



HAL
open science

Applying Pervasive and Flexible Access Control to Distributed Multimedia Retrieval

Dana Al Kukhun, Dana Codreanu, Ana-Maria Manzat, Florence Sèdes

► **To cite this version:**

Dana Al Kukhun, Dana Codreanu, Ana-Maria Manzat, Florence Sèdes. Applying Pervasive and Flexible Access Control to Distributed Multimedia Retrieval. 2nd International Workshop on Information Management for Mobile Applications (IMMoA 2012), Aug 2012, Istanbul, Turkey. pp.41-48. hal-03761366

HAL Id: hal-03761366

<https://hal.science/hal-03761366v1>

Submitted on 26 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Applying Pervasive and Flexible Access Control to Distributed Multimedia Retrieval

Dana Al-Kukhun, Dana Codreanu, Ana-Maria Manzat, Florence Sedes
Université de Toulouse – IRIT – UMR 5505

118 Route de Narbonne, 31062 Toulouse, France

{kukhun, codreanu, manzat, sedes}@irit.fr

ABSTRACT

The distribution of data sources has formed a classical challenge for data management. The LINDO framework is an open system that manages the indexing, storage and retrieval of multimedia contents that are distributed in different remote servers and generated in a real time basis. The main objective of this framework is to provide efficient information retrieval with minimal processing costs. This was achieved through the proposal of an efficient decentralized content indexing mechanism. When considering the pervasive and mobile access to the managed content, the need of an access control becomes essential. In this paper, we apply an access control layer on top of the LINDO architecture that manages access based on the RBAC model and realizes decision making using the XACML standard. We explore the challenges that face the system in processing access requests showing how an access denial could influence the system's usability especially when returned to a user facing an important situation. Thus, we propose to apply flexible decision making that searches for alternative resources. This operation is performed using PSQRS, a query rewriting system that aims to provide users with pervasive accessibility where they could access any needed multimedia source at anytime, anywhere and anyhow.

1. INTRODUCTION

The necessity of handling a huge quantity of multimedia content created by multiple sources in a distributed environment emerges and raises new challenges concerning the indexing and access to the multimedia content, such as: distributed storage and decentralized processing, choice of the indexing algorithms, real time information retrieval and location-aware retrieval. On top of that we have to consider also that the users are more and more mobile and they need to access the system from anywhere. In such mobile and pervasive contexts, privacy and security management is a central issue.

In this paper, we present a new layer on top of the architecture proposed by the LINDO project¹ in order to tackle the above-mentioned challenges. The objective of the LINDO project was to build a distributed system for multimedia content management, and to ensure effective indexing and storage of data acquired in real time. The project didn't address the issues linked to data privacy and security.

Knowing that ensuring the protection of multimedia content is a key issue in certain application domains (e.g., video surveillance,

medical domain, etc.), the access control management should be taken into consideration at the different levels of data processing and should take into account the user's mobility. Meanwhile, these security constraints should not affect the user's accessibility needs especially in important situations.

Our objective is to include the access control within the query processing and enrich it within the LINDO framework in order to attain a pervasive accessibility that enables the user to access multimedia sources at anytime, anywhere and anyhow. To achieve this goal, we have employed PSQRS – Pervasive Situation-aware Query Rewriting System – that offers adaptive context and situation-aware access solutions. The decision making within the system is based on the RBAC model [10] and employs the XACML standard [16]. These technologies are adapted to the distributed access management needs within the LINDO framework.

The solution overcomes the access denials taking place in real time mobile situations by modifying the query processing mechanism of the LINDO framework and by providing adaptive solutions that can bypass the access control constraints.

Next in section 2, we introduce a state of the art covering the different systems managing distributed multimedia content in 2.1, the basic standards for distributed access control management in 2.2 and some research about multimedia access control in 2.3. The LINDO approach for efficient multimedia distributed content management is described in Section 3 through its architecture, as well as its indexing and querying mechanisms. In section 4, we apply an access control layer on top of the LINDO architecture. In section 5, the adaptive access control solution is illustrated through a video surveillance use case. Finally, conclusions and future work directions are provided in section 6.

2. STATE OF THE ART:

2.1 Distributed Multimedia Systems

The constant growing dimension of the multimedia collections that are generated every day brings to the light problems of efficient indexing and retrieval. The solution to these issues passes through the generation and management of the metadata associated to the multimedia content.

These metadata are obtained through the application of indexing algorithms, which have different performances, purposes and constraints. Besides, a great heterogeneity of indexing algorithms has been defined in the state of the art (e.g., [4] for texts, [13] for images, [18] for audios, [20] for the videos). In a multimedia information system it is not desirable to execute all available

¹ <http://www.lindo-itea.eu>

indexing algorithms on all multimedia contents; because these will (i) overload the system and (ii) produce metadata that might never be used.

In the following, we present some distributed systems that manage multimedia contents by emphasizing the architectural choice and the adopted solution for multimedia indexing.

A distributed management of the multimedia is used by many projects due to the mobile acquisition context of these contents. An advantage of this kind of systems is that they benefit from the distributed storage and processing of the multimedia content and thus, the performances of the system can be improved.

The distributed systems that handle multimedia contents employ peer-to-peer or service-oriented architectures. The major problem that these systems encounter is the heterogeneity of indexing algorithms and of the generated metadata. The following projects addressed this problem in different manners.

The SAPIR (Search on Audio-visual content using Peer-to-peer Information Retrieval) project [2], [15] proposes a hybrid peer-to-peer architecture for the management of multimedia contents. It employs three specialized indexing servers, where each peer sends its ingested contents in order to be indexed. The resulted metadata is sent back to the peer that ingested the multimedia content in order to store it.

The DISCO (Distributed Indexing and Search by Content) project² has chosen a structured peer-to-peer architecture for the management of multimedia contents [5]. The indexing is accomplished at each peer, at the contents acquisition time. Each peer sends a summary of its index that is concatenated to a global index which is sent to all the other peers.

The CANDELA (Content Analysis and Network DELivery Architectures) project³ is focused on the video content analysis and retrieval into a Service Oriented Architecture, where the content is stored and indexed on the distributed servers. The proposed solution was implemented for several use cases: personal mobile multimedia management [17], video surveillance [14], [12].

The WebLab project⁴ proposes an integration infrastructure that enables the management of indexing algorithms as Web Services in order to be used in the development of multimedia processing applications [11]. These indexing services are handled manually through a graphical interface.

The VITALAS (Video & image Indexing and retrieval in the Large Scale) project⁵ capitalizes the WebLab infrastructure in a distributed multimedia environment [22]. The architecture enables the integration of partner's indexing modules as web services. The multimedia content is indexed off-line, at acquisition time.

The MODEST (Multimedia Object Descriptors Extraction from Surveillance Tapes) project⁶ proposes a multi-agent system for the

² <http://www.lamsade.dauphine.fr/disco/index>

³ <http://www.hitech-projects.com/euprojects/candela>

⁴ <http://weblab-project.org/>

⁵ <http://vitalas.ercim.org>

⁶ <http://www.tele.ucl.ac.be/PROJECTS/MODEST/index.html>

video surveillance of motorways, in which they detect strange events, identify objects (persons, cars, trucks) and track the objects in the videos acquired by different cameras [1]. The video content is indexed by a segmentation agent on the same server where it is stored. The obtained segmentation is employed by other collaborative agents in order to detect anomalies, which are displayed to the user as summaries.

A comparative study of these systems shows that no matter what the architectural choice is, the content indexing is usually done on dedicated servers (the content and the associated resulting metadata are transferred over the network) using a pre-defined set of indexing algorithms. These algorithms are executed on all ingested multimedia. Thus, the resource consumption is not optimal. This important consumption problem was addressed by the LINDO project, which proposes a distributed architecture for the management of multimedia contents, which is favoring reduced resource consumption, in terms of data transfers over the network, storage and CPU utilization.

2.2 Distributed Access Control

Access control and privacy protection are key issues nowadays, especially in the context of distributed systems. In this section, we present two main standards that are widely employed for managing access control within distributed environments: the RBAC model and the XACML standard.

2.2.1 The RBAC Model

The principal motivation behind the proposal of the RBAC (Role Based Access Control) model [10] was to enable easy specification and enforcement for enterprise specific security policies in a way that maps naturally to an organization's structure. The RBAC model has simplified the administration and modification (updates) of access privileges especially in the case of assigning permissions for a large number of users accessing distributed resources.

The main concept of the RBAC model was to group users within roles that reflect their organizational positions then, simply

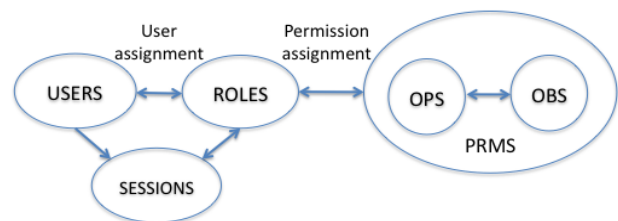


Figure 1: The RBAC Model

distribute permissions to these roles instead of repeating the process for each individual.

As illustrated in Figure 1, the role is placed at the heart of the RBAC model and is seen as an intermediary element that connects between the users and permissions as it attributes a set of privileges to those users based on their roles. These permissions (PRMS) allow the users to perform operations (OPS) on system sources expressed as objects (OBS).

2.2.2 XACML

The RBAC model managed to solve the challenge of administrating access permissions to distributed data sources by

providing centralized management for permissions through roles. With the evolution of service-oriented architectures and web services, new challenges has arisen and the problem of managing access becomes more complicated as the access control policies are also being distributed and more dynamic since they're managed by different administrating authorities. To resolve this problem, the XACML standard was introduced by [16].

XACML (extensible Access Control Markup Language) is an XML based policy language that describes access control policies to allow the attribution of user privileges on system sources. The standard provides a system for authentication and authorization taking into account various factors related to the user's context.

XACML provides an expressive security policy for data exchange within dynamic environments, which enables a flexible way to express and enforce access control policies.

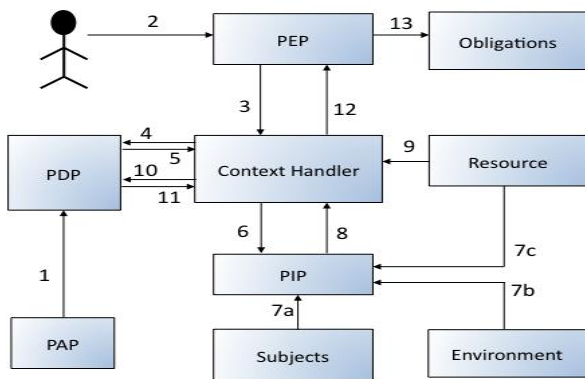


Figure 2: The XACML dataflow

As shown in Figure 2, as a client makes a resource request upon a server; a PEP (*Policy Enforcement Point*) interferes to ensure a secure and authorized access. In order to enforce a security policy, PEP will formalize attributes describing the requester (these attributes can be extracted from the user profile) to the PIP (*Policy Information Point*) and delegate the authorization decision to the PDP (*Policy Decision Point*). Applicable policies are located in a policy store PAP (*Policy Administration Point*) and evaluated at the PDP, which then returns the authorization decision. Using this information, the PEP can deliver the appropriate response to the client and ensures that only authorized resources are accessed.

2.3 MULTIMEDIA ACCESS CONTROL

The projects mentioned in Section 2.1 were focused on the indexing and retrieval of multimedia contents, but none of them took into consideration problems related to the privacy and access control management of the contents and systems resources.

Meanwhile, many solutions have been proposed in order to secure the access to multimedia databases and systems. While some authors were interested in the security of the connection to the systems and on the distribution of the contents [19], others were focused on the content-based multimedia access control with fine-grained restrictions at a specific level of the multimedia data [9].

[8] proposes a framework that addresses multi-level multimedia access control by adopting RBAC, XML, and Object-Relational Databases. The authors associated roles to users, IP addresses, objects and time periods. All multimedia contents handled by

their system have to be segmented. Only the objects which have roles associated to are extracted from the multimedia contents. The system stores several versions of the multimedia contents, the original one and one for each user-based restriction.

[21] Studied the confidentiality and privacy issues in the context of a video surveillance system. They also defined access rights to different hierarchical objects that can be extracted from the video contents. They focused on the detection of suspicious events.

3. THE LINDO APPROACH

3.1 System Architecture

The main goal of the LINDO project (Large scale distributed INDEXation of multimedia Objects) is to define a distributed system for multimedia content management, while focusing on the efficient use of the resources in the indexing and query processes. Thus, not only the multimedia contents storage is distributed but also the indexing process. The originality of this solution is that: (a) the content is not moved to indexing servers, but indexing algorithms are deployed on the servers where the content is acquired; (b) the indexing process is accomplished in two steps: a generic indexing at ingest time (i.e., implicit indexing) and a more detailed one at query time (i.e., explicit indexing). The Figure 3 illustrates an example of the distributed architecture proposed within LINDO project. A more detailed presentation of the LINDO architecture can be found in [6].

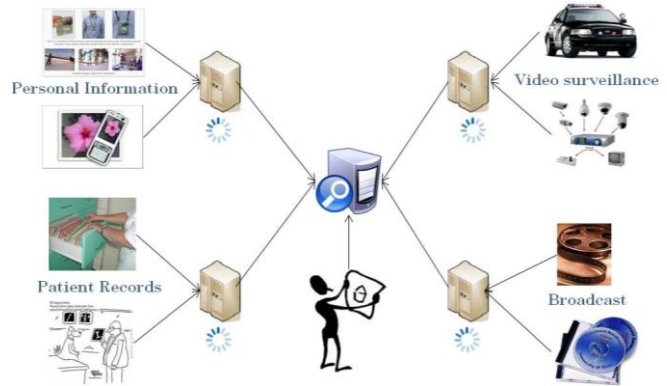


Figure 3: Example of LINDO architecture

Thus, the adopted distributed architecture enables to bypass problems that are specific to centralized systems like:

- (1) The query processing slowness: executing the query on all metadata existing in the system might overload the central server, especially when processing complex queries and when several queries are executed simultaneously.
- (2) The network bandwidth overload: in a classical approach all contents and associated metadata are transferred to central server or to dedicated servers.
- (3) The system centralization: this could rise problems like fault resistance, if the central server is no longer available the metadata collection needs to be recomputed.
- (4) The violation of access rights concerning the contents: some metadata shouldn't be stored on the central server for privacy reasons.



	Indoor	Outdoor
Intrusion	- Presence of people	- Presence of people & vehicles
Counting	- Number of people - Main color of the upper part of the people	- Number of people, number of vehicles - Main color of the people upper part. - Main color of vehicles
		

Figure 4: Examples of Metadata attained by applying Implicit Indexing Algorithms



	Indoor	Outdoor
Intrusion	- Presence of people	- Presence of people & vehicles
Counting	- Number of people - Main color of the upper part of the people - Face recognition - voice recognition & speech-to-text	- Number of people, number of vehicles - Main color of the people upper part. - Main color of vehicles - Car plate number - Face recognition
		

Figure 5: Examples of Metadata attained by applying Explicit Indexing Algorithms

3.2 System Functionality

The functionality adopted within the previously presented system architecture goes as follows: the content is acquired and stored on the remote servers, and the collection of indexing algorithms is stored and managed on the central server. This collection is variable; at any moment we can integrate new algorithms with different functionalities, execution constraints and performances.

3.2.1 Indexing Mechanism

In order to reduce resource consumption, the architecture allows the indexing of multimedia contents to be accomplished at acquisition time (i.e., implicit indexing) with some generic algorithms (e.g., person detection, dominant color detection) and on demand (i.e., explicit indexing) with some algorithms that will analyze the contents more in detail (e.g., person recognition, register plate detection). This avoids executing all the indexing algorithms at once and producing metadata that might never be used but raises access rights issues concerning the explicit indexing. The Figure 4 and Figure 5 offer some indexing algorithms examples that illustrate the difference of the level of detail attained by the implicit and explicit indexing. These algorithms differentiate between two types of context acquisition (indoor and outdoor).

3.2.2 Query Processing Mechanism

The query processing (illustrated in Figure 6) begins with the query specification on the central server. First, the query is processed and executed on the metadata collection on the central server (which is a summary of the metadata collections from

remote servers) in order to select the remote servers that could provide answers to the query and it is sent for execution to the selected servers. Among the servers that were not selected at the first step, there could be some servers that contain relevant information that has not been indexed with the right algorithms. For this reason, the LINDO solution detects such supplementary algorithms [7] and starts their execution (i.e., explicit indexing) on a sub-collection of multimedia contents. All the results obtained from the remote servers are sent to the central server, where they are combined and displayed to the user.

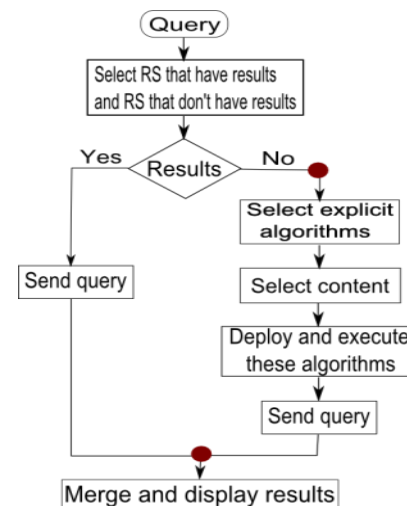


Figure 6: Query Processing Flow Chart

4. ADDING AN ACCESS CONTROL LAYER TO THE LINDO ARCHITECTURE

The sensitivity of the multimedia content and the privacy protection law that imposes anonymity constraints justify the need of applying an access control scheme on top of the LINDO architecture. The proposed layer customizes access based on the user's role (RBAC model) and is responsible for managing:

1. The access rights granted to users or services demanding access to the multimedia sources (e.g., video surveillance, medical domain, etc.) that vary not only according to their role but also in terms of their context (time, location, etc.).
2. The access rights for executing queries that employ the explicit indexing algorithms: the risk of disclosing personal or confidential information arises with the level of detail sought and provided by the indexing algorithm increases.

We highlight that in the context of adding this access control layer, the lack of responses returned to a user's query might not only be due to the lack of results existing within the system but also due to access restrictions imposed by the security layer.

4.1 A Pervasive Vision for LINDO

Our goal is to apply the access control layer and to balance between the security constraints and the user needs to find solutions that can ensure seamless accessibility to the requested resources at any time, from anywhere and anyhow.

The pervasive accessibility that we aim to provide matches with the pervasive characteristics of the LINDO system, which are:

- The distribution of multimedia sources.
- The variation of the entities managing these resources.
- The evolutive nature of these resources (generated and indexed in real time).
- The sensitivity and confidentiality of their content.
- The diversity of contextual information.
- The distribution of the indexing process performed by a variety of indexing algorithms.
- The execution of access requests in real time.
- The importance level of obtaining reactive solutions in important consultations or critical situations.

4.2 Confronting Accessibility Challenges with Adaptive Access Control

Managing access requests becomes more challenging within pervasive environments due to the dynamicity of contextual and situational information. Our objective is to ensure an efficient information retrieval process despite the security challenges. In order to achieve this objective, we employ PSQRS (Pervasive Situation-aware Query Rewriting System) - an adaptive decision-making system that confronts access denials taking place in real-time consulting situations by rewriting access requests in order to offer alternative-based access solutions.

The access control relaxation that we propose to carry out respects the access rights defined to protect the multimedia content and applies the adaptive decision-making at two functionalities:

1. The choice of using the explicit indexing algorithms (located on remote servers).

2. The presentation of the video contents (the identity of filmed persons in a video surveillance system is protected by privacy laws that assure their anonymity).

Next, we introduce the detailed functionality of the PSQRS architecture.

4.3 The PSQRS Architecture

As illustrated in Figure 7, the PSQRS (Pervasive Situation-aware Query Rewriting System) architecture contains several components and the sequence of its functionality starts from the user, who enters the system through an *authentication portal* (step 1) and launches an access request to a certain element (step 2). This request will be interpreted by our *Query Interpreter* that will translate the request into an XACML request and send it to the *Query Analyzer* (step 3). The request (R) will be analyzed in consideration with the user's profile - automatically extracted at the sign in process - and according to his context (XACML flow chart, Figure 2). As the analysis finishes, the *Query Analyzer* would send the result directly to the user if it's a Permit (step 4a) or back to the *Query Interpreter*, if it's a deny (step 4b).

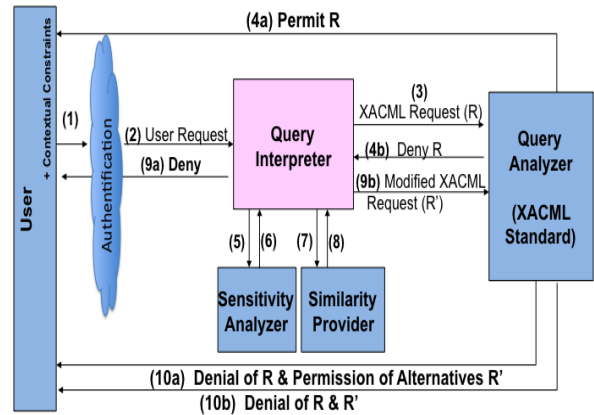


Figure 7: The PSQRS Architecture

In a deny situation the adaptive situation-aware query rewriting mechanism will take place and function as follows: the *Query Interpreter* will check the sensitivity of the consulting situation with the help of the *Sensitivity Analyzer* component (steps 5, 6) and according to the importance level of the situation, the *Query Interpreter* will search for similar or alternative resources through the *Similarity Provider* component (steps 7, 8) and employ them to rewrite the XACML request (R') and send it again to the *Query Analyzer* that will analyze the request and transfer the result back to the user (steps 10a, 10b).

5. VIDEO SURVEILLANCE USECASE

In this section, we present an example where the implementation of our proposal is used to overcome the lack of answers provided by the system. As we will illustrate next, the system will modify the query processing and will adapt access decisions according to the level of importance of the querying situation.

Scenario: Taking the metro from « Trocadéro » station to « Place d'Italie » station at 14:15, Helen has forgotten her red bag on a bench at the waiting line. As soon as she realized, she went out to report the problem at the information counter.

A typical treatment of such situations goes through the customer service agent who would open a lost object file, take the descriptions and transmit them to the security officer on site. The security agent will follow different steps in order to find the object; he will check if the object has already been found or returned to the lost and found office by someone. Otherwise, he will try to see the video surveillance system to check if the object is still in the same location.

5.1 Typical LINDO Query Processing

Figure 8 shows the typical interpretation performed by the information retrieval system provided by LINDO. The launched request will be processed and parsed to extract the main keywords that are then reformulated in the form of an XML user query.

Query: Find all videos containing a *red bag*, forgotten in *Trocadéro, Paris* metro station, on the *2nd of February*, between *2:00pm and now (3:00pm)*.

```

<UserQuery>
  <QueryInText> find all videos containing a red bag, forgotten in Trocadéro, Paris metro station, on 2 February, between 2:00pm and 3:00pm.
</QueryInText>
  <MediaLocation>metro station, Paris, Trocadéro </MediaLocation>
  <MediaFormat>Video</MediaFormat>
  <TimeSpan>
    <From>2012-02-02T14:00:00</From>
    <To> 2012-02-02T15:00:00</To>
  </TimeSpan>
</UserQuery>

```

Figure 8: Request represented in XML

The distributive nature of resource management and query processing in the LINDO system justifies the use of a filtering-based retrieval mechanism. The objective is to find the results that strictly meet the expressed needs in the application and minimize the subset of metadata that the system has to scan in real-time while processing the request.

After keyword extraction [6], the query processing proceeds by locating the servers responsible for managing the data streams captured by the cameras located in the Trocadéro station waiting line. Next, a filtering step is performed to restrict the search within the segments captured between 14:00 and 15:00.

The system will then, determine a list of indexing algorithms that would meet the needs, properties and context expressed within the query. This step will retrieve a subset of metadata describing the segments corresponding to the query.

In this scenario, the requested information are generic thus, the query processing will perform the search on the metadata generated by the implicit indexing algorithms and placed at the central server. The system will continue the search to find a red object in the retrieved list of metadata describing the chosen segments.

A filtering process is applied to take into account access control rules. Analyzing the access rights assigned to the security agent, we find that he is not authorized to access the videos containing passenger faces nor to use the personalized search options that employ the explicit indexing algorithms existing at remote servers. Therefore, considering these access restrictions, the system will perform another filtering step to eliminate the segments that contain people faces and finally return to the user the list of segments that contain a red object (if available).

5.2 Employing PSQRS for Adaptive and Alternative based Query Processing

The search results returned to the security agent in this case might be insufficient especially that the red bag might be present in the unauthorized segments containing passenger faces. Our proposal can take place at this level as a step towards ensuring a better quality of service by offering a wider subset of resources to the user while respecting the access rights defined on the consultation of the video surveillance data sources.

Through the usage of our proposed PS-RBAC model, the system would be able to offer more accessibility and adapt the permissions assigned to the security agent according to his contextual attributes and to the importance level of the situation of the consultation.

This adaptive solution can be employed when the system identifies access challenges related to the user's context or at an important situation. In this scenario, the « lost object » situation identification can be obtained from the file number.

The implementation of the adaptive solutions is performed by the PSQRS that adapts decision-making by rewriting the XACML queries. The solution proves its effectiveness due to its ability to achieve decision making to access video surveillance sources that are distributed and administrated by different authorities.

```

<Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
  http://docs.oasisopen.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
  <Subject>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:subject:subject-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>John Smith</AttributeValue> </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
      DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      <AttributeValue>Security Agent</AttributeValue> </Attribute>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:
      2.0:example:attribute:securityAgent-id"
      DataType="http://www.w3.org/2001/XMLSchema#string" >
      <AttributeValue>sa2023</AttributeValue> </Attribute> </Subject>
  <Resource>
    <ResourceContent>
      <UserQuery> <QueryInText> find all videos containing a red bag,
        forgotten in Trocadéro, Paris metro station, on Thursday,
        2 February, between 2:00pm and 3:00pm).</QueryInText>
      <MediaLocation>metro station, Paris, Trocadéro </MediaLocation>
      <MediaFormat>Video</MediaFormat>
      <TimeSpan>
        <From>2012-02-02T14:00:00</From>
        <To> 2012-02-02T15:00:00</To>
      </TimeSpan> </UserQuery> </ResourceContent> </Resource>
    <Action>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:action:action-id"
        DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>Read</AttributeValue> </Attribute> </Action>
    <Environment>
      <Attribute
        AttributeId="urn:oasis:names:tc:xacml:2.0:environment:environment-id"
        DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>Situation</AttributeValue> </Attribute>
      <Attribute
        AttributeId="urn:oasis:names:tc:xacml:2.0:environment:situation-id"
        DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>Forgotten Object</AttributeValue> </Attribute>
      <Attribute
        AttributeId="urn:oasis:names:tc:xacml:2.0:environment:sitLevel-id"
        DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>1</AttributeValue> </Attribute> </Environment>
  </Request>

```

Figure 9: XACML request embedding the user's query

As shown in Figure 8, the richness of the elements that we can embed within an XACML query enables it to describe the contextual attributes characterizing: (i) the requested source in the « resource » tag, (ii) the user launching the request in the « subject » tag and (iii) the situation at which the user has launched the access request in the « environment » tag.

The importance level of the situation will determine the level of adaptation to be realized. The activation of the adaptive search mode will be communicated from the XACML response in the form of an « obligation » that accompanies the resulting access decision, see Figure 10.

```

<Response>
  <Result>
    <Decision>Deny</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:2.0:status:ok"/>
    </Status>
    <Obligation FulfillOn="Deny"
      ObligationId="ApplyAdaptiveQueryingMode">
      <AttributeAssignment AttributeId="AQM"
        DataType="http://www.w3.org/2001/XMLSchema#string">
        On
      </AttributeAssignment>
    </Obligation>
  </Result>
</Response>

```

Figure 10: XACML response containing the obligation

As the adaptive querying mode is triggered, the query processing mechanism will change to ensure the success of the search by providing a variety of adaptive solutions in correspondence with the situation’s sensitivity level.

This adaptive search solution is realized by the PSQRS that detects the situation sensitivity through the *Situation Analyzer* component and turns to the *Similarity Provider* component to find similar resources that will guide the query rewriting process (see Figure 7).

In the case where the search didn’t retrieve satisfactory results to the user and the consultation is taking place in a normal situation (Sit_Lvl = 0), the system will perform the adaptive query rewriting step through semantic similarity. The keywords of the user query will be reformulated using similar words or more generic concepts offered by the *Similarity Provider*. Similar works have been introduced in [3], the objective is to maximize accessibility chances without crossing the security boundaries.

The semantic reformulation options can be achieved with the help of a standard lexical dictionary such as WordNet. For example, the word "bag" can be replaced by various synonyms {backpack, luggage, purse, etc.}.

At the other hand, the adaptation process in the mentioned scenario will follow another scheme since the lost object situation is judged to be of higher importance (Sit_Lvl = 1). Hence, the *Similarity Provider* component will be replaced by an *Adaptive Solutions Provider*. This component will provide some predefined solutions that could bypass the access control challenge or would assist the user in adapting and reformulating his query by pointing out the access challenge and offering him adaptive solutions that would suit his context, the solutions are often saved in a predefined database. Table 1 shows examples of the solutions that the system can offer.

Table 1: The adaptive solutions that our adaptive query processing can employ

Problem	The adaptive solution
The privacy law imposing the protection of anonymity of audiovisual contents	
Passenger faces are not authorized	Display the content after the execution of an algorithm that applies a blur face function.
Voices are not-authorized	Use an algorithm for speech-to-text transcription
Volume of the video	
Lack of storage capacity on the user’s machine	Use a compression algorithm in order to obtain a smaller file
Format not supported by the user’s machine	Use a conversion algorithm into a compatible format.
Download problems due to a low bandwidth	Use a summarization algorithm in order to obtain a concise version of the content.

New solutions can also be inserted to the adaptive solutions database through a learning mechanism that detects the solutions that users employ when encountered with access challenges in real time.

The success of the adaptive solutions suggested by the users would eventually be more efficient if they knew the reason behind the access denial. The error messages that often accompany the returned access denial responses can serve as indicators to help the users in finding alternative solutions.

Therefore, the adaptive solution for this example will modify the treatment process and will: (i) neglect the filtering step responsible for imposing the access control constraints and (ii) replace it with an adaptive step-related to the presentation of resources with unauthorized content.

By applying this process to the scenario described above, the system will return the video segments taken from the Trocadéro station between 14:00 and 15:00 and containing a red object.

These results will be filtered in order to detect the unauthorized segments (containing passenger faces). This is where the system will apply the adaptation process that would filter the display to conform with the access restrictions imposed by the system.

The adaptation will be performed through a face detection step and the use of an algorithm that applies a “blur function” to protect the privacy of passengers appearing in these segments in order to return to the user a list of pertinent results that respect the access rules.

6. CONCLUSION

In this paper, we have presented an adaptive approach for access control management within multimedia distributed systems. Our solution overcomes the access denials that take place in real time access demands by modifying the query processing mechanism and by providing adaptive solutions to bypass the access control constraints. The proposed solution has been validated within the LINDO framework in the context of a video surveillance use case. We applied and validated the same access control approach for other use cases, such as Healthcare Systems [3].

The adaptive and alternative based situation-aware solution can increase the complexity of processing the request, but if we consider the usefulness of the results provided in real time and the fact they do not violate the access rights defined by the privacy law, this complexity seems quite acceptable.

In future works, we aim to extend our proposal by taking into account different contextual elements that might also influence the accessibility to multimedia content (e.g., hardware, network bandwidth, etc.) and to apply the adaptive process not only at the presentation level but also at the choice of the explicit indexing algorithms that are protected by RBAC constraints.

7. ACKNOWLEDGMENTS

This work has been supported by the EUREKA project LINDO (ITEA2-06011).

8. REFERENCES

- [1] Abreu, B., Botelho, L., Cavallaro, A., Douxchamps, D., Ebrahimi, T., Figueiredo, P., Macq, B., Mory, B., Nunes, L., Orri, J., Trigueiros, M. J., and Violante, A. Video-Based Multi-Agent Traffic Surveillance System. In *Proc. of the IEEE Intelligent Vehicles Symposium*. 2000, 457-462
- [2] Agosti, M., Buccio, E. D., Nunzio, G. M. D., Ferro, N., Melucci, M., Miotto, R., and Orio, N. Distributed information retrieval and automatic identification of music works in SAPIR. In *Proc. of the 15th Italian Symposium on Advanced Database Systems (SEBD '07)*, 2007, 479-482.
- [3] Al Kukhun, D. and Sedes, F., Adaptive Solutions for Access Control within Pervasive Healthcare Systems. In *Proc. of International Conference On Smart homes and health Telematics (ICOST 2008)*, 2008, 42-53.
- [4] Berry, M. W. and Castellanos, M., *Survey of Text Mining II: Clustering, Classification, and Retrieval*, Springer, 2008.
- [5] Boisson, F., Crucianu, M., and Vodislav, D. Publication Framework for Content-Based Search in Heterogeneous Distributed Multimedia Databases. *Scientific Rapport CEDRIC No 1585*, 2008. 18 pages.
- [6] Brut, M., Codreanu, D., Dumitrescu, S., Manzat, A.-M., Sedes, F. A distributed architecture for flexible multimedia management and retrieval. In *Proc. of Database and Expert Systems Applications (DEXA, 2011)*, 2011, 249-263
- [7] Brut, M., Codreanu, D., Manzat, A.-M., and Sèdes, F. Adapting Indexation to the Content, Context and Queries Characteristics in Distributed Multimedia Systems. In *Proc. of International Conference on Signal-Image Technology & Internet-Based Systems (SITIS 2011)*, 2011, 118-125.
- [8] Chen, S.-C., Shyu, M.-L., and Zhao, N. SMARXO: towards secured multimedia applications by adopting RBAC, XML and object-relational database. In *Proc. of the 12th annual ACM international conf. on Multimedia*, 2004, 432-435.
- [9] El-Khoury, V. A Multi-level Access Control Scheme for Multimedia Database. In *9th Workshop on Multimedia Metadata (WMM'09)*, 2009.
- [10] Ferraiolo, D. F., and Richard Kuhn, D. Role-Based Access Controls. In *Proc. of the 15th National Computer Security Conference*, 1992, 554-563.
- [11] Giroux, P., Brunessaux, S., Brunessaux, S., Doucy, J., Dupont, G., Grilheres, B., Mombrun, Y., and Saval, A. Weblab : An integration infrastructure to ease the development of multimedia processing applications, In *the 21st Conference on Software and Systems Engineering and their Applications*, 2008
- [12] Jaspers, E.G.T., Wijnhoven, R.G.J., Albers, A.H.R., Desurmont, X., Barais, M., Hamaide, J., and Lienard B. Candela-Storage, Analysis and Retrieval of Video Content in Distributed Systems: Real-Time Video Surveillance and Retrieval. In *Proc. of the IEEE International Conference on Multimedia and Expo*, 2005, 1553-1556.
- [13] Kosch, H. and Maier, P. Content based image retrieval systems – reviewing and benchmarking, In *Proc. of the 9th Workshop on Multimedia Metadata*, 2009, 1-21.
- [14] Merkus, P., Desurmont, X., Jaspers, E., Wijnhoven, R., Caignart, O., Delaigle, J.-F., and Favoreel, W. Candela - integrated storage, analysis and distribution of video content for intelligent information systems. In *European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT'04)*, 2004
- [15] Michal, B., Fabrizio, F., Claudio, L., David, N., Raffaele, P., Fausto, R., Jan, S., and Pavel, Z. Building a web-scale image similarity search system. In *Multimedia Tools and Applications*. 47, 3(May 2010), 599-629.
- [16] OASIS, A brief Introduction to XACML, http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html, 14 mars 2003
- [17] Pietarila, P., Westermann, U., Jarvinen, S., Korva, J., Lahti, J., and Lothman, H. Candela-storage, analysis, and retrieval of video content in distributed systems: Personal mobile multimedia management. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME '05)*, 2005, 1557-1560.
- [18] Pinquier, J., André-Obrecht, R. Audio Indexing: Primary Components Retrieval - Robust Classification in Audio Documents. In *Multimedia Tools and Applications*, 30,3 (September 2006), 313-330.
- [19] Sánchez, M., López, G., Cánovas, O., Sánchez, J.-A., and Gómez-Skarmeta, A. F. An access control system for multimedia content distribution. In *Proc. of the Third European conference on Public Key Infrastructure: theory and Practice (EuroPKI 2006)*, 2006, 169-183.
- [20] Snoek, C. G., Worring, M. Multimodal video indexing: A review of the state of the art. In *Multimedia Tools and Applications*, 25, 1(January 2005), 5- 35.
- [21] Thuraisingham, B., Lavee, G., Bertino, E., Fan, J., and Khan, L. Access control, confidentiality and privacy for video surveillance databases. In *Proc. of the eleventh ACM symposium on Access control models and technologies (SACMAT '06)*, 2006, 1-10.
- [22] Viaud, M.-L., Thièvre, J., Goëau, H., Saulnier, A., and Buisson, O. Interactive components for visual exploration of multimedia archives. In *Proc. of the International Conference on Image and Video Retrieval*, 2008, 609-616