VideoAcM: a transitive and temporal access control mechanism for collaborative video database production applications

Shermann S.M. Chan · Qing Li · José A. Pino

Abstract Access control models play an important role in database management systems. In general, there are three basic access control models: Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Non-Discretionary Access Control (NAC). Currently, the majority of commercial DBMSs provide only DAC, and some temporal access control models have been derived based on either DAC or NAC. In the context of video database applications, since the structure of video data is complex in nature, it requires a specific and tailor-made access control mechanism which should include MAC as well as DAC and NAC. However, only few efforts have been put on access control models for video database systems. In this paper, a transitive and temporal access control mechanism for collaborative video database production applications has been proposed, which subsumes the properties of DAC, MAC, and NAC. Moreover, our proposed mechanism is integrated with the intellectual property concerns by constructing an access control hierarchy of video data with authorization rules. In particular, our mechanism can derive novel authorization rules not only on conventional client-data access control, but also on data–data access control. Besides video data, the proposed model is applicable to other data types which exhibit a hierarchical data structure.

Keywords Transitive and temporal access control model · Client-data access control · Data–data access control · Authorization rule · Video database

S.S.M. Chan (✉)
Media Research Institute, Faculty of Human Sciences,Waseda University,
2-579-15 Mikajima, Tokorozawa, Saitama, 359-1192, Japan
e-mail: shermann@aoni.waseda.jp

Q. Li
Department of Computer Science,
City University of Hong Kong, 83 Tat Chee Avenue,
Kowloon Tong, Kowloon, Hong Kong SAR, China
e-mail: itqli@cityu.edu.hk

J.A. Pino
Department of Computer Science, Unversidad de Chile,
Av. Blanco Encalada 2120, Tercer Piso, Santiago, Chile
e-mail: jpino@dcc.uchile.cl

1. Introduction

Research on access control models was started several decades ago but the specific topic on Role-based Access Control (RBAC) model has just been investigated in recent years [1, 4, 5, 11–21]. In general, there are three basic access control models: Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Non-Discretionary Access Control (NAC). DAC is identity-based and access is defined for every user. MAC is object-based and access is granted only when both the user and data object have corresponding clearance levels. NAC is role-based and access is granted according to the roles of the users. Currently, the majority of commercial DBMSs provide only DAC. Due to the popularity of the Internet, the requirements of access control for some advanced database applications have been rapidly changing.

Recently, some temporal access control models [3–5] have been derived based on either DAC or NAC. These models concentrated on the temporal effects on user/role relations. In the context of video database applications, since the structure of video data is complex in nature, it requires a specific and tailor-made access control mechanism which should include MAC as well as DAC and NAC. However, only few efforts have been put on access control models for video database systems [2, 6, 7]. Furthermore, video production (including editing) activities involve the collaboration of groups of video producers/editors. This may involve complicated security issues in the collaborative work [19–21], especially for developing collaborative applications based on cross-organizational and collaborative nature of business [9, 10].

This paper proposes a transitive and temporal access control mechanism for collaborative video database production applications. The proposed mechanism is combined with the intellectual property concerns by constructing an access control hierarchy of video data with authorization rules, which are based on groups, sessions, roles and users. In particular, the mechanism can derive authorization rules not only on client-data access control, but also on data–data access control. Besides video data, the proposed model is applicable to other data types which exhibit a hierarchical data structure.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 gives an overview of the proposed collaborative video production database system. Section 4 presents the access control mechanism as well as the access control hierarchy and the intellectual property concerns. Experimental prototype and implementation issues are discussed in Section 5. And finally, Section 6 concludes the paper and suggests future research directions.

2. Related work

Bertino et al. [6, 7] proposed a content-based hierarchical access control model for video database systems. This multi-level access control model combined a video database indexing mechanism with the hierarchical organization of visual concepts. The model defined authorization objects for video elements, which can include semantic clusters, sub-clusters, video scenes, video shots, video frames, and the salient objects. Therefore, users can access video elements at different levels of quality (i.e., granularity levels) according to their permission. The authors focused on the construction of the access control hierarchy by the video analysis and feature extraction techniques, by which the hierarchy considered individual video sources separately. However, video production usually involves several processes and teamwork, which may result in the construction of high-level (logical) video programs possibly composed of different raw video sources. Moreover, their work did not consider the latent access control that can be derived from the multi-level video data hierarchy. In this paper, by contrast, our proposed access control mechanism complements these deficiencies through classifying access controls with the intellectual property concerns, and by providing data–data authorization rules.

Aref et al. [2] have developed a video database research platform, which is embedded with a content-based access control mechanism. Their video database management system consists of three layers: object storage system layer, object relational database management layer, and a user interface layer. However, their access control mechanism is directly based on streaming video, in which there is no data access management for the specific layers.

3. Collaborative video database production: an overview

Existing media production tools and methods require tightly coupled community of users to create media content from the processes of capture, editing, and reuse of media streams. These tools make media content accessible and manipulable with limited access control. Besides motion pictures studios, special interest groups of the Internet users lack a secure platform to share their video captures for later production and storage. In this section, the architecture of this type of video database production application with the proper access control is introduced and the associated data structure is proposed to facilitate this collaborative work.

3.1. Architecture

Figure 1 shows the architecture of our proposed collaborative video database production system. The Video Production Client consists of groups of collaborative

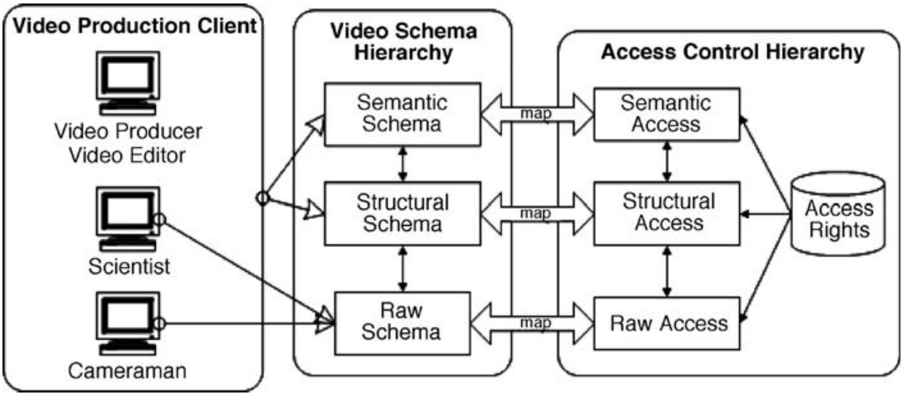


Fig. 1 Architecture of the proposed collaborative video database production system

users who are involved in the various video production processes: Video Producer and Video Editor, Scientist, and Cameraman. In the Video Schema Hierarchy (VSH), video-related structures are classified into Raw Schema, Structural Schema, and Semantic Schema. Groups of Cameramen produce raw video streams and store them into the Raw Schema. Groups of Scientists, who may possibly be programmers and mathematicians, store their algorithms and procedures into the Raw Schema, which will then be used to build the Structural Schema. Finally, groups of Video Producers and Video Editors initiate sessions for the construction of the Structural Schema and the Semantic Schema. They can invite Scientists and/or Cameramen to be observers during their sessions. Each level of the VSH is associated with the corresponding access rights (i.e., Raw Access, Structural Access, and Semantic

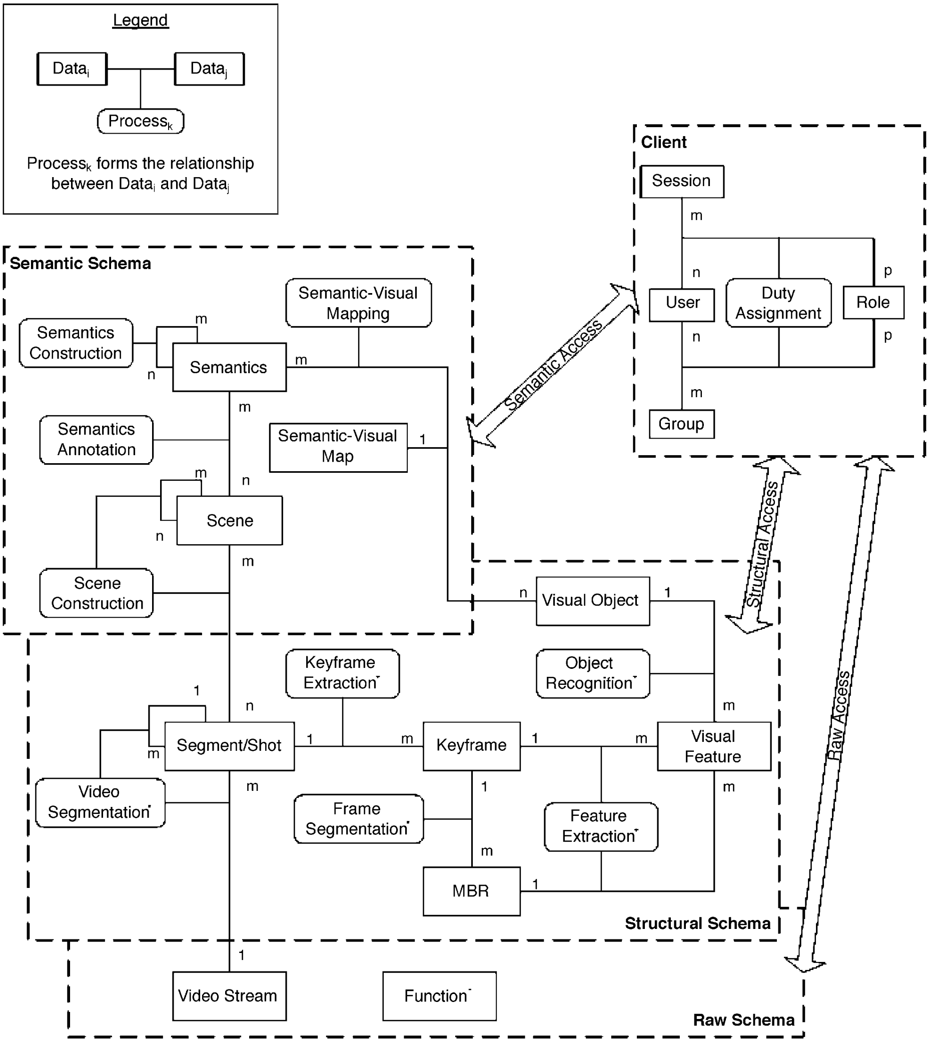


Fig. 2 Data structure of the proposed collaborative video database production system

Access). Since there are co-relations between levels of VSH, authorization rules of the Access Control Hierarchy can be transitive.

3.2. Data structure

Figure 2 shows the data structure used in the proposed collaborative video database production system. For the Video Production Client, the notions of Group, Session, Role, and User are used to model collaborative participants through the process of Duty Assignment. In the Raw Schema, raw videos are imported and stored in the Video Stream (in fact, references are often used to locate the external raw videos), and algorithms are stored in the Function repository. The Function repository plays an important role in the construction of the Structural Schema, within which each process can have a number of associated algorithms. Therefore, the same raw video can be processed several times using various Video Segmentation algorithms. After the process of Video Segmentation, a raw video is segmented into a number of segments/shots. With the Keyframe Extraction component, keyframes can be extracted from the segments and shots. The extracted keyframes will then go through two processes: Frame Segmentation and Feature Extraction. Therefore, MBRs (Minimum Bounding Rectangles) and low-level visual features can be obtained. With the set of visual features, visual object can be recognized and reasoned out. For other relevant data structures such as moving objects, they can be modeled by combining visual objects, temporal information among keyframes, and/or spatial information from MBRs.

Besides the Structural Schema, logical and semantic concepts can be attached to the video data. Segments are dynamically grouped together to form a scene, and the scenes can then be clustered by other scenes. This type of human-oriented process may likely be very subjective, collaborative teamwork is therefore necessary. Similarly, Semantics Construction, Semantics Annotation, and Semantic-Visual Mapping require collaboration as well.

3.3. Access flow

Figure 3 shows the access flow of the proposed collaborative video database production system. At the beginning, the process of *User authentication and authorization* masters the gateway to the system. After passing through the check of the security gateway, a list of authorized and corresponding sessions is available for the user to join. If the user refuses to join any session, s/he should possess the role of administrator or coordinator in order to start the process of video database and session management, else the system will terminate.

After the user has chosen to join an authorized session, the corresponding data objects of that session are retrieved and checked by the authorization rules. In the case of passing through the check of the authorization rules successfully, access modes will be assigned to the data objects. Otherwise, the system will terminate.

When the valid data objects are retrieved, the access control mechanism can start to retrieve transitive session objects. This iterative process of retrieval will continue until the transitive session objects cannot pass through the check of the authorization rules. Then, the user can start the process of video production, which is monitored by the object tracking module.

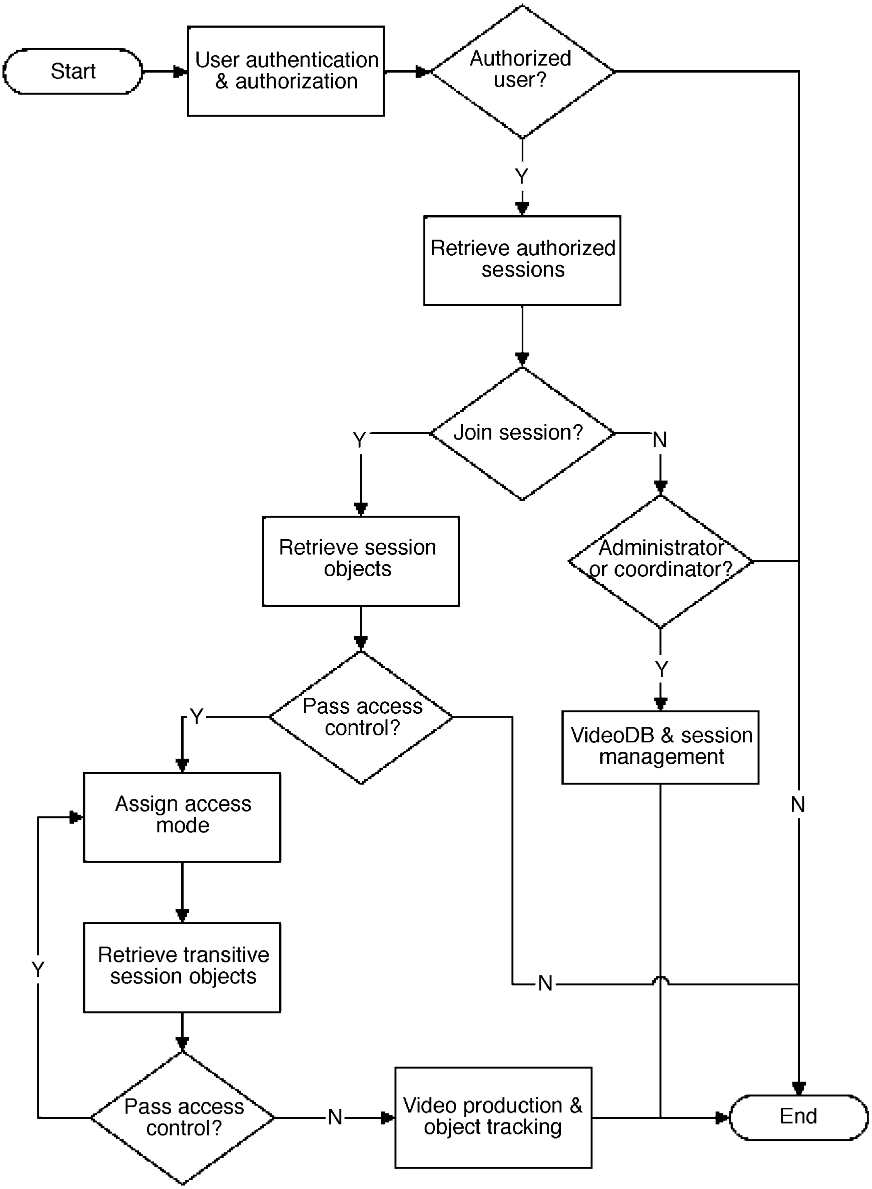

Start
User authentication & authorization
Authorized user?
Y
N
Retrieve authorized sessions
Join session?
Y
N
Retrieve session objects
Administrator or coordinator?
Y
Pass access control?
Y
N
VideoDB & session management
Assign access mode
Retrieve transitive session objects
Y
Pass access control?
N
Video production & object tracking
End

Fig. 3 Access flow of the proposed collaborative video database production system

4. Access control mechanism

Access control management can be integrated into database systems by defining a set of authorization rules. In general, an authorization rule is a triplet <*S, O, M*>, where *S* represents the subject/entity trying to access the object *O* by using the access mode *M*. The authorization rules centralize the access control by linking the subject (e.g., user, role, or group) and object (i.e., the data) together [6, 7].

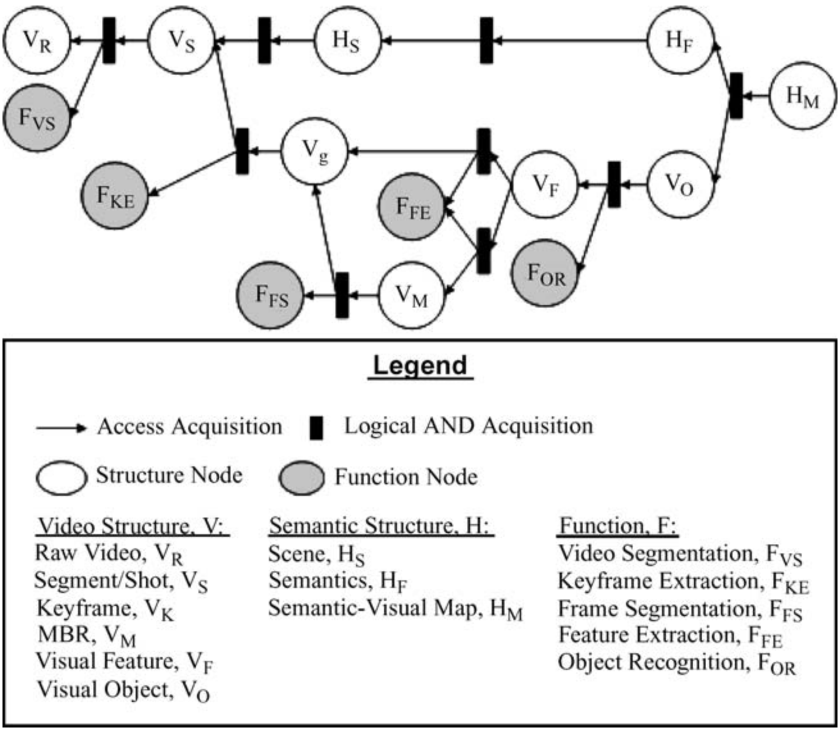


Fig. 4 Access control levels of the video schema hierarchy

4.1. Definition of authorization rule

In our proposed access control model, an authorization rule is a quadruplet <*S, O, AM, P*>, where *S* is a set of Subject entities that want to access a set of object entities *O* and *AM = {A:M}* is a set of Access-Mode matrix and $P \subseteq (P_V \cup P_T)$ represents a set of Predicates which specify conditions and constraints for the entire tuple.

4.1.1. Subject & object

Subject entities can be groups, sessions, roles, users and the data. In order to eliminate possible conflicts, the set *S* of Subject entities is restricted to be group-based (i.e., group, group–role, group–user–role), session-based (i.e., session, session–

Fig. 5 Access-mode matrix

			M			
			M_X	M_R	M_W	M_D
↑		A_H	M_X^H	M_R^H	M_W^H	M_D^H
composite of	A	A_V	M_X^V	M_R^V	M_W^V	M_D^V
↓		A_R	M_X^R	M_R^R	M_W^R	M_D^R

Table 1 Client–data authorization rules, with subject *S*, object *O*, and access-mode matrix *AM*

Transitive predicate	Temporal predicate	*SchLevel* constraint	*SchLevel* accessible	Permission constraint	Temporal constraint	Priority
N	N	*O.SchLevel≥ AM.SchLevel*	*O.SchLevel*	N	N	4
Y	N	*O.SchLevel≥ AM.SchLevel*	*O.SchLevel*	*O.SchLevel to AM.SchLevel*	N	3
N	Y	*O.SchLevel≥ AM.SchLevel*	*O.SchLevel*	N	Y	2
Y	Y	*O.SchLevel≥ AM.SchLevel*	*O.SchLevel*	*O.SchLevel to AM.SchLevel*	Y	1

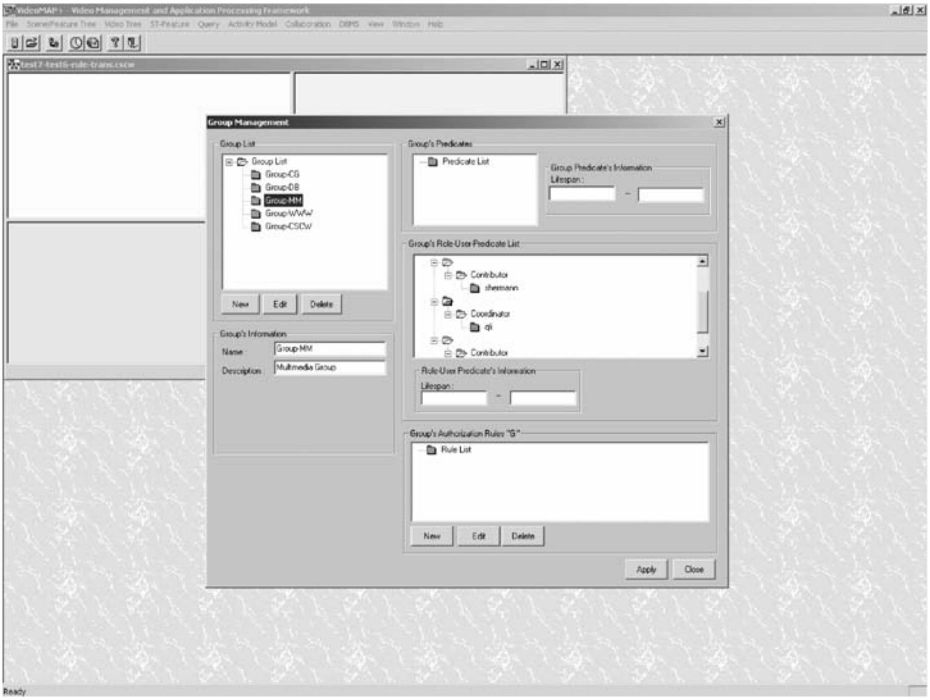
role, session–user–role), and data-oriented (i.e., data objects stored inside the Video Schema Hierarchy). For group-based and session-based Subject entities, we classify them into Client *C*. For data-oriented Subject entities, they are categorized into Video Structure *V*, Semantic Structure *H*, and Function *F*. Hence $S \subseteq (C \cup V \cup H \cup F)$.

In general, a static group is useful for users management, while a dynamic session is used to manage collaborative tasks. In real situations, a user may belong to a fixed number of static groups and any number of dynamic sessions. We make the assumption that all tasks involved in a video database production system are collaborative. Although there could be only one user to complete a task, he should form a session to allow others to access his work at a later stage.

However, the Object set *O* can only be data-oriented. It is a set of video-related data objects (i.e., $O \subseteq (V \cup H \cup F)$, Video Structure *V*, Semantic Structure *H*, and Function *F*). Thus, our authorization rules can be defined in the form of "client-

Table 2 Data–data authorization rules, with subject, object *O*, and access-mode matrix *AM*

Transitive predicate	Temporal predicate	*SchLevel* constraint	*SchLevel* accessible	Permission constraint	Temporal constraint	Priority
N	N	*(S.SchLevel> O.SchLevel)* ∨ *(S.SchLevel≥ AM.SchLevel)*	*S.SchLevel*	N	N	3
Y	N	*(S.SchLevel> O.SchLevel)* ∨ *(S.SchLevel≥ AM.SchLevel)*	*S.SchLevel to O.SchLevel*	*S.SchLevel to AM.SchLevel*	N	4
N	Y	*(S.SchLevel> O.SchLevel)* ∨ *(S.SchLevel≥ AM.SchLevel)*	*S.SchLevel*	N	Y	1
Y	Y	*(S.SchLevel> O.SchLevel)* ∨ *(S.SchLevel≥ AM.SchLevel)*	*S.SchLevel to O.SchLevel*	*S.SchLevel to AM.SchLevel*	Y	2



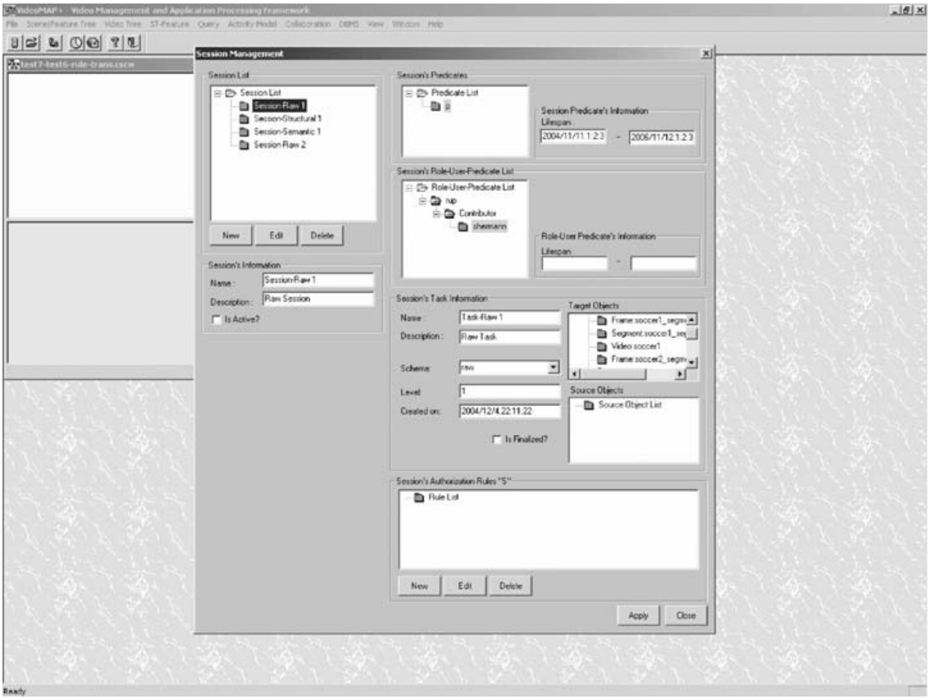
Name: Group-MM
Description: Multimedia Group

Fig. 6 Static group management

data" specification and "data–data" specification. Therefore, implicit access control of data from one level to another can be derived. In the case of video production, one level of data is constructed from another level, except for the Raw Schema. For other cases, a system administrator can predefine the "data" hierarchy in order to have implicit and automatic access control assignments.

Client. As mentioned above, a group-based and session-based Subject set can be represented by Client: $C \subseteq (C_G \cup (\{C_{Gi}{:}C_{Rj}\} | C_{Gi} \in C_G \wedge C_{Rj} \in C_R) \cup (\{C_{Gi}{:}C_{Uj}{:}C_{Rk}\} | C_{Gi} \in C_G \wedge C_{Uj} \in C_U \wedge C_{Rk} \in C_R) \cup C_S \cup (\{C_{Si}{:}C_{Rj}\} | C_{Si} \in C_S \wedge C_{Rj} \in C_R) \cup (\{C_{Si}{:}C_{Uj}{:}C_{Rk}\} | C_{Si} \in C_S \wedge C_{Uj} \in C_U \wedge C_{Rk} \in C_R))$. For a static group C_G, the definition is $C_G = \{C_{Gi}, \{C_{Rj} : C_{Uk}\}\}$ where C_{Gi} is a group instance with a set of role–user associations $\{C_{Rj} : C_{Uk}\}$, with C_{Rj} and C_{Uk} being a role instance and user instance, respectively. For a dynamic session C_S, the definition is $C_S = \{<C_{Si}, \{C_{Rj}{:}C_{Uk}, P_{j:k}\}, P_i>\}$ where C_{Si} is a session instance with a set of role–user associations $\{C_{Rj}{:}C_{Uk}, P_{j:k}\}$ and a set of predicates P_i with C_{Rj} and C_{Uk} being a role instance and user instance, respectively; furthermore, $(C_{Rj}{:}C_{Uk})$ has a particular set of predicates $P_{j:k}$.

In general, there is a set C_R of standard Roles, $C_R = \left(C_R^L \cup C_R^C \cup C_R^O\right)$, where C_R^L is a set of coordinators who are leaders among groups or sessions, C_R^C a set of contributors who are responsible for contributing their work to the groups or sessions, and C_R^O a set of observers who can only read the content but can give comments to the groups/sessions. Roles are used to manage access modes of clients. In order to support additional user-defined role types based on these standard roles, C_R^L, C_R^C and C_R^O have their own priority levels. However, within a group or a session, there


VideoMAP+ - Video Management and Application Processing Framework
File Scene/Feature Tree Video Tree ST-Feature Query Activity Model Collaboration DBMS View Window Help
test7-test5-rule-trans.cscw
Session Management
Session List
New
Edit
Delete
Session's Predicates
Session's Role-User-Predicate List
Session's Task Information
Target Objects
Source Objects
Session's Authorization Rules "S"
New
Edit
Delete
Apply
Close
Ready

Session List: Session List; Session-Raw 1; Session-Structural 1; Session-Semantic 1; Session-Raw 2

Session's Information
Name: Session-Raw 1
Description: Raw Session
□ Is Active?

Predicate List: p

Session Predicate's Information
Lifespan: 2004/11/11 1:2:3 - 2006/11/12 1:2:3

Role-User-Predicate List: rup; Contributor; shermann

Role-User Predicate's Information
Lifespan: -

Name: Task-Raw 1
Description: Raw Task
Schema: raw
Level: 1
Created on: 2004/12/4 22:11:22
□ Is Finalized?

Target Objects: Frame:soccer1_segm; Segment:soccer1_se; Video:soccer1; Frame:soccer2_segm

Source Object List

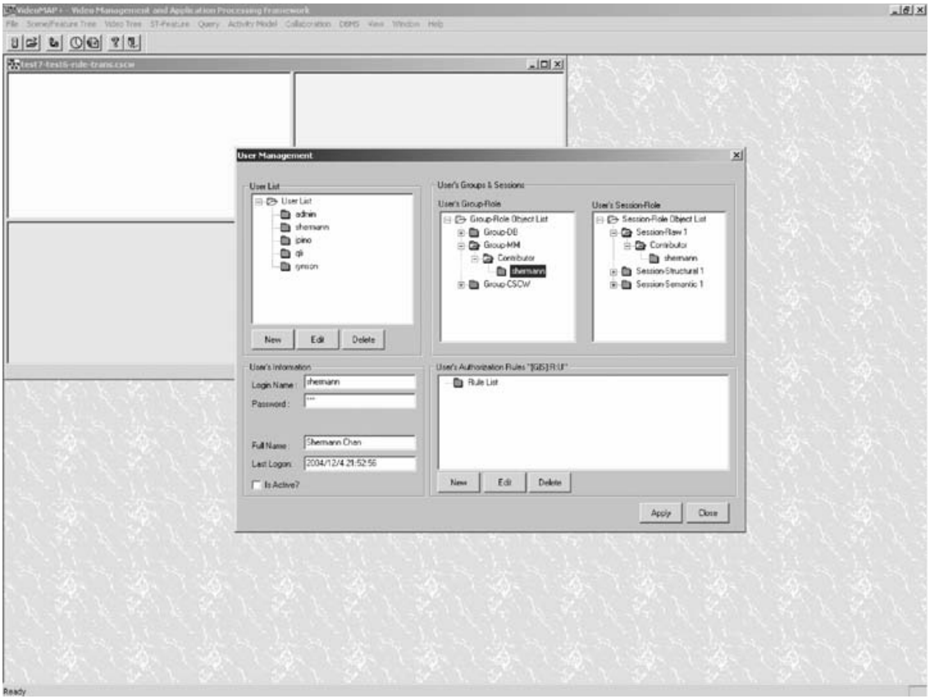
Rule List

Fig. 7 Dynamic session management

should be only one coordinator with the highest priority level. This is very important for critical events and decision-making processes (e.g., creation, modification, deletion, communication, and member invitation) in a collaborative environment. For example, for the event of deleting a piece of data within a session, the contributor only requires the permission from other contributors who have the same or higher priority levels, instead of from the coordinators, observers, or contributors who have lower priorities. Nevertheless, the actual acquisition of permissions from different role types (e.g., coordinators in this case) depends on the specific application domain.

In addition, with an attribute of effort distribution (for contributors only), we can evaluate the intellectual property (*IP*) among contributors within/among session(s). In particular, there is no *IP* for the roles of coordinators and observers, and the summation of *IP* among all contributors should be less than or equal to one. For example, a manager has a role of coordinator, but he has also made some contribution to the task, hence he should have two types of roles: coordinator and contributor.

Video structure, semantic structure & function. The data structure for collaborative tasks (i.e., Video Structure *V*, Semantic Structure *H*, and Function *F*) is a 9-tuple: <*TaskId, TaskName, TaskDesc, SchLevel, SessLnk, LocLnk, CreateDt, IsFinalized, {SrcLnk}*>. *TaskId* is used to uniquely identify the data structure of the collaborative task, *TaskName* stores the name, *TaskDesc* stores the description of the task, and *SchLevel* stores the level of the Video Schema Hierarchy. *SessLnk* is a link pointing to the session object, which stores the information of the collaborative clients.



Login Name : shemann

Password : ***

Full Name : Shemann Chan

Last Logon : 2004/12/4 21:52:56

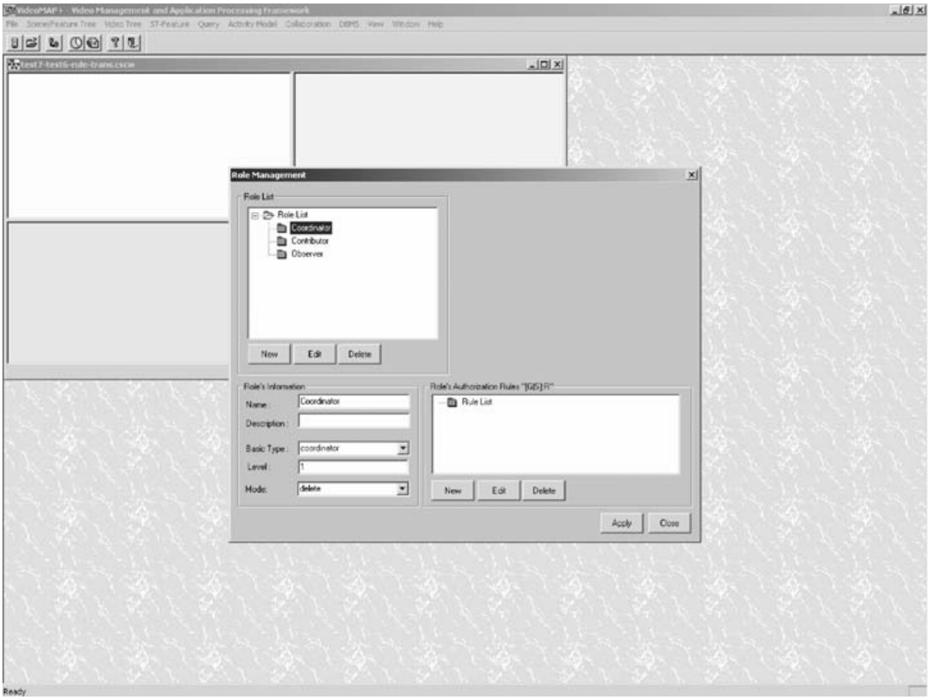
□ Is Active?

Fig. 8 Collaborative user management

LocLnk is the actual storage location of the data structure, and it can be an object pointer or an external link (e.g. video streams can be stored externally). *CreateDt* denotes the creation date and time, and *IsFinalized* is an indicator to mark whether the data is available for public access. *SrcLnk* stores the link pointing to the source of the data. When other groups/users want to access this data, they may need to acquire sufficient access permission from all its sources. This can be done by checking the access permission from the authorization rules, and sending requests to the source owners.

The Video Structure is defined as $V \subseteq (V_R \cup V_S \cup V_K \cup V_M \cup V_F \cup V_O)$, where V_R represents Raw Videos, V_S represents Segments/Shots, V_K represents Keyframes, V_M represents MBRs, V_F represents Visual Features, and V_O represents Visual Objects. The Semantic Structure H is defined as $H \subseteq (H_S \cup H_F \cup H_M)$, where H_S represents Scenes, H_F represents Semantics, and H_M represents Semantic-Visual Maps. The Function $F \subseteq (F_{VS} \cup F_{KE} \cup F_{FS} \cup F_{FE} \cup F_{OR})$ includes a set of algorithms and procedures for Video Segmentation F_{VS}, Keyframe Extraction F_{KE}, Frame Segmentation F_{FS}, Feature Extraction F_{FE}, and Object Recognition F_{OR}.

There is a possibility that a user may specify improper authorization rules, which may allow a low-level video related data object (e.g., raw video stream) to be able to access its relevant (or may be irrelevant) high-level video related data object (e.g., segments generated from this raw video stream). Therefore, an assumption is made for the construction of the data–data authorization rule: *S.SchLevel*>*O.SchLevel*. Figure 4 shows the access control levels of the Video Schema Hierarchy.



Role List
- Coordinator
- Contributor
- Observer

Role's Information

Name: Coordinator

Description:

Basic Type: coordinator

Level: 1

Mode: delete

Role's Authorization Rules "[G|S]:R"
- Rule List

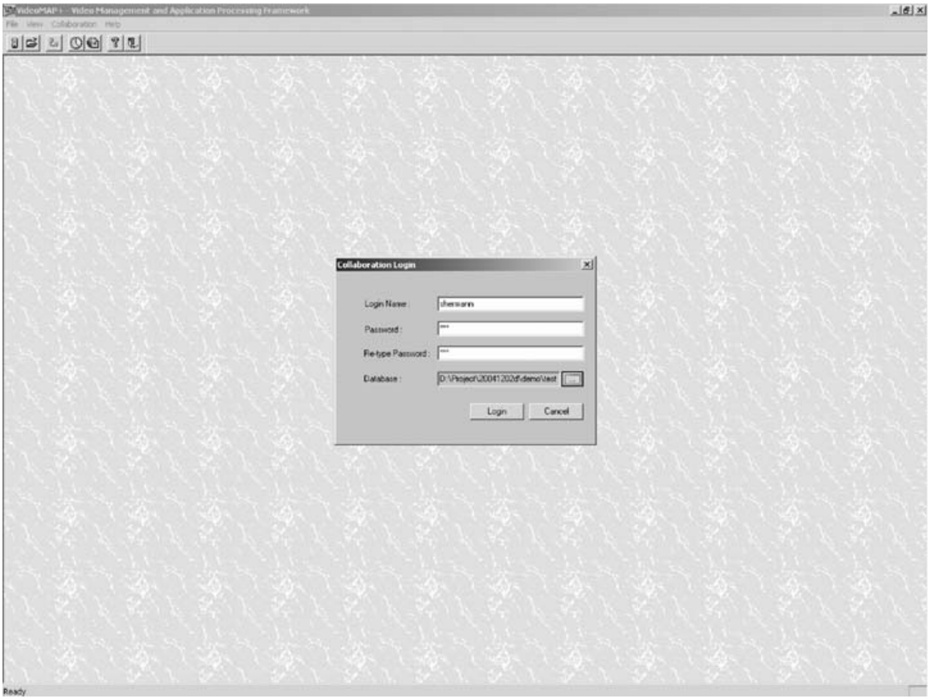
Fig. 9 User role management

4.1.2. Access-mode matrix

The access-mode matrix *AM={A:M}* forms the central component of our access control model, where *A* denotes the Access Control, and *M* the Operation Mode.

Access control & operation mode. The Operation Mode is defined as $M \in \{M_X, M_R, M_W, M_D\}$, where M_X, M_R, M_W, M_D are the execute mode, read mode, write mode, and delete mode across levels of the Access Control Hierarchy, respectively. For each mode, it can be divided into more specific levels. We use the superscript *R* to indicate that the mode is at the Raw Access Level, the superscript *V* to denote that the mode is at the Structural Access Level, and the superscript *H* to denote that the mode is at the Semantic Access Level. For example, $M_X \in \{M_X^R, M_X^V, M_X^H\}$, $M_R \in \{M_R^R, M_R^V, M_R^H\}$, $M_W \in \{M_W^R, M_W^V, M_W^H\}$, and $M_D \in \{M_D^R, M_D^V, M_D^H\}$.

The Access Control is defined as $A = A_R \cup A_V \cup A_H$, where $A_R \in \{M_X^R, M_R^R, M_W^R, M_D^R\}$ represents operation modes at the Raw Access level of the Access Control Hierarchy, $A_V \in \{M_X^V, M_R^V, M_M^V, M_D^V\} \oplus A_R$ represents operation modes at the Structural Access level plus the corresponding operation modes from the Raw Access level, and $A_H \in \{M_X^H, M_R^H, M_W^H, M_D^H\} \oplus A_V$ represents operation modes at the Semantic Access level plus the corresponding operation modes from the Structural Access and Raw Access levels. For example, an access-mode matrix may be $AM_i = A_V{:}M_X = \{M_X^V, M_X^R\}$, yet another may be $AM_j = A_H{:}M_R = \{M_R^H, M_R^V, M_R^R\}$. The Composite Access Operator, ⊕, allows users to specify access control from one level to another level. Figure 5 illustrates the Access-Mode matrix in a tabular way.


VideoMAP+ - Video Management and Application Processing Framework
File View Collaboration Help
Collaboration Login
Login
Cancel
Ready

Login Name : chenssnn

Password : ***

Re-type Password : ***

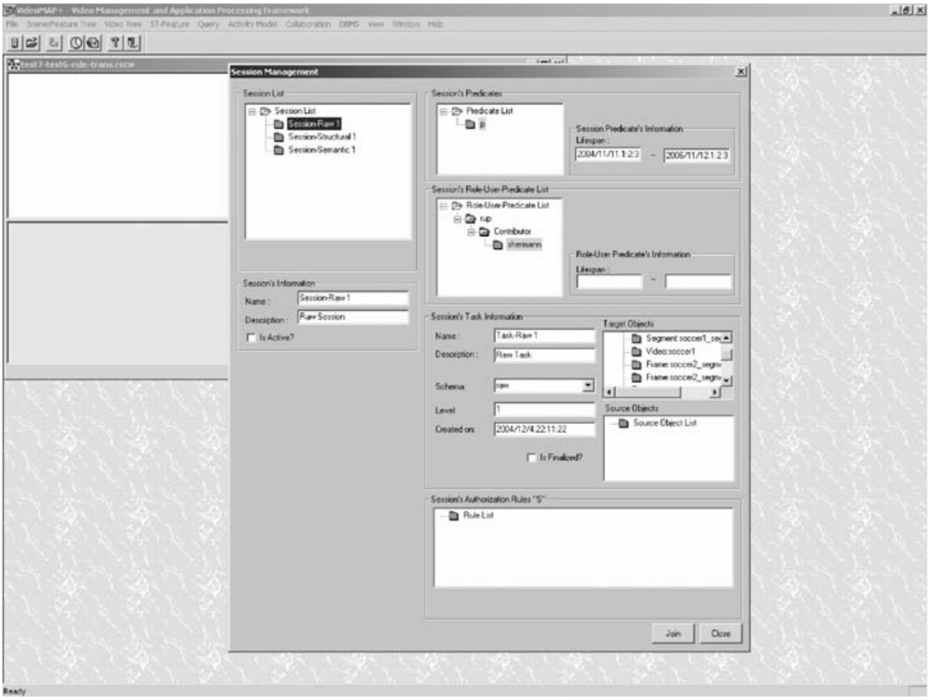
Database : D:\Project\20041202df-demo\test

Fig. 10 User authentication and authorization

4.1.3. Predicate

The Predicate is defined as $P \subseteq (P_V \cup P_T)$, where P_V represents valid-time predicates, and P_T represents transitive predicates. A predicate P can be found in the specification of authorization rules, sessions, and role-user relations within a session. It is used to specify constraints and/or conditions for special cases of access control. For example, we can define, with the access-mode matrix, an authorization rule for a group to access a particular level of data if and only if this group has gained the permission from the source level of the data. Therefore, together with the specification of the predicate, we can accommodate pending authorization control, and no extra effort is needed at the application program level. Another example is that a session can be defined with a valid duration of life for a specific task by the temporal constraints [3–5].

Predicate for authorization rules. When specifying authorization rules, we can categorize the predicates into two types: rule-dependent, and client-dependent. For a rule-dependent predicate, we are interested in the temporal constraints of the rules (e.g., the rules are valid only before or after a specified date), and whether the rules are transitive or non-transitive (e.g., the data–data authorization rule allows to access from one level of object to another level of object). In the specification of a subject–object authorization rule, it can be defined as client–data and data–data access controls, so it is necessary to provide an explicit way to determine the transitive nature of the rules. For a client-dependent predicate, it can be used to create pending


VideoMAP+ - Video Management and Application Processing Framework
File Scene/Feature Tree Video Tree ST-Feature Query Activity Model Collaboration DBMS View Window Help
test7-test6-rule-trans.cscw
Session Management
Session List
Session List
Session-Raw 1
Session-Structual 1
Session-Semantic 1
Session's Predicates
Predicate List
p
Session's Role-User Predicate List
Role-User-Predicate List
rup
Contributor
shermann
Role-User Predicate's Information
Lifespan :
Session's Task Information
Target Objects
Segment:soccer1_se
Video:soccer1
Frame:soccer2_segm
Frame:soccer2_segm
Source Objects
Source Object List
Session's Authorization Rules "S"
Rule List
Join
Close
Ready

Session Predicate's Information
Lifespan : 2004/11/11.1:2:3 – 2006/11/12.1:2:3

Session's Information
Name : Session-Raw 1
Description : Raw Session
□ Is Active?

Name : Task-Raw 1
Description : Raw Task
Schema raw
Level 1
Created on: 2004/12/4.22:11:22
□ Is Finalized?

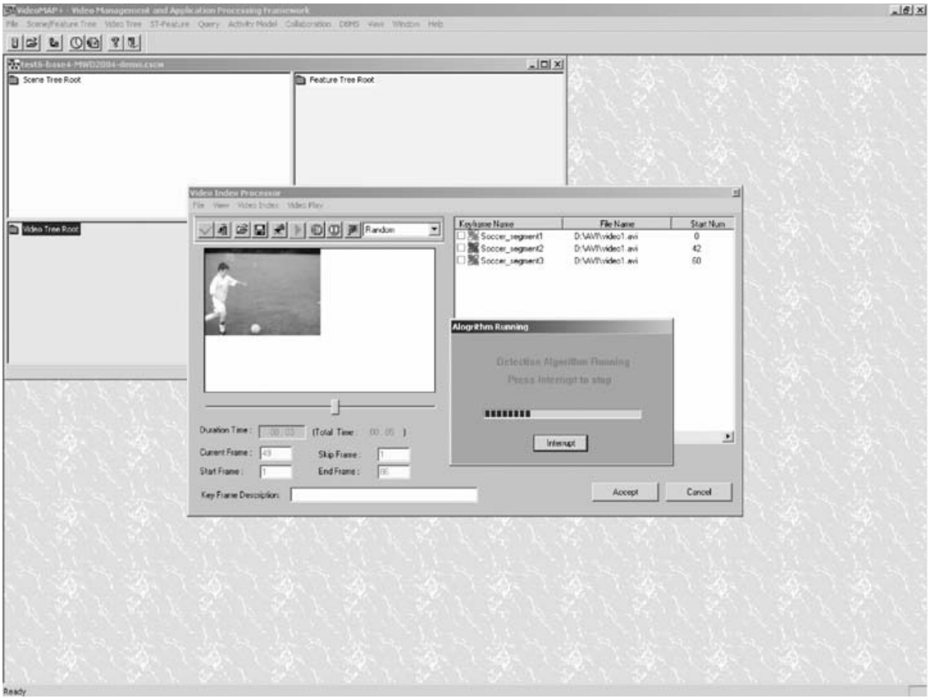
Fig. 11 Join an authorized session

authorization rules for clients; therefore, the rule is valid if and only if it acquires sufficient access permission from the specified clients.

Predicate for specification of clients. When specifying sessions and session–role–user relations, we are interested in their temporal constraints. Therefore, we can specify valid duration of life for a specific task by the session's predicate. We can also temporarily block some users' activities during the video production process by a session–role–user's predicate.

4.2. Mastering authorization rules

In our proposed access control mechanism, there are three kinds of access control properties that can be defined: Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Non-Discretionary Access Control (NAC). DAC can be realized by a client–data authorization rule, whose subject S is: $(\{C_{Gi}:C_{Uj}:C_{Rk}\}|C_{Gi}\in C_G\wedge C_{Uj}\in C_U\wedge C_{Rk}\in C_R)\cup(\{C_{Si}:C_{Uj}:C_{Rk}\}|C_{Si}\in C_S\wedge C_{Uj}\in C_U\wedge C_{Rk}\in C_R)$, where C_{Gi} is a group instance, C_{Si} is a session instance, C_{Uj} is a user instance, and C_{Rk} is a role instance. NAC can be realized by a client–data authorization rule, whose subject S is: $(\{C_{Gi}:C_{Rj}\}|C_{Gi}\in C_G\wedge C_{Rj}\in C_R)\cup(\{C_{Si}:C_{Rj}\}|C_{Si}\in C_S\wedge C_{Rj}\in C_R)$, where C_{Gi} is a group instance, C_{Si} is a session instance, and C_{Rj} is a role instance. In addition to NAC, subject S can be defined as: $C_G\cup C_S$, where C_G is a set of groups and C_S is a set of sessions. For the property of MAC, it can be realized by validating both client–data and data–data authorization rules for the corresponding clearance level. Therefore,



Keyframe Name	File Name	Start Num
□ Soccer_segment1	D:\AVI\video1.avi	0
□ Soccer_segment2	D:\AVI\video1.avi	42
□ Soccer_segment3	D:\AVI\video1.avi	60

Detection Algorithm Running
Press Interrupt to stop

Duration Time : 00:03 (Total Time : 00:05)
Current Frame : 49 Skip Frame : 1
Start Frame : 1 End Frame : 86
Key Frame Description:

Fig. 12 Video segmentation during the process of video production

next steps are how to define authorization rules and how to deal with a set of authorization rules.

4.2.1. Authorization rule and constraint

For the client–data authorization rules, there are four kinds of combinations with regard to the values of the transitive predicates and the temporal predicates. In Table 1, the values of the predicates are omitted and represented simply by Yes (Y) and No (N). These rules allow the clients (i.e., Subject $S \subseteq C$) to access the data (i.e., Object O) governing by the constraints such as schema level constraints, permission constraints and temporal constraints as shown in Table 1. Schema level (*SchLevel*) constraint ensures the schema level specified in the Access-Mode matrix *AM* not higher than the schema level of the Object *O*. Permission constraint is the core of the client–data authorization rule with which the client should acquire sufficient permission in either one of the following situations: (1) no permission is required (denoted by N) or (2) acquisition of permission is demanded from the schema level of the Object *O* to the schema level specified in the Access-Mode matrix *AM* (i.e., *O.SchLevel to AM.SchLevel*). In the case of both schema levels are the same (i.e., *O.SchLevel==AM.SchLevel*), it means the rule allows the client to access the Object *O* without the need of any permission. Temporal constraint is used to specify the lifespan of the authorization rule. After validating the rule and its constraints, the client is able to access Object *O* governed by the accessible schema level. Table 1 also shows the priorities of the client–data authorization rules.

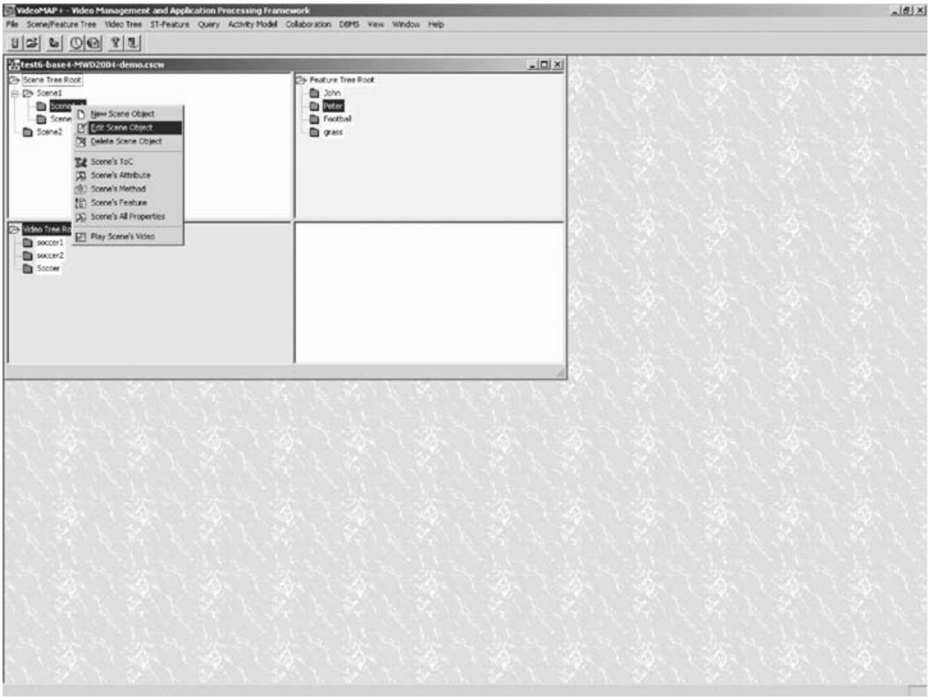
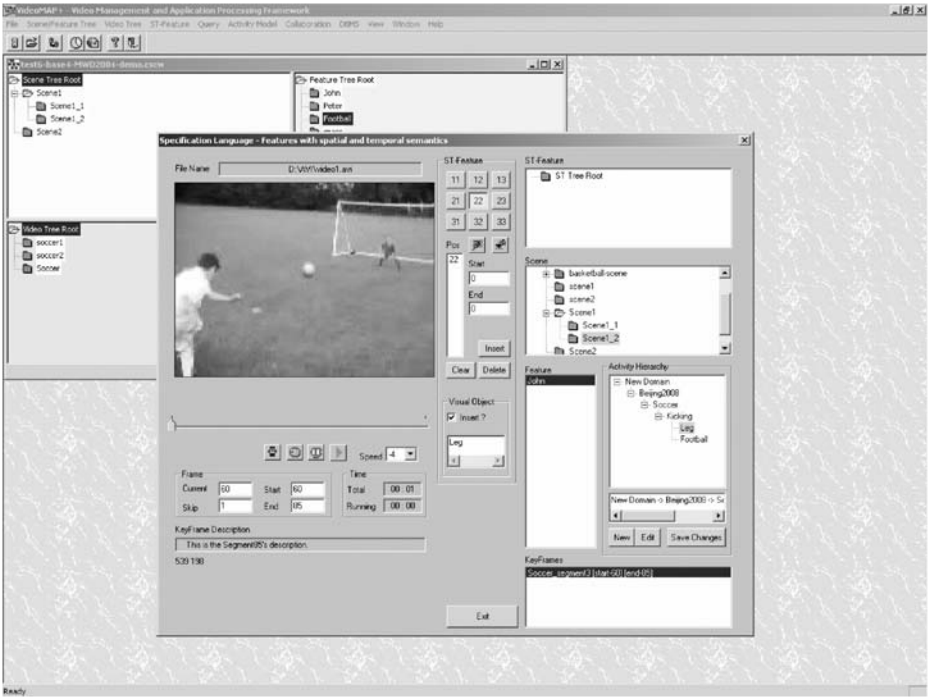


Fig. 13 Construction of scene and semantics during the process of video production

For the data–data authorization rules, there are four kinds of combinations with regard to the values of the transitive predicates and the temporal predicates as well. Similarly, the values of the predicates are omitted and represented by Yes (Y) and No (N) in Table 2. These rules allow the data (i.e., Subject $S \subseteq V \cup H \cup F$) to access another data (i.e., Object O) governing by the constraints such as schema level constraints, permission constraints and temporal constraints as shown in Table 2. Schema level (*SchLevel*) constraint is used to ensure the schema level of the Subject S higher than the schema level of the Object O and the schema level specified in the Access-Mode matrix AM not higher than the schema level of the Subject S. Permission constraint is the core of the data–data authorization rule with which the Subject S could access the Object O in one of the following situations: (1) the rule is non-transitive and access is denied (denoted by N) or (2) the rule is transitive but it requires acquisition of permission from the schema level of the Subject S to the schema level specified in the Access-Mode matrix AM (i.e., *S.SchLevel to AM.SchLevel*). If both schema levels are the same (i.e., *S.SchLevel==AM.SchLevel*), that means no acquisition of permission is required. However, if there exists a similar but non-transitive rule, the Subject S cannot access the Object O finally. As the validation of a data–data authorization rule is originally triggered by a client–data access, the permission constraint is logically and dynamically transformed from data–data to client–data in the end. Temporal constraint is used to specify the lifespan of the authorization rule. After validating the rule and its constraints, Subject S is able to access Object O governed by the accessible schema level. The priorities of the data–data authorization rules are shown in Table 2 and the



File Name: D:\ViVi\video1.avi

Pos: 22; Start: 0; End: 0

☑ Insert ?

Leg

Speed: 4

Frame — Current: 60; Skip: 1; Start: 60; End: 85

Time — Total: 00 : 01; Running: 00 : 00

KeyFrame Description: This is the Segment05's description.

539 198

Feature: John

New Domain -> Beijing2008 -> Sc

KeyFrames: Soccer_segment3 [start-60] [end-85]

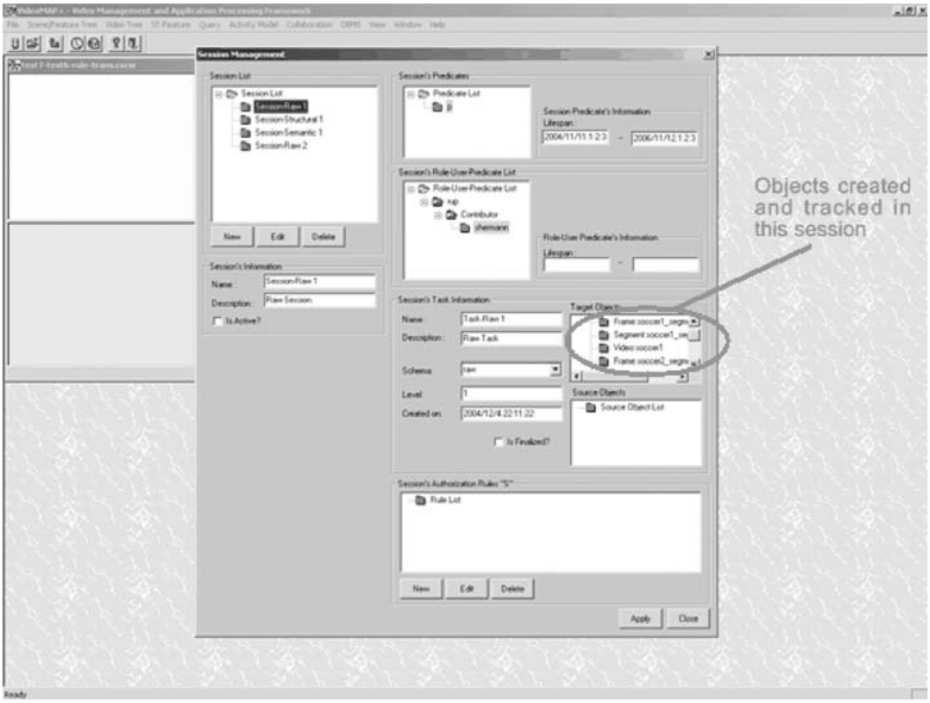
Fig. 14 Visual object and spatio-temporal data specification during video production process

priorities of the client–data authorization rules are superior to that of the data–data authorization rules.

4.2.2. Processing authorization rules

When a user C_{Uj} tries to access an object instance O_m, the access control mechanism obtain a set of session–user–role associations $\{C_{Si}{:}C_{Uj}{:}C_{Rk}\}$ and a set of group–user–role associations $\{C_{Gi}{:}C_{Uj}{:}C_{Rk}\}$. Session C_{Si} is the key linkage between C_{Uj} and O_m, where C_{Uj} belongs to C_{Si} with a set of roles C_R, and O_m is owned and accessible by C_{Si}. Thus, the authorization rules with subject C_{Si} and object O_m are validated first and then followed by those authorization rules with subject $\{C_{Si}{:}C_{Rj}\}$, $\{C_{Si}{:}C_{Uj}{:}C_{Rk}\}$, C_G, $\{C_{Gi}{:}C_{Rj}\}$ and $\{C_{Gi}{:}C_{Uj}{:}C_{Rk}\}$. The valid-time restrictions of the authorization rules (incl. client–data access and data–data access) and the clients (incl. session and session–role–user associations) can be validated by the temporal predicates which are specified in $<S,O,AM,P>$ and $<C_{Si},\{C_{Rj}{:}C_{Uk},P_{j:k}\},P_i>$, respectively.

The second type of validation is to check whether the client C_{Uj} needs to acquire permissions from the source objects, i.e., C_{Si}'s parent sessions $\{C_{Sp}\}$ that own the source objects for the construction of the object instance O_m (ref. Section 4.1.1 for $\{SrcLnk\}$ of the data structure of collaborative task) by comparing the schema level of the object instance O_m and the access schema level specified in the access mode specification. This validation can be figured out by checking whether the client C_{Uj} belongs to the parent sessions $\{C_{Sp}\}$ by $\{<C_{Sp},\{C_{Rj}{:}C_{Uk},P_{j:k}\},P_p>\}$, whether the client C_{Uj} can access the source objects by $<S,O,AM,P>$, or whether the object instance


Session's Information
Name: Session-Raw 1
Description: Raw Session
□ Is Active?

Session Predicate's Information
Lifespan: 2004/11/11 1:2:3 - 2006/11/12 1:2:3

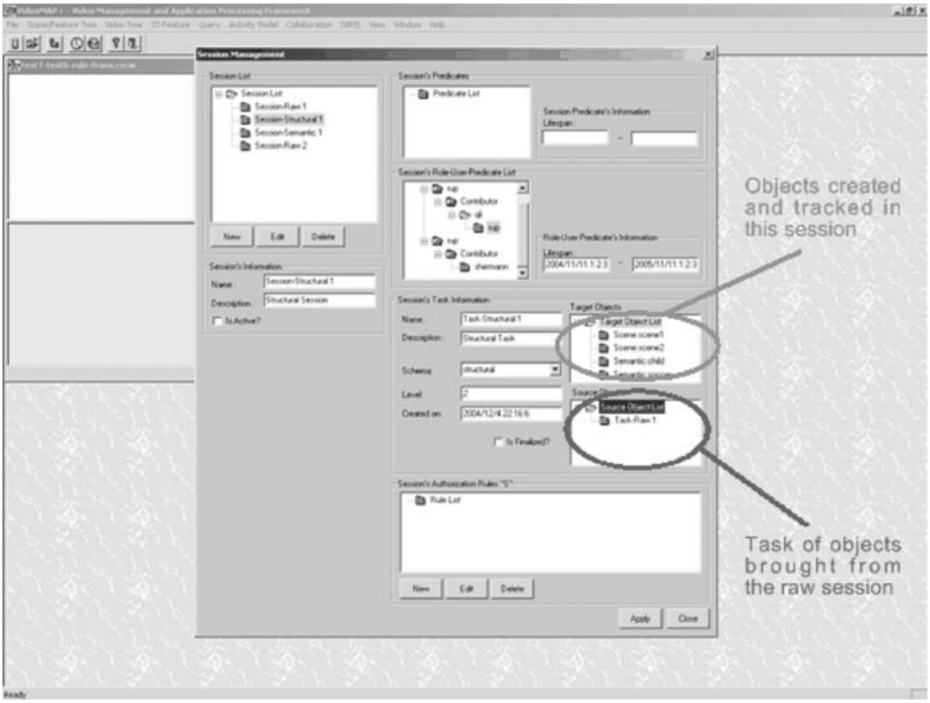
Session's Task Information
Name: Task-Raw 1
Description: Raw Task
Schema: raw
Level: 1
Created on: 2004/12/4 22:11:22
□ Is Finalized?

Fig. 15 Raw session and task management after object tracking

O_m can access its source objects transitively by <*S,O,AM,P*>. This type of iteration will proceed until no further requests (ref. figure 3).

4.3. Scenario and intellectual property

In the case of video production, if there is a cameraman who contributes a video stream to the system, a new session (e.g., C_{Si}) will be formed with two roles (i.e., coordinator, and contributor with *IP* = 1) attached to the cameraman. If there is a group of cameramen who contribute a video stream to the system, a new session (e.g., C_{Sj}) will be formed and each of the cameramen should be assigned with the role of contributor, with *IP* assigned by negotiation (the negotiation process can be taken by a separate communication module) or by dynamic *IP* estimation (ref. Section 4.3.1). One of the cameramen should be assigned to be the coordinator by negotiation too. Both cases (i.e., C_{Si} and C_{Sj}) will create Raw Video objects (e.g., V_{Ri} by C_{Si}, and V_{Rj} by C_{Sj}) and which are stored into the Raw Schema. Meanwhile, an authorization rule for each session will be generated by mapping the session object (e.g., C_{Si}) to the *Subject*, with a raw video object (e.g., V_{Ri}) being the *Object*. These sessions can also predefine who can access these raw video objects either by adding roles of observers in their sessions or by creating extra authorization rules with the potential groups/sessions being the *Subject*. In order to have a strict access control model, authorization rules are used to validate whether the clients can access the data object with read and/or execute modes. But for more critical access modes (e.g., write and delete), both the validation of the authorization rules and the



Session's Information
Name : Session-Structural 1
Description : Structural Session
□ Is Active?

Lifespan : 2004/11/11 1:2:3 – 2005/11/11 1:2:3

Session's Task Information
Name : Task-Structural 1
Description : Structural Task
Schema : structural
Level : 2
Created on : 2004/12/4 22:16:6
□ Is Finalized?

Objects created and tracked in this session

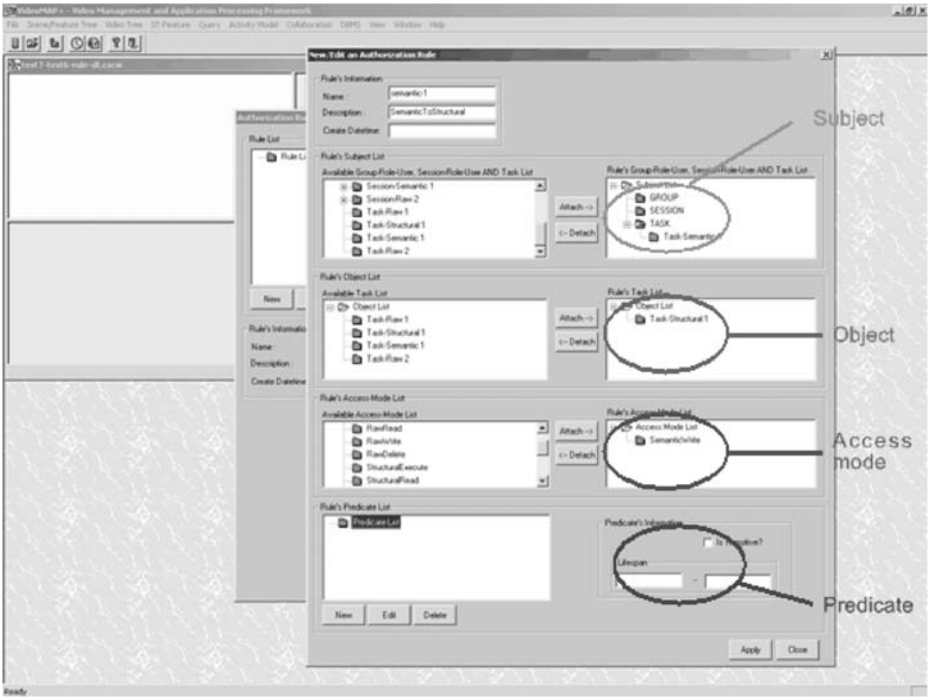
Task of objects brought from the raw session

Fig. 16 Structural session and task management after object tracking

validation of the data object itself will be performed. As a result, our model subsumes the properties of DAC, MAC and NAC [3–5].

For the scientists (e.g., groups of programmers and mathematicians), they store their algorithms and procedures in the Raw Schema of the Video Schema Hierarchy (VSH). The role and *IP* assignment strategy for a scientist is the same as that for cameraman. A video producer and video editors can form a session in order to construct the Semantic Schema of VSH. The video producer should have two roles, viz., coordinator and contributor. Each of the video editors should have the role of contributor. Again, *IP* distribution is negotiated or estimated among the contributors. However, this session (e.g., C_{Sk}) should gain sufficient permission from the corresponding data sources (e.g., *V* and *F*). This can be done by adding a pending authorization rule with a predicate specifying the request. After granting the permission to this session, the predicate will be removed. Otherwise, this rule will be kept to prevent mischievous pranks.

For the sessions mentioned above, they can predefine a number of observers. Thus, within a group/session, there are different types of roles, which have different sets of access modes. In general, a coordinator C_R^L can access data by the modes of $\{M_X, M_R, M_D\}$, a contributor C_R^C can have the modes of $\{M_X, M_R, M_W, M_D\}$, and an observer C_R^O can have the modes of $\{M_X, M_R\}$. Only contributors can have write mode due to the need to centralize the contribution of the work and to evaluate their effort distribution for *IP*. For other roles such as administrators, they are out of the scope of a collaborative video production process, hence their discussions are omitted from this paper.



New/Edit an Authorization Rule

Rule's Information

Name : semantic-1

Description : SemanticToStructural

Create Datetime :

Rule's Subject List

Available Group-Role-User, Session-Role-User AND Task List: Session-Semantic 1, Session-Raw 2, Task-Raw 1, Task-Structural 1, Task-Semantic 1, Task-Raw 2

Rule's Group-Role-User, Session-Role-User AND Task List: SubjectList, GROUP, SESSION, TASK, Task-Semantic1

Rule's Object List

Available Task List: Object List, Task-Raw 1, Task-Structural 1, Task-Semantic 1, Task-Raw 2

Rule's Task List: Object List, Task-Structural 1

Rule's Access Mode List

Available Access Mode List: RawRead, RawWrite, RawDelete, StructuralExecute, StructuralRead

Rule's AccessMode List: Access Mode List, SemanticWrite

Rule's Predicate List: Predicate List

Predicate's Information: □ Is Negative?

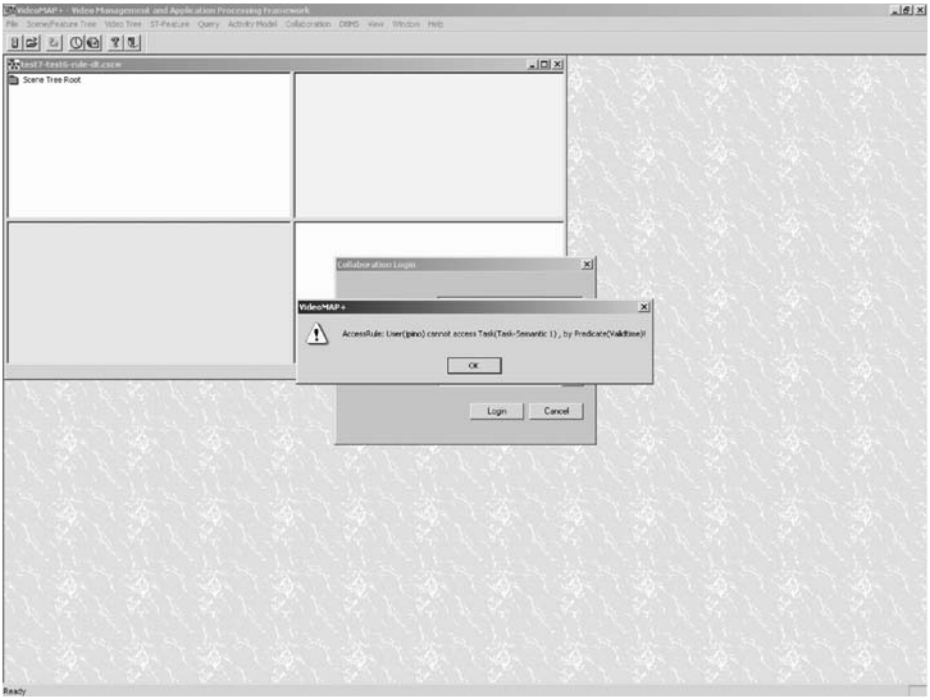
Lifespan

Fig. 17 Define a quadruplet authorization rule <*S, O, AM, P*>

4.3.1. Dynamic IP estimation

Distribution of intellectual property (*IP*) can also be estimated by our proposed mechanism. A module has been developed to keep track of the objects being created and/or being manipulated in any session. A coordinator can dynamically define a set of weights, $W \subseteq (W_S \cup W_O \cup W_M)$, for one session where $W_S = \left(W_S^R \cup W_S^V \cup W_S^H\right)$ represents a set of weights assigned for the raw schema type W_S^R, structural schema type W_S^V, and semantic schema type W_S^H; where $W_O = \left(W_O^{VR} \cup W_O^{VS} \cup W_O^{VK} \cup W_O^{VM} \cup W_O^{VF} \cup W_O^{VO} \cup W_O^{HS} \cup W_O^{HF} \cup W_O^{HM} \cup W_O^{FV} \cup W_O^{FK} \cup W_O^{FF} \cup W_O^{FE} \cup W_O^{FO}\right)$ represents a set of weights for such object types as raw video type W_O^{VR}, segment type W_O^{VS}, keyframe type W_O^{VK}, MBR type W_O^{VM}, visual feature type W_O^{VF}, visual object type W_O^{VO}, scene type W_O^{HS}, semantics type W_O^{HF}, semantic-visual map type W_O^{HM}, video segmentation function type W_O^{FV}, keyframe extraction function type W_O^{FK}, frame segmentation function type W_O^{FF}, feature extraction function type W_O^{FE}, and object recognition function type W_O^{FO}; and where $W_M = \left(W_M^C \cup W_M^E\right)$ represents a set of weights for the operation modes such as create operation W_M^C, and edit operation W_M^E.

In brief, contribution of a contributor can be measured by the number of objects s/he created and/or manipulated. With the adjustable sets of weights for different types of data objects, schema and operation modes, further estimation can be made and varied according to the requirements of the session and in light of the coordinator's point of view. Hence, distribution of the intellectual property can be



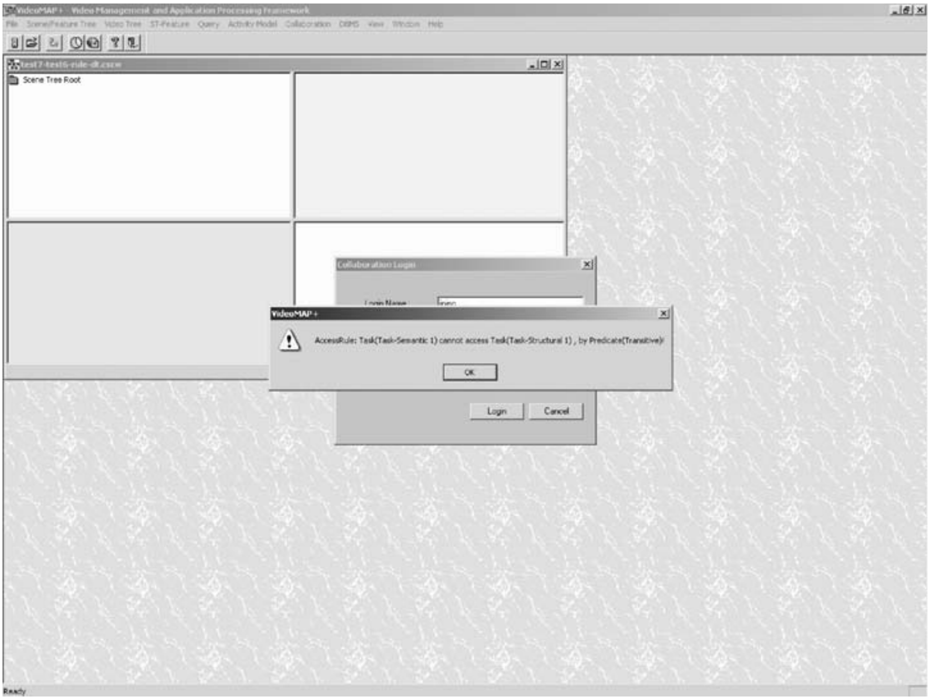
AccessRule: User(pino) cannot access Task(Task-Semantic 1) , by Predicate(Validtime)!

Fig. 18 Access denied by the temporal predicate of the authorization rule

estimated semi-automatically and dynamically from a quantitative approach to a qualitative approach. The *IP* measurement of a session, in turn, can be applied to several sessions whenever the sets of weights (esp. weights for cross sessions e.g., weights for schema types) are well defined.

Given N number of contributors involved in a particular session (say, k), the estimation of the intellectual property of a particular contributor i in session k is: $IP^{ik} = \left(\sum_j \left(C_j^{ik} \times W_{Oj}^k \times W_M^{Ck} + E_j^{ik} \times W_{Oj}^k \times W_M^{Ek}\right)\right)\left(\sum_{i=1}^N \sum_j \left(C_j^{ik} \times W_{Oj}^k \times W_M^{Ck} + E_j^{ik} \times W_{Oj}^k \times W_M^{Ek}\right)\right)^{-1} \Big| j \in \left(V_R \cup V_S \cup V_K \cup V_M \cup V_F \cup V_O \cup H_S \cup H_F \cup H_M \cup F_{VS} \cup F_{KE} \cup F_{FS} \cup F_{FE} \cup F_{OR}\right),$ where C_j^{ik} and E_j^{ik} represent the number of objects with type j created by contributor i in session k, and the number of objects with type j manipulated by contributor i in session k, respectively, W_{Oj}^k represents the weight of the object type j in session k, W_M^{Ck} and W_M^{Ek} represent the weights of the "create" and "edit" operation modes in session k, respectively.

Given N number of contributors involved in a finite set of subsequent sessions *{k}*, the estimated intellectual property of a particular contributor i among sessions *{k}* is: $IP^i = \left(\sum_k \left(IP^{ik} \times W_s^k\right)\right)\left(\sum_{i=1}^N \sum_k \left(IP^{ik} \times W_s^k\right)\right)^{-1}$, where IP^{ik} represents the intellectual property of the contributor i in session k, and W_S^k represents the weight of the schema type of session k.



AccessRule: Task(Task-Semantic 1) cannot access Task(Task-Structural 1) , by Predicate(Transitive)!

Fig. 19 Access denied by the transitive predicate of the authorization rule

5. Experimental prototype and implementation issues

We have implemented our proposed transitive and temporal access control model into one of our experimental prototypes, viz. VideoMAP$^+$ [8]. VideoMAP$^+$ is a video database management system which provides a platform for video segmentation, keyframe selection, construction of scenes and semantics, specification of visual objects and the associated spatio-temporal data, and visual feature extraction. We distinguished the processes of video database production and introduced the concept of collaborative task into VideoMAP$^+$. The new prototype (namely, VideoAcM) embedded with the collaborative access control was developed using MS Visual C++ and NeoLogic NeoAccess Object-oriented Database Engine, upon MS Windows.

In a collaborative environment, users can belong to a number of groups and sessions. Within a group or a session, a user can have several roles. A session offers a virtual workplace for a cluster of users to do a common task. In VideoAcM, a task is defined in a session in order to keep track of the objects created/manipulated by the users. Objects involved in a task can then be brought into another session for further processing. Therefore, the scope of the authorization rules can be defined to a cluster of objects rather than individual objects. Those objects being considered for object tracking are: video, segment, frame, scene, semantics, visual object and spatio-temporal feature. A tradeoff exists between the performance and the granularity of access control; therefore, a user can access some objects such as keyframes, and in turn, s/he can also access the associated composite objects (e.g., visual features). The access control of groups, sessions, roles, and users can be further restricted by the temporal predicates as well. Figures 6, 7, 8, 9 demonstrate the processes of video database and session management of the proposed system such as static group management (ref. figure 6), dynamic session management (ref. figure 7), collaborative user management (ref. figure 8), and role management (ref. figure 9). Figure 10 shows the process of user authentication and authorization. The GUI dialog shown in figure 11 allows the user to join an authorized session. Figures 12, 13, 14 illustrate the processes of video production with object tracking, which include video segmentation (ref. figure 12), construction of scene and semantics (ref. figure 13), and visual object and spatio-temporal data specification (ref. figure 14). Figures 15 and 16 list the objects being tracked by the process of object tracking in the raw session (ref. figure 15) and in the structural session (ref. figure 16). Figure 17 displays how to define an authorization rule in VideoAcM, in which the rule composes of subject, object, access mode and predicate. Figures 18 and 19 demonstrate the access control invalidated by the temporal predicate (ref. figure 18) and the transitive predicate (ref. figure 19) in our proposed access control model. A complete video demonstration is available from http://www.cs.cityu.edu.hk/~shermann/Work/work.html.

6. Conclusions

In this paper, we have proposed a transitive and temporal access control mechanism for collaborative video database production applications. Traditional authorization rules are specified by the client–data access control mechanism (where a client can be a group, role, and user). In our model, specifications of authorization rules by both types of client–data and data–data access control are allowed. Moreover, our model is combined with the Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Non-Discretionary Access Control (NAC) models. In addition to the combined features of the proposed model, an innovative transitive and temporal access control mechanism has been incorporated into the proposed data model as well. With the predefined hierarchical levels of data, our model can be applied to further domains other than video database production. Another feature of our system is to organize production tasks together with the Intellectual Property (IP) estimation. Through our proposed mechanism, access control and IP can be systemically managed. For our subsequent research, we are investigating other types of applications whose characteristic scenarios involve the type of client–client access control.

Acknowledgements This work was partially supported by a grant from MECESUP (Chile) No. UCH0109, a grant from Fondecyt (Chile) No. 1040952, and a grant from the Research Grants Council of the HKSAR, China (Project No. CityU 1125/04E).

References

1. Adam NR, Atluri V, Bertino E, Ferrari E (2002) A content-based authorization model for digital libraries. IEEE Trans Knowl Data Eng 14(2):296–315
2. Aref WG, Catlin AC, Fan J, Elmagarmid AK, Hammad MA, Ilyas IF, Marzouk MS, Zhu X (2002) A video database management system for advancing video database research, Proceedings of the International Workshop on Multimedia Information Systems (MIS 2002), Tempe, Arizona, USA, Oct. 30–Nov. 1
3. Bertino E, Bettini C, Ferrari E, Samarati P (1996) A temporal access control mechanism for database systems. IEEE Trans Knowl Data Eng 8(1):67–80

4. Bertino E, Bettini C, Ferrari E, Samarati P (1998) An access control model supporting periodicity constraints and temporal reasoning. ACM Trans Database Syst 23(3):231–285
5. Bertino E, Bonatti PA, Ferrari E (2001) TRBAC: a temporal role-based access control. ACM Trans Inf Syst Secur 4(3):191–233
6. Bertino E, Fan J, Ferrari E, Hacid M-S, Elmagarmid AK, Zhu X (2003) A hierarchical access control model for video database systems. ACM Trans Inf Sys 21(2):155–191
7. Bertino E, Hammad MA, Aref WG, Elmagarmid AK (2000) An access control model for video database systems. In: Agah A, Callan J, Rundensteiner E (eds) Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM 2000), McLean, Virginia, USA, Nov. 6–11
8. Chan SSM, Li Q, Wu Y, Zhuang Y (2002) Accommodating hybrid retrieval in a comprehensive video database management system. IEEE Trans Multimedia 4(2):146–159, June
9. Ellis CA, Gibbs SJ, Rein GL (1991) Groupware: some issues and experiences. Commun ACM 34(1):39–58, Jan
10. Guerrero LA, Fuller DA (2001) A pattern system for the development of collaborative applications. Inf Softw Technol 43(7):457–467, May
11. Role-based Access Control, http://csrc.nist.gov/rbac/
12. Proceedings of the 1st ACM Workshop on Role-based Access Control (RBAC 1995), Gaithersburg, Maryland, USA, Nov. 20–Dec. 2, 1995
13. Proceedings of the 2nd ACM Workshop on Role-Based Access Control (RBAC 1997), Fairfax, Virginia, USA, Nov. 6–7, 1997
14. Proceedings of the 3rd ACM Workshop on Role-Based Access Control (RBAC 1998), Fairfax, Virginia, USA, Oct. 22–23, 1998
15. Proceedings of the 4th ACM Workshop on Role-Based Access Control (RBAC 1999), Fairfax, Virginia, USA, Oct. 28–29, 1999
16. Proceedings of the 5th ACM Workshop on Role-based Access Control (RBAC 2000), Berlin, Germany, July 26–28, 2000
17. Proceedings of the 6th ACM Symposium on Access Control Models and Technologies (SACMAT 2001), Chantilly, Virginia, USA, 2001
18. Sandhu RS (1998) Role-based access control, advances in computers: the engineering of large systems. In: Zelkowitz MV (ed) Academic, pp 238–285, September
19. Sandhu RS (2001) Future directions in role-based access control models. In: Gorodetski VI, Skormin VA, Popyack LJ (eds) Proceedings of the international workshop on information assurance in computer networks: Methods, Models, and Architectures for Network Security (MMM-ACNS 2001), LNCS 2052, St. Petersburg, Russia, May 21–23, pp 22–26
20. Sandhu RS, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. IEEE Comput 29(2):38–47
21. Zhao B (2001) Collaborative access control, Article in T-110.501 Seminar on Network Security 2001 (NetSec 2001), Publications in Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology

Shermann S. M. Chan holds a B.A. (Hons) degree in Computing obtained from The Hong Kong Polytechnic University and a Ph.D. degree in Computer Science from the City University of Hong Kong. She has been a postdoctoral researcher with the Department of Computer Science, University of Chile. Currently, she is a Visiting Assistant Professor with the Media Research Institute at the Waseda University, Japan. Her research interests include video and multimedia databases, query language processing and retrieval, Internet search engines, ubiquitous intelligence and wireless sensor networks. She is a member of the IEEE, IEEE Computer Society, and the Hong Kong Web Society.

Qing Li holds a B.Eng. from Hunan University (China), and M.Sc. and Ph.D. degrees from the University of Southern California (USA), all in Computer Science. His main research interests include semantic modeling, distributed OODB design, data warehousing and web mining, flexible workflow management, mobile and multimedia databases, and e-learning systems. He has published over 180 publications in internationally refereed journals and conference proceedings in these areas. Dr Li is actively involved in the research community and acted as organizer/co-organizer for major international conferences including DASFAA, WISE, WAIM and VLDB. He has been a co-Guest Editor for Information Sciences (Elsevier Sci.), Knowledge and Information Systems (Springer), co-edited two special issues for World Wide Web journal (Kluwer), and is organizing a special issue on Distributed Media for IEEE Transactions on Multimedia. As a steering committee member of the WISE Society (http://www.i-wise.org/), Dr. Li is the Chairman of the Hong Kong Web Society. He is a senior member of IEEE, and is listed in Who's Who in Science and Engineering by Marquis Who's Who.

José A. Pino is an Associate Professor of Computer Science and Director of the PhD Program in Computer Science at the Universidad de Chile. His research interests include Computer-Supported Cooperative Work, Human-Computer Interaction and Software Industry Studies. He has served as President of the Chilean Computer Science Society (SCCC) and President of CLEI (the Latin American Association of Universities concerning Information Technology). He has co-authored six books and published research papers in international conferences and journals, including Journal of the ACM, Communications of the ACM, Decision Support Systems, Interacting with Computers and Information Technology and People.