⟨par⟩ $C_{l_2}(\tilde{s})$ ⟨/par⟩ ... ⟨par⟩ $C_{l_n}(\tilde{s})$ ⟨/par ⟩ ⟨/seq ⟩ where $l_1, l_2, \ldots, l_n$ represent all security labels, and in C_{l_i} ($\tilde{s}$) is the media stream classified at label l_i.*

As stated in Definition 4, a Normal Form in mlsNF is one that is a parallel composition of single security level documents. This parallel compositions can be viewed as single-level views of the multi-level security multimedia document.

5.3 Normal Form for RBAC

Definition 5 (RBAC Normal Form) *We say that a smilNF specification ($\tilde{s}$) is in the rbacNF (RBAC Normal Form) if it is of the form ⟨seq⟩ ⟨ par ⟩ $C_{r_1}(\tilde{s})$ ⟨/par⟩ ⟨par⟩ $C_{r_2}(\tilde{s})$ ⟨/par⟩ ... ⟨par⟩ $C_{r_n}(\tilde{s})$ ⟨/par⟩ ⟨/seq⟩ where $r_1, r_2, \ldots, r_n$ are $role_1, role_2, \ldots role_n$, and $C_{r_1}(\tilde{s}), C_{r_2}(\tilde{s}), \ldots C_{r_n}$ are media streams permitted to $role_1, role_2, \ldots role_n$, respectively.*

As stated in Definition 5, a Normal Form in rbacNF is one that is parallel composition of at one or more role specifications, where each specification belongs to a particular role assignment, and is said to be the view corresponding to the assigned role.

Note, that generating security normal forms may result in redundant data representation, i.e., if the same object is permitted to both $role_1$ and $role_2$, it appear in the view for both roles. The justification of this approach is to increase the efficiency of play back of the multimedia stream by reducing the complexity of data reconstruction. This approach is similar to the replicated multilevel database architecture.

The algorithms for the reduction of the smilNF to the appropriate secure normal forms, based on the security paradigm that we are using are described in detail in [KFW03b, KFW03c]. When we convert to a secure normal form we encounter different time containers, some of which are nested. The Figure 3 shows an example decoration and reduction in the Role-based environment. The figure represents the tree structure as well as the syntax of the a)smilNF b)RBAC decorated smilNF and c)the view derived after the application of algorithm. During the rewrite, some of the nodes are represented as ⟨ empty ⟩. This representation is used to establish an audio or video *silence* in the playout. As we noted earlier, we assume that each multimedia interval has the same duration. When grouping elements that satisfy a particular access control rule, there is a need to eliminate those that do not qualify. Normally, a silent audio segment or a blank video segment is used during playout to maintain continuity without losing synchronization.

6 DAML Metastructure

The metadata is needed for specifying access control policies for multimedia because the current specification of SMIL [Aya01] does not have constructs for security and QoS. The SMIL metamodule [Mic01] claims that RDF could be used to declare metadata to be used within a SMIL document, but does not provide sufficient details. The RDF [KC03] and RDFS [BG03] enable defining metadata but not the interpretation or anticipated meaning applicable to multimedia. To interpret metadata for enforcing security and quality on SMIL documents, we propose a metastructure based on DAML+OIL.

Figure 4 represents the class hierarchy of the metadata we define in DAML+OIL for specifying security in a SMIL formatted multimedia document. Figure 4 represents those components necessary to represent security parameters chosen for this study.

Figure 5 represents the class hierarchy of the metadata we define in DAML+OIL for specifying QoS in a SMIL formatted multimedia document. Figure 5 represents those

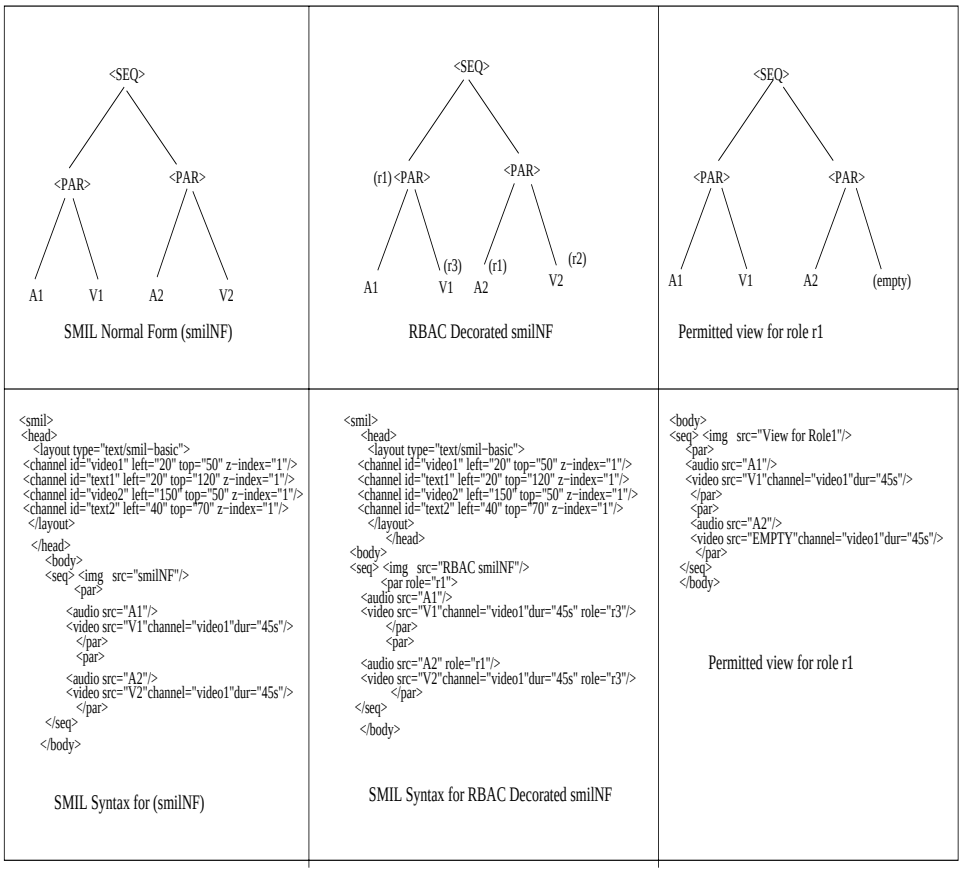

Fig. 3. Decoration and Reduction : RBAC

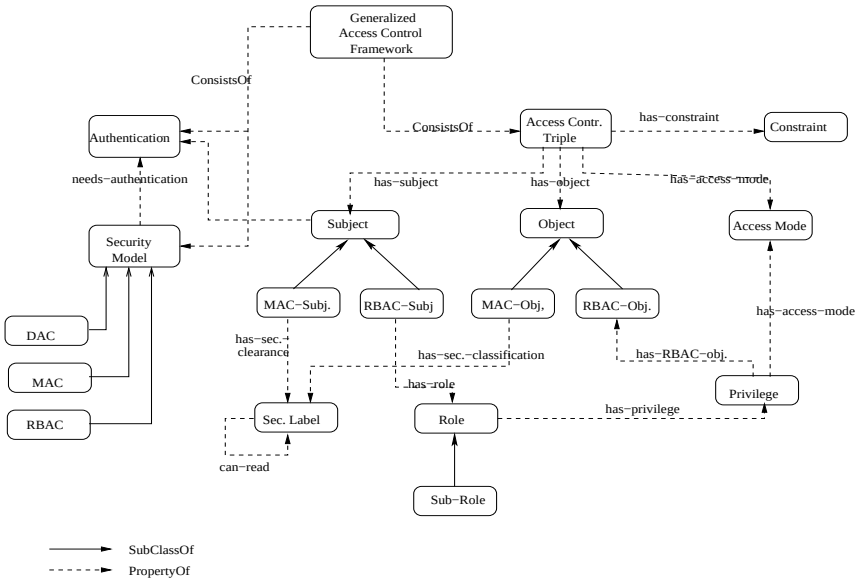

Fig. 4. Class Hierarchy of the Security Metastructure

components necessary to represent QoS parameters chosen for this study.

6.1 Security Metastructure

All DAC, MAC, and RBAC models have been used in practice to ensure secure accesses to protected information. A user may be permitted to accesses as a DAC user, MAC user, and RBAC user. However, it is assumed that

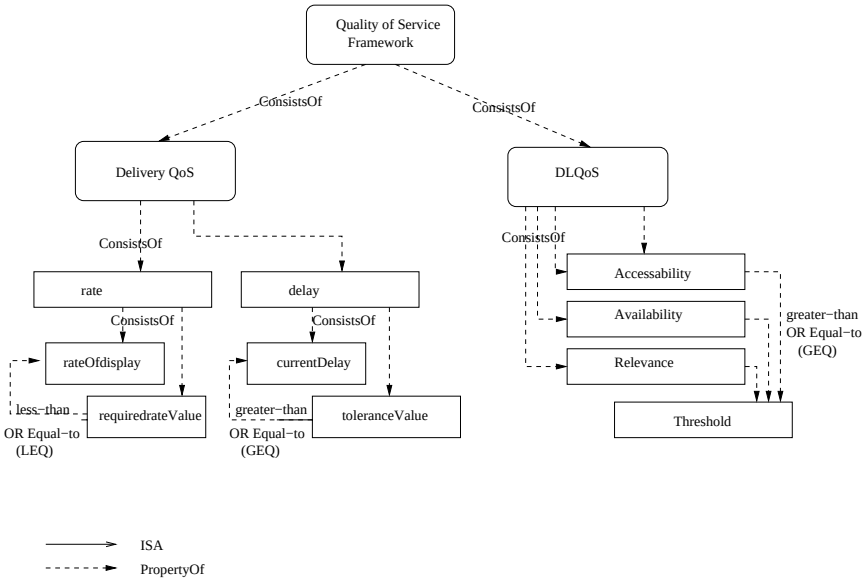

Fig. 5. Class Hierarchy of the QoS Metastructure

objects have only one access control specification, that is an object is either DAC, MAC, or a RBAC object, but only one of these. 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 triples*, and *constraints* that further restrict accesses.

We require, that all documents are in their appropriate security normal forms. To process a data request, the security and QoS ontologies are queried to identify relevant data.

6.2 Access Control Models

First, we define the three access control models used in our framework. They are subclasses of the general class *Security Model*, supporting future extension of our security framework with additional access control. Following is the DAML-OIL specification of the security models.

```
<daml:Class rdf:ID="AC Models">
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>


<daml:Class rdf:ID="Disc Access Control">
<rdfs:comment>
Discretionary Access Control allows the owner
of a security object to define who is allowed
to access that object and what access mode,
i.e., read, write, execute.
</rdfs:comment>
<rdfs:subClassOf rdf:resource="#AC Models"/>
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="Mandatory Access Control">
<rdfs:comment>
   Mandatory Access Control assigns security
   labels to subject and objects. Access
   requests are evaluated based on the
   comparison of these labels.  For example
   the BLP model allows information flow
   from low security labels to high security
   labels.
</rdfs:comment>

<rdfs:subClassOf rdf:resource="#AC Models"/>
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>


<daml:Class rdf:ID="Role-Based Access Control">
<rdfs:comment>
Role-Based Access Control assigns users
to roles and roles to privileges (object,
access mode) pairs. Each access request
is evaluated based on the role a user
plays within a session and the privileges
associated with those roles.
</rdfs:comment>

<rdfs:subClassOf rdf:resource="#AC Models"/>
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>
```

6.3 Access Control Triples

As we mentioned earlier, our aim is to combine all access permissions for a user into a single (s, o, a) triple. For each access control model we define basic concepts, like security label, and their relationships. 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, but can be easily extended to incorporate abstract privileges, like activities. For our application domain, we only need the read (retrieve) permission. Appendix B contains the DAML+OIL specifications for *Access Control Triples* subtree.

```
<daml:Class rdf:ID="Access Control Triples>
 <rdfs:comment>
    Defines the (s,o,+-a) triples.
</rdfs:comment>

<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="Subject"> <rdfs:comment>
   Subjects are the active entities in
   the system. Access permissions are
   requested by the subjects.
</rdfs:comment>

<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

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

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

<daml:Class rdf:ID="Object">
 <rdfs:comment>
 Objects are the passive entities in
 the system. Access permissions are
 requested to the objects.
</rdfs:comment>

<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

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

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

6.4 Constraints

The last component of the security framework represent additional restrictions over access control triples 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. Following are simple examples of constraint specifications in DAML+OIL.

```
<daml:Class rdf:ID="Constraint">
<rdfs:comment>
   Restricts access control rules.
</rdfs:comment>

<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="Time-dependent Constraints">
 <rdfs:subClassOf rdf:resource="#Contraints"/>
 <rdfs:subClassOf rdf:resource="#Sec Struct"/>
 <rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="Context-dependent Constraints">
 <rdfs:subClassOf rdf:resource="#Contraints"/>
 <rdfs:subClassOf rdf:resource="#Sec Struct"/>
 <rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="Obligatory Constraints">
 <rdfs:subClassOf rdf:resource="#Contraints"/>
 <rdfs:subClassOf rdf:resource="#Sec Struct"/>
 <rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>
```

6.5 QoS Metastructure

The DAML-QoS metastructure defines metadata for enabling QoS restrictions and negotiations on Multimedia

digital libraries. The are two sets of metadata defined for DeliveryQoS(Client-Server related) and DLQoS(Digital Library organization and retrieval related). The complete metastructure is given in the Appendix, but we describe the contents in this section.

The DeliveryQoS class and its related sub-classes define metadata pertaining to the issues relating during delivery of multimedia content form the library to the recipient device. The most important delivery factors are *rate* and *delay*. In the service level agreement(SLA) between the library 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.

The DLQoS class and its related sub-classes define metadata for the QoS issues related to the Digital Library itself. QoS requirements for Multimedia digital libraries have been proposed by Bertino et al. [BEH01]. We consider *accessability*, *availability* and *relevance* to be the driving factors for such a QoS requirement. Each of these parameters are marked on a uniform scale, upon which thresholds can be dynamically defined. The allegiance to the threshold would be the deciding factor in deciding the conformance to the QoS requirements.

7 Metadata in SMIL

This section describes how the designed DAML+OIL metastructure could be used in association with a SMIL specification. As stated earlier, the document on which we use the designed metadata must be in one of the normal forms based on the security paradigm that we using. The URI DAML+OIL and RDF metastructure would enable the DAML Interpreter to understand the intended meaning the metadata. For namespace references *smilmetadata* is utilized for the DAML-metastructure. Once the name space is determined, the metadata allowed by the metastructure (*smilmetadata*)could be embedded within the SMIL document. The DublinCore metadata [BMB02]is also used for describing information about the document. The SMIL document represented in Figure 6 uses the DAML metadata that we have created to enforce security and quality restrictions. The Title, Description, Publisher, Date, Rights and Format are from the Dublin Core URI that identifies these as standard descriptors. Additional attributes defined using RDF-Schema (delay and rate) could be used to enforce QoS. But, these parameters are negotiated initially with the display device, even before the body of the SMIL document is interpreted during and if they do not validate to TRUE the document is rejected.

```
?xml version="1.0" ?>
< smil xmlns ="http://aparna.gmu.edu/SMIL−2.0.dtd">
< metadata id="meta−daml">
< rdf:RDF>
< head>
xmlns:rdf = "http://www.w3.org/1999/02/22−rdf−syntax−nssharp$"
xmlns:rdfs = "http://www.w3.org/TR/1999/PR−rdf−schema−19990303sharp$''
xmlns:daml = "http://www.daml.org/2001/03/daml+oil sharp$"
xmlns:dc = "http://purl.org/metadata/dublin\_coresharp$"
xmlns:smilmetadata  = "http://svp/gmu.edu/AudioVideo/.../smil−nssharp$"

< !−− Metadata about the Digital Library −−>

< rdf:Description about="http://aparna.gmu.edu/smilmetadata"
    dc:Title="A QoS Metastructure for Digital Libraries"

  dc:Description=" Authorization Model for DL".
  dc:Publisher="Vishnu"
  dc:Creator="Rajit"
  dc:Date="2004−05−03"
  dc:Rights="Copyright 2004 Thibha"
  dc:Format="text/smil">
< /rdf:Description>


<daml:Class rdf:ID="Role−Based Access Control">
<rdfs:subClassOf rdf:resource="#AC Models"/>
<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>

 <daml:Class rdf:ID="RBAC−Subject">
 <RBAC−Subject>Kodali<\RBAC−Subject>
 <daml:Class rdf:ID="RBAC−Object">

<seq>
    <par>
        < video src = "v1.rm">
        < audio src = "a1.rm">
   </par>
    <par>
      ==========−
    </par>
</seq>

 <daml:Class rdf:ID="Role">
 <Role> Supervisor</Role>
 <daml:Class rdf:ID="Sub−Role">
 <Sub−Role> Facilitator</b>

  <daml:Class rdf:ID="Privilege">
  <Privelege>Can_Read</Privilege>
 < /head>
 < /rdf:RDF>
 < /metadata>
```

Fig. 6. Representing RBAC metadata within SMIL

8 Runtime Operations

Our metastructure can be used by a multimedia client that seeks to obtain SMIL documents with proposed DAML/RDF decorations. Our client must use an RDF based query system for this purpose to generate views for DAC, MLS and RBAC. The DAML Query [FHH03] uses a declarative syntax for selecting DAML representations that meet specified criteria. We show how to construct a DAML query to retrieve the view for a given generalized subject. We show an example queries to retrieve all objects corresponding to particular security classification and for a particular role.An DAML-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 need to have two interacting interpreters, where the SMIL-Interpreter calls the RDF-interpreter to interpret RDF decorations.

8.1 DAML Query Language

The DAML Query Language [FHH03,FHH02] is a formal language that enables query-answer interaction be-

tween the server and the client agents. A DQL query is a query pattern written in DAML+OIL. Literals in the patterns are specified as unbounded variables. The answer consists of a set of terms that bind to the variables and the conjunction of answer sentences are entailed by the knowledge base (referred to as the answer KB) of the client agent.

DQL is designed to support a query-answering dialogue in which the answering agent may use automated reasoning methods to derive answers to queries, the knowledge to be used in answering a query may be in multiple knowledge bases on the Semantic Web, and those knowledge bases need not be specified by the querying agent. In addition, the DQL specification includes formal descriptions of the semantic relationships among a query, a query answer, and the knowledge base(s) used to produce the answer

A DQL query contains a query pattern and an answer KB pattern that could be an answer KB, a list of answer KBs or a variable. If the variable forms the answer KB pattern then, the replying agent send reference to the answer KB that was used.

All variables should either be categorized as *must-bind* or *may-bind*. The answer must contain bindings to all must-bind variables and optionally for the may-bind variables. Examples of optional elements are *Query premise* and *Answer Bundle Size*.

The response to a query (answer) has the following components: Answer sentences to the query, query pattern and agent server that replied to the query. The answer set is enveloped in an answer bundle which contains a continuation token that indicates future interaction patterns.

8.2 Operation of DAML Query

DAMLJessKB [Kop03]is a description logic reasoner for the DARPA Agent Markup Language. The semantics of the language are implemented using Jess, JESS (Java Expert System Shell). The *defquery*is used to obtain the answer set. The answer set is sent across to the querying agent.It is convenient to use JESS defqueries for the purpose of querying the answer KB and returning the answers. We have built an ontology (MSDLQ-Ont) as suggested by Seshagiri et al. [SK03]that lets us express query-patterns in DAML+OIL. We represent facts in our JESS KB as follows:

```
(PropertyValue (s subject) (o object) (a access)
```

We give an example for establishing facts and rules based on the DAMLQuery Ontology in Appendix. below:

```
(defrule  subclassInstances
   "An instance of a subclass is an
   instance of the parent class. This
   enforces and makes meaningful the
   daml:subClassOf relationship"
(PropertyValue daml:subClassOf ?child ?parent)
(PropertyValue rdf:type ?instance ?child) =>
(assert(PropertyValue rdf:type
                      ?instance ?parent)))
```

The *subclassInstances* rule shown above will be activated when the two facts appear in the knowledge base, i.e. when a child that is a subclass of the parent, and an instance that is a subclass of child is found. The result of the rule activation is that a fact stating the resulting instance is a type of the parent class is asserted. Ordered facts can be added to the knowledge base using the assert function. We establish rules and facts as shown above for the entire ontology.
An example defquery is as follows:

```
(defquery Constraints "Find all values having
type constraint:Time-dependent"
(PropertyValue rdf:type
               ?n constraint:Time-dependent))

(defquery RBAC-Object "Find all values having
type  subject:RBAC-Object"
(PropertyValue rdf:type
                    ?n subject:RBAC-Object))
```

By using such a query mechanism, we could obtain views for the generalized subject based on the *SOA-Triples* that we define. The *normal-forms* ensure that all SMIL fragments associated with a generalized subject are grouped together, making the queries easier. Although we propose a syntax, queries could use arbitrary external syntax, as [FHH03, FHH02] do not define a fixed syntax.

8.3 The Run-Time Algorithm

The run-time algorithm detailed in Algorithm 1 describes the retrieval of a secure SMIL document. During the first stage, the algorithm negotiates the QoS parameters. A failure of available QoS would result in the termination of the media transfer. Once the query answer is obtained, the access control policy is evaluated. If access is granted the associated action is initiated. Views could be encrypted to enforce integrity and unwanted stream acquisition and guarantee unforgability. Several encryption techniques can be used, such as the ones suggested in [KWJ03, KW02].

9 Conclusions

In this paper we presented a framework to support the representation of security and QoS requirements in multimedia digital libraries. We showed that syntactic trees used in textual XML documents to specify access control policies are insufficient to specify access control policies

Algorithm 1 Run-time Evaluation Algorithm

```
begin
start DAML Interpreter
negotiate QoS Parameters
if rateOfDisplay = TRUE and toleranceValue = True
then
  query DAML ontology
else
  return with failure
end if
if security-domain = "DAC" then
  run DAC Query
  retrieve associated elements from dacNF
else if security-domain = "RBAC" then
  run RBAC Query
  retrieve associated elements from rbacNF
else if security-domain = "MLS" then
  run MLS Query
  retrieve associated elements from mlsNF
end if
close DAML Interpreter.
GRANT or DENY access to elements
if access == GRANT  then
  create SMIL-View for the set of conditions
  activate action
  start SMIL Interpreter
else if access = DENY then
  return
end if
close SMIL Interpreter
end
```

for SMIL formatted multimedia documents. As a solution, we proposed that SMIL documents to be translated to a normal form and restrictions to be expressed on these normal forms.

We also developed metastructure (ontology) using DAML+OIL [CHH01] to represent access control and QoS policies for multimedia documents. We have shown via examples the applicability of the structure for DAC, MAC, and RBAC. Our security normal forms are similar to secure views computed for XML and other textual documents. We present algorithms to compute normal forms and show a run-time that uses RDF and SMIL queries to securely retrieve documents decorated by security and QoS attributes.

The presented framework and technologies will ensure efficient processing of multimedia documents, while they also guarantee security and semantic preservation. We propose technique to seamlessly integrate different access control paradigms.

Results presented here consider limited aspects of security models with a fragments of SMIL syntax. Our future work extends our current results by addressing additional SMIL constructs. We also plan to implement the extended model, developing joint capabilities for DAML Queries and SMIL processing. Our model focuses on effective data retrieval by preprocessing multimedia data and storing then in secure normal form. While this approach slows data insertion and increase storage requirement (replicated data), it speeds up data request processing.

References

[AABF02] N.R. Adam, V. Atluri, E. Bertino, and E. Ferrari. A content-based authorization model for digital libraries. *IEEE Trans. on Knowledge and Data Engineering*, 14(2):296–315, March 2002.

[AHS02] Anupriya Ankolekar, Frank Huch, and Katia Sycara. Concurrent execution semantics of DAML-S with subtypes. In *The Semantic Web - ISWC 2002, First International Semantic Web Conference, Sardinia, Italy, June 9-12, 2002 / I. Horrocks, J. Hendler (Eds.)*, pages 1–318pp. Springer Verlag, LNCS 2342, May 2002.

[Arm00] W. Arms. *Digital Libraries*. MIT Press, 2000.

[AS03] Kemafor Anyanwu and Amit Sheth. P-queries: enabling querying for semantic associations on the semantic web. In *Proceedings of the Twelfth international conference on World Wide Web*, pages 690–699. ACM Press, 2003.

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

[BEH01] Elisa Bertino, Ahmed K. Elmagarmid, and Mohand-Sad Hacid. Quality of service in multimedia digital libraries. *ACM SIGMOD Record*, 30(1):35–40, 2001.

[BFP02] E. Bertino, E. Ferrari, and A. Perego. Max: An access control system for digital libraries. In *Proceedings of the 26th International Computer Software and Applications Conference*. IEEE, August 2002.

[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 *Conference on Information and Knowledge Management*, 2002.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *The Scientific American Journal*, 2001.

[BMB02] Dave Beckett, Eric Miller, and Dan Brickley. *Expressing simple Dublin Core in RDF/XML*. Dublin Core Metadata Initiative, July 21 2002.

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

[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.

[Den02] Grit Denker. Towards security in daml. *Internal Report, SRI International, Menlo Park*, 2002.

[FHH02] Richard Fikes, Patrick Hayes, and Ian Horrocks. Designing a query language for the semantic web. *The Knwoledge Systems Laboratory, Stanford University*, 2002.

[FHH03] Richard Fikes, Patrick Hayes, and Ian Horrocks. *DAML Query Language(DQL)*, April,2003. http://www.daml.org/2003/04/dql/.

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

[Hor02] Ian Horrocks. Daml+oil: a reason-able web, 2002.

[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.

[KFW03a] 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.

[KFW03b] Naren Kodali, Csilla Farkas, and Duminda Wijesekera. Metadata for multimedia access control. In *In submission to the Journal of Computer Systems Science and Engineering*, 2003.

[KFW03c] Naren Kodali, Csilla Farkas, and Duminda Wijesekera. Multimedia access contol using rdf metadata. In *Workshop on Metadata for Security, WMS 03*, 2003.

[Kop03] Joe Kopena. Daml jesskb. *http://plan.mcs.drexel.edu/DAMLJessKB/*, 2003.

[KW02] Naren Kodali and Duminda Wijesekera. Regulating access to smil formatted pay-per-view movies. In *2002 ACM Workshop on XML Security*, 2002.

[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.

[Lag95] C. Lagoze. A secure repository design for digital libraries. *D-Lib Magazine*, 1995.

[Mar03] David Martin. *DAML based Web-Service Ontology*, May 2003.

[MG01] A.T. McCray and M. E. Gallagher. Principles for digital library development. *Communications of the ACM*, 44(5):48–54, May 2001.

[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.

[PSS02] Peter Patel-Schneider and Jrme Simon. The yin/yang web: Xml syntax and rdf semantics. In *Proceedings of the 11th Int Conf on World Wide Web*, pages 443–453. ACM Press, 2002.

[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.

[SK03] Mithun Sheshagiri and Anugeetha Kunjithapatham. A fipa compliant query mechanism using daml query language. *http://www.csee.umbc.edu/dqlFIPA.html*, June 2003.

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

10 APPENDIX

10.1 APPENDIX A: Access Modes(Roles,Privileges and Classification)

```
<daml: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>

<daml:oneOf rdf:parseType="daml: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"/>
  </daml:oneOf>
</daml:Class>

<daml:Class rdf:ID="Role">
 <rdfs:comment>
   Role defines the "role" a subject/user
   plays within the organization.
</rdfs:comment>

<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="Sub-Role">
 <rdfs:subClassOf rdf:resource="#Role"/>
```

```
 <rdfs:subClassOf rdf:resource="#Sec Struct"/>
 <rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="Privilege">
<rdfs:comment>
  Privilege defines access rights of
  objcets. It corresponds to (object,
  access mode)pairs.
</rdfs:comment>

<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>


<daml:Class rdf:ID="Security Label">
 <rdfs:comment>
 Defines the security classification of
 data of security clearance of subject.
 Has two components: hierarchical
 component and subset components. Forms
 lattice structure and ordered by the
 "dominance" relation represented
 as "can-read" and "can-write" properties.
</rdfs:comment>

<rdfs:subClassOf rdf:resource="#Sec Struct"/>
<rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="Hierarchical Component">
<rdfs:comment>
Defines the hierarchical component of
a security label
</rdfs:comment>

<daml:oneOf rdf:parseType="daml:collection">
 <Hierarchical Component rdf:ID="Top Secret"/>
 <Hierarchical Component rdf:ID="Secret"/>
 <Hierarchical Component rdf:ID="Unclassified"/>
</daml:oneOf>
</daml:Class>

<daml:Class rdf:ID="Subset Component">
<rdfs:comment>
Defines the subset (lattice) component of
a security label
</rdfs:comment>

<daml:oneOf rdf:parseType="daml:collection">
    <Subset Component rdf:ID="{A,B}"/>
    <Subset Component rdf:ID="{A}"/>
    <Subset Component rdf:ID="{B}"/>
    <Subset Component rdf:ID="{ }"/>
  </daml:oneOf>
</daml:Class>
```

10.2 APPENDIX B: QoS Metastructure

```
 <daml:Class rdf:ID="Digital Library">
   <rdfs:label  >MultDL </rdfs:label >
   <rdfs:comment>
   This class of Digital libraries is
   representative of Metastructure.
   </rdfs:comment>
 </daml:Class>

 <daml:Class rdf:ID="QoS Framework">
  <rdfs:subClassOf rdf:resource="#MultDL"/>
  <daml:disjointWith rdf:resource="#Sec Struct"/>
 </daml:Class>

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

 <daml:Class rdf:ID="DLQoS">
  <daml:disjointWith rdf:resource="#SystemQoS"/>
  <rdfs:subClassOf rdf:resource="#QoS Struct"/>
  <rdfs:subClassOf rdf:resource="#MultDL"/>
 </daml:Class >

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

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

<daml:Class rdf:ID= "rateOfDisplay">
 <rdfs:subClassOf rdf:resource="#rate"/ >
 <rdfs:domain rdf:resource="#SystemQoS"/>
</daml:Class>

<daml:Class rdf:ID= "requiredRateValue">
 <rdfs:subClassOf rdf:resource="#delay"/>
 <rdfs:domain rdf:resource="#SystemQoS"/>
</daml:Class>

<daml:Class rdf:ID= "currentDelay">
 <rdfs:subClassOf rdf:resource="#delay"/>
 <rdfs:domain rdf:resource="#SystemQoS"/>
</daml:Class>

<daml:Class rdf:ID= "toleranceValue">
 <rdfs:subClassOf rdf:resource="#delay"/>
 <rdfs:domain rdf:resource="#SystemQoS"/>
```

```
</daml:Class>

<daml:Class rdf:ID="accessability" >
 <rdfs:subClassOf rdf:resource="#DLQoS"/>
 <rdfs:subClassOf rdf:resource="#QoS Struct"/>
 <rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="availability" >
 <rdfs:subClassOf rdf:resource="#DLQoS"/>
 <rdfs:subClassOf rdf:resource="#QoS Struct"/ >
 <rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>

<daml:Class rdf:ID="relevance">
 <rdfs:subClassOf rdf:resource="#DLQoS"/>
 <rdfs:subClassOf rdf:resource="#QoS Struct"/>
 <rdfs:subClassOf rdf:resource="#MultDL"/>
</daml:Class>
```

10.3 APPENDIX C: DAML Query Ontology

```
<rdf:RDF xmlns:rdf="www.w3.org/rdf-syntax-ns#"
xmlns:rdfs="www.w3.org/2000/01/rdf-schema#"
xmlns:daml="www.daml.org/2001/03/daml+oil#"
xmlns:xsd="www.w3.org/2000/10/XMLSchema#"


<daml:Ontology rdf:about="">
   <daml:versionInfo>DLDOQLOntology.daml,v 1.0
   </daml:versionInfo>
   <rdfs:comment>
   Syntax-Ontology for the DAML
   Query Language for MDL
   </rdfs:comment>
   <daml:imports rdf:resource =
   "http://www.daml.org/daml+oil" />
</daml:Ontology>

<daml:Class rdf:ID="Query">
 <rdfs:label>QueryClass</rdfs:label>
 <rdfs:comment>The class of Queries
 </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="QueryId"> <rdfs:label>Query
Identifier</rdfs:label> <rdfs:comment>
   A Unique ID assigned to the Query
</rdfs:comment> </daml:Class>

<daml:Class rdf:ID="QueryPattern">
  <rdfs:label>Query Pattern as (s,o,a)Triples
</rdfs:label>
  <rdfs:comment>
  The Set of Triples that constitute
  the Query
  </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="SOATriple">
   <rdfs:label>SOA Triples</rdfs:label>
   <rdfs:comment>
   Constitutes a Subject and Object
   and Access
   </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="QueryPremise">
   <rdfs:label>QueryPremise</rdfs:label>
   <rdfs:comment>
   A KB  specified as an URI or included in
   the query to be included in the answerKB
   </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="AnswerKB">
   <rdfs:label>AnswerKB</rdfs:label>
   <rdfs:comment>
   The KB in which the Query Pattern needs
   to be quantified
</rdfs:comment> </daml:Class>

<daml:Class rdf:ID="Indication">
<rdfs:label>Bind Indication</rdfs:label>
<rdfs:comment> The List of variables
in the query and the corresponding
bind-type
</rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="Variable">
   <rdfs:label>Variable</rdfs:label>
   <rdfs:comment>
   The List of Variables
   </rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="BindType">
   <daml:oneOf rdf:parseType="daml:collection">
   <daml:Literal rdf:value="may-bind" />
   <daml:Literal rdf:value="must-bind" />
</daml:oneOf>

</daml:Class>
<daml:ObjectProperty rdf:ID="QueryId">
<rdfs:domain rdf:resource="#Query" />
<rdfs:range rdf:resource=
   "http://rdf-schema#Literal"/>
<rdfs:comment />
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="QueryPattern">
   <rdfs:domain rdf:resource="#Query" />
```

```
   <rdfs:range rdf:resource="#QueryPattern" />
   <rdfs:comment />
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="SOATriple">
   <rdfs:domain rdf:resource="#QueryPattern" />
   <rdfs:range rdf:resource="#SOATriple" />
<rdfs:comment />

</daml:ObjectProperty>
   <daml:ObjectProperty rdf:ID="subject">
   <rdfs:domain rdf:resource="#SOATriple" />
   <rdfs:range rdf:resource=
        "http://rdf-schema#Literal"/>
<rdfs:comment />

</daml:ObjectProperty>
<daml:ObjectProperty rdf:ID="object">
   <rdfs:domain rdf:resource="#SOATriple" />
   <rdfs:range rdf:resource=
         "http://rdf-schema#Literal"/>
   <rdfs:comment />
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="access">
<rdfs:domain rdf:resource="#SOATriple" />
 <rdfs:range rdf:resource=
     "http://rdf-schema#Literal"/>
<rdfs:comment /> </daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="AnswerKB">
   <rdfs:domain rdf:resource="#Query" />
   <rdfs:range rdf:resource="#SOATriple" />
   <rdfs:comment />
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="QueryPremise">
   <rdfs:domain rdf:resource="#Query" />
   <rdfs:range rdf:resource="#SOATriple" />
   <rdfs:comment />
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="Indication">
   <rdfs:domain rdf:resource="#Query" />
   <rdfs:range rdf:resource="#Indication" />
   <rdfs:comment />
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="Variable">
   <rdfs:domain rdf:resource="#Indication" />
   <rdfs:range rdf:resource="#Literal" />
   <rdfs:comment />
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="Value">
   <rdfs:domain rdf:resource="#Indication" />
   <rdfs:range rdf:resource="#BindType" />
<rdfs:comment /> </daml:ObjectProperty>

<daml:Class rdf:ID="SOATriple">
 <rdfs:subClassof>
<daml:intersectionOf
          rdf:parseType="daml:collection">
<daml:Restriction>
<daml:onProperty rdf:resource="#subject" />
<daml:cardinality>1</daml:cardinality>
</daml:Restriction>

    <daml:Restriction>
    <daml:onProperty rdf:resource="#object" />
    <daml:cardinality>1</daml:cardinality>
    </daml:Restriction>

    <daml:Restriction>
    <daml:onProperty rdf:resource="#access" />
    <daml:cardinality>1</daml:cardinality>
    </daml:Restriction>
</daml:intersectionOf>
</rdfs:subClassof>
</daml:Class>

<daml:Class rdf:ID="Variable">
<rdfs:subClassof>
    <daml:Restriction daml:cardinality="1">
    <daml:onProperty rdf:resource="#Value" />
    </daml:Restriction>
</rdfs:subClassof>
</daml:Class>
</rdf:RDF>
```