A Semantic Modeling Approach for Video Retrieval by Content[*]

Edoardo Ardizzone

Computer Science & Artificial Intelligence Lab.
Dipartimento di Ingegneria Elettrica
Universita' di Palermo, Viale delle Scienze
90128 Palermo, Italy
ardizzone@diepa.unipa.it

Mohand-Said Hacid[†]

LuFG Theoretical Computer Science
RWTH Aachen
Ahornstraße 55, 52074 Aachen, Germany
hacid@cantor.informatik.rwth-aachen.de

Abstract

A knowledge-based approach to model and retrieve video data by content is developed. Selected objects of interest in a video sequence are described and stored in a database. This database forms the object layer. On top of this layer, we define the schema layer used to capture the structured abstractions of the objects stored in the object layer. We propose two abstract languages on the basis of description logics: one for describing the contents of these layers, and the other, more expressive, for making queries. The query language provides possibilities for navigation of the schema through forward and backward traversal of links, sub-setting of attributes, and constraints on links.

1 Introduction

Video data management poses special challenges which call for new techniques allowing an easy development of applications. These techniques require multi-disciplinary research effort in areas such as computer vision, image processing, data compression, databases, artificial intelligence, information systems, etc. (see [9]). To facilitate retrieval, all useful semantic objects and their features appearing in the video must be *appropriately* indexed. The use of keywords or free text [10] to describe the necessary semantic objects is not sufficient. Additional techniques are needed. As stated in [4], some of the issues that need to be addressed are: (1) the representation of video information in a form that facilitates retrieval and interaction, (2) the organization of this information for efficient manipulation, and (3) the user-friendly presentation of the retrieved video sequences. Being able to derive an adequate content description from a video, however, does not guarantee a satisfactory retrieval effectiveness, it is only a necessary condition to this end.

In this paper we exploit the possibility of using two languages: one for defining the schema (i.e. the structure) of a video database and populating it, and the other for querying the database through the schema. The query language, which is more powerful than the schema language, provides possibilities for navigation of the schema through *forward* and *backward* traversal of links (attributes and relations), *sub-setting* of attributes, and *constraints* on links.

We propose two layers for representing video's conceptual content:

(1) *Object Layer:* This layer contains objects of interest, their descriptions, and relationships among objects based on the extracted features[1]. This layer constitutes what we call the extensional part of a video database. Objects in a video sequence are represented in the object layer as *visual entities*. Instances of visual objects consist of conventional attributes (e.g. name, actorID, sequence_duration, etc.).

(2) *Schema Layer:* This layer is intended to capture the structured abstractions and knowledge that are needed for video retrieval. It contains a general schema about the classes of objects stored in the object layer, their general properties and mutual relationships. In this

---

[*] This work is partially supported by France Télecom (through CNET/CCETT) research contract no. 96-ME-17, and the French Ministry of Foreign Affairs and the Italian Ministry of Science and Technology Project Galilee'99.

[†] Work supported in part by the Deutscher Akademischer Austausch Dienst (DAAD) grant A/98/05793.

[1] Features can be extracted *manually*, *semi-automatically* or *automatically*.

layer, visual entities can be classified into a hierarchical structure known as a concept hierarchy. This layer is well suited for integrating domain knowledge. It enables a user to determine a video's distinguishing content without investing long viewing times or requiring high network transfer speeds.

Paper outline: In Section 2, we develop our languages and give their Tarski-style extensional semantics. Section 3 discusses related work. We conclude in Section 4 by anticipating on the necessary extensions.

2 The Languages

Before we give the syntax and semantics of our *abstract languages*, we define *concrete domains*, which are used to incorporate application-specific domains (i.e., strings, reals, integers, etc.) into the abstract domain of individuals.

Definition 1 (Concrete Domains) *A concrete domain $\mathcal{D} = (\mathsf{dom}(\mathcal{D}), \mathsf{pred}(\mathcal{D}))$ consists of:*

- *the domain $\mathsf{dom}(\mathcal{D})$,*
- *a set of predicate symbols $\mathsf{pred}(\mathcal{D})$, where each predicate symbol $P \in \mathsf{pred}(\mathcal{D})$ is associated with an arity n and an n-ary relation $P^{\mathcal{D}} \subseteq \mathsf{dom}(\mathcal{D})^n$,*

In many applications (in particular when querying databases), one would like to be able to refer to concrete domains and predicates on these domains when defining queries. An example of such a concrete domain could be the set of (nonnegative) integers with comparisons ($=, <, \leq, \geq, >$).

2.1 Schema Language ($\mathcal{SL}$)

We now introduce a simple description logic that will be used for describing the structure of a video document. Starting from atomic concepts and roles, complex concepts are built by using the universal quantification ($\forall$) and predicate restrictions.

The syntax and the semantics of this description logic are given below.

Definition 2 (Syntax) *Let N_C, N_R, N_f be three pairwise disjoint sets of concept names, role names, and feature (i.e., functional role) names respectively, $\mathcal{D}_1, \ldots, \mathcal{D}_k$ be concrete domains. Let P be a role name, $f, f_1, \ldots, f_n$ be feature names, A be a concept name, A' be a concept name or a concrete domain name, and P_r be an n-ary predicate name. Concept terms C, D are defined by the following rules:*

$$\begin{array}{lcll}
C, D & \longrightarrow & \top & \textit{(universal concept)} \\
 & & A \mid & \textit{(primitive concept)} \\
 & & \forall P.A \mid & \textit{(typing of role)} \\
 & & \forall f.A' \mid & \textit{(typing of feature)} \\
 & & P_r(f_1, \ldots, f_n) & \textit{(predicate restriction)}
\end{array}$$

Let A, A_1, A_2 be concept names, A_3 be a concept name or a concrete domain name, D be a concept term, P be a role name, and f be a feature name. Then $A \dot{\preceq} D$ (we say A is a subconcept of D), $P \dot{\preceq} A_1 \times A_2$, $f \dot{\preceq} A_1 \times A_3$ are called *axioms*.

A $\mathcal{SL}$ schema $\mathcal{S}$ consists of a finite set of axioms. In the following, we consider only *acyclic* schemas. A schema $\mathcal{S}$ is acyclic if no concept name occurs–neither directly nor indirectly–within its own specification.

Definition 3 (Semantics) *The semantics is given by an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, which consists of an (abstract) interpretation domain $\Delta^{\mathcal{I}}$, and an interpretation function $\cdot^{\mathcal{I}}$. The abstract domain has to be disjoint from any given concrete domain, i.e., $\Delta^{\mathcal{I}} \cap \mathsf{dom}(\mathcal{D}_i) = \emptyset$ for all concrete domain $\mathcal{D}_i$ ($i \in [1, k]$), the concrete domains are pairwise disjoint, and $\mathsf{pred}(\mathcal{D}_i) \cap \mathsf{pred}(\mathcal{D}_j) = \emptyset$ for $i \neq j$. The interpretation function $\cdot^{\mathcal{I}}$ associates each concept C with a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role P with a binary relation $P^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and each feature name f with a partial function $f^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow (\Delta^{\mathcal{I}} \cup (\bigcup_{i=1}^{k} \mathsf{dom}(\mathcal{D}_i)))$. Additionally, $\mathcal{I}$ has to satisfy the following equations:*

$$\begin{array}{rcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
(\forall P.A)^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \forall d' \in \Delta^{\mathcal{I}} : \\
 & & \qquad (d^{\mathcal{I}}, d'^{\mathcal{I}}) \in P^{\mathcal{I}} \rightarrow d'^{\mathcal{I}} \in A^{\mathcal{I}}\} \\
(\forall f.A')^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \text{ if } f^{\mathcal{I}}(d^{\mathcal{I}}) \text{ is defined then} \\
 & & \qquad f^{\mathcal{I}}(d^{\mathcal{I}}) \in A'^{\mathcal{I}}\} \\
(P_r(f_1, \ldots, f_n))^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid (f_1^{\mathcal{I}}(d^{\mathcal{I}}), \ldots, f_n^{\mathcal{I}}(d^{\mathcal{I}})) \in P_r^{\mathcal{D}}\}
\end{array}$$

An interpretation $\mathcal{I}$ satisfies the axiom $A \dot{\preceq} D$ iff $A^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, the axiom $P \dot{\preceq} A_1 \times A_2$ iff $P^{\mathcal{I}} \subseteq A_1^{\mathcal{I}} \times A_2^{\mathcal{I}}$, and the axiom $f \dot{\preceq} A_1 \times A_3$ iff $f^{\mathcal{I}} \subseteq A_1^{\mathcal{I}} \times A_3^{\mathcal{I}}$. If A_3 is a concrete domain name then $A_3^{\mathcal{I}}$ stands for the domain of A_3 (i.e., $\mathsf{dom}(A_3)$) for all $\mathcal{I}$. In the following, individuals of the abstract domain are called *abstract individuals* and those of a concrete domain are called *concrete individuals*.

Definition 4 (Model) *An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ is a model, also called a valid interpretation, of a schema $\mathcal{S}$ iff it satisfies every axiom in $\mathcal{S}$.*

If $\mathcal{S}$ is a schema, an interpretation $\mathcal{I}$ that satisfies all axioms in $\mathcal{S}$ is called an $\mathcal{S}$-interpretation.

Example 1 *Figure 1 shows a simple fragment of the structure of a video database. Each inclusion assertion (introduced by $\dot{\preceq}$) imposes a constraint on the instances of the class it refers to. Concrete domains required here are the sets of integers and strings.*

$Person \mathrel{\dot{\preceq}} \forall name.\mathtt{STRING}$

$Producer \mathrel{\dot{\preceq}} Person$
$Producer \mathrel{\dot{\preceq}} \forall produced\_films.Film$

$Director \mathrel{\dot{\preceq}} Person$
$Director \mathrel{\dot{\preceq}} \forall directed\_films.Film$

$Actor \mathrel{\dot{\preceq}} \forall name.\mathtt{STRING}$
$Actor \mathrel{\dot{\preceq}} \forall address.\mathtt{STRING}$
$Actor \mathrel{\dot{\preceq}} \forall played\_in.Film$

$VideoDocument \mathrel{\dot{\preceq}} \forall structuredby.StructuralComponent$
$VideoDocument \mathrel{\dot{\preceq}} \forall representedby.VideoStream$
$\ldots$

$Film \mathrel{\dot{\preceq}} \forall title.\mathtt{STRING}$
$Film \mathrel{\dot{\preceq}} \forall directedBy.Director$
$Film \mathrel{\dot{\preceq}} \forall producedBy.Producer$
$Film \mathrel{\dot{\preceq}} \forall date.\mathtt{INTEGER}$
$Film \mathrel{\dot{\preceq}} \forall duration.\mathtt{INTEGER}$
$Film \mathrel{\dot{\preceq}} \forall actor.Actor$

$Sequence \mathrel{\dot{\preceq}} \forall entities.\top$
$Sequence \mathrel{\dot{\preceq}} \forall event.\mathtt{STRING}$
$Sequence \mathrel{\dot{\preceq}} \forall film.Film$
$Sequence \mathrel{\dot{\preceq}} \forall actor.Actor$
$Sequence \mathrel{\dot{\preceq}} \forall sequence\_scene.Scene$
$\ldots$

Figure 1. A fragment of video database structure

The language introduced previously allows to describe knowledge about classes of individuals and relationships between these classes. We can now turn our attention to the extensional level (i.e., description of individuals).

Definition 5 *Let N_I and N_D be two disjoint alphabets of symbols, called abstract individual names and concrete individual names respectively. Instance-of relationships are expressed in terms of membership assertions of the form:*

$$a : C,\ (a, b) : P,\ (a, b) : f,\ (a, z) : f,\ (z_1, \ldots, z_n) : P_r$$

where a and b are abstract individual names, $z, z_1, \ldots, z_n$ are concrete individual names, C is a concept name or an arbitrary concept, P is a role name, and P_r is an n-ary predicate name of a concrete domain.

In order to assign a meaning to membership assertions, the extension function $.^{\mathcal{I}}$ of an interpretation $\mathcal{I}$ is extended to individuals by mapping them to elements of $\Delta^{\mathcal{I}}$ in such a way that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ iff $a \neq b$ (Unique Name Assumption). For concrete individuals, the unique name assumption does not hold.

An interpretation $\mathcal{I}$ satisfies the assertion:

$$a : C \text{ iff } a^{\mathcal{I}} \in C^{\mathcal{I}}, \quad (a, b) : P \text{ iff } (a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}},$$

$$(a, b) : f \text{ iff } f^{\mathcal{I}}(a^{\mathcal{I}}) = b^{\mathcal{I}}, \quad (a, z) : f \text{ iff } f^{\mathcal{I}}(a^{\mathcal{I}}) = z^{\mathcal{I}},$$

$$(z_1, \ldots, z_n) : P_r \text{ iff } (z_1^{\mathcal{I}}, \ldots, z_n^{\mathcal{I}}) \in P_r^{\mathcal{D}}$$

An $ABox$ $\mathcal{A}$ is a finite set of membership assertions.

An interpretation $\mathcal{I}$ is a model for an $ABox$ $\mathcal{A}$ iff $\mathcal{I}$ satisfies all the assertions in $\mathcal{A}$.

Example 2 *On the basis of the schema given Figure 1, we can assert the following facts:*

$\mathsf{gi}_1 : \mathsf{sequence}$, $(\mathsf{gi}_1, \mathsf{o}_1) : \mathsf{actor}$, $(\mathsf{gi}_1, \mathsf{o}_2) : \mathsf{actor}$, $(\mathsf{gi}_1, \mathsf{o}_4) : \mathsf{entities}$, $(\mathsf{gi}_1, \mathsf{E}_1) : \mathsf{event}$, $\mathsf{E}_1 :=_{\text{"murder"}}$, $(\mathsf{gi}_1, \mathsf{o}_1) : \mathsf{victim}$, $(\mathsf{gi}_1, \mathsf{o}_2) : \mathsf{murderer}$, $(\mathsf{gi}_1, \mathsf{o}_3) : \mathsf{murderer}$

The first statement says that gi_1 is a sequence, and the second says that o_1 is a filler for the role actor of the individual (object) gi_1. In the assertion $(\mathsf{gi}_1, \mathsf{E}_1) : \mathsf{event}$, E_1 is a concrete individual name. This concrete individual name is linked to the constant string "murder" through $:=_{\text{"murder"}}$ which stands for the unary predicate $\{\mathsf{S} \mid \mathsf{S} = \text{"murder"}\}$.

2.2 Query Language ($\mathcal{QL}$)

Querying a database means retrieving stored objects that satisfy certain *conditions* or *qualifications* and hence are interesting for a user. In the case of relational databases, queries are constructed by means of algebra expressions defined on relations from the database. As a property, answers are also relations (i.e., sets of tuples). This correspondence between database entities and answer formats presents advantages that lead to the design and development of query optimization techniques. In object-oriented databases, classes are used to represent sets of objects. By analogy with the relational approach, classes can be used for describing query results. If such a possibility exists, then we can consider some kind of *reasoning* on the structure[2] of classes that will lead to reveal, for example,

---

[2] And hence the semantics of class hierarchies.

subsumption relationships between queries.

In our framework, we follow this approach. Queries are represented as concepts in our abstract language.
In the following, we give the syntax and semantics of a concept language for making queries.

Definition 6 (Syntax) *Let A be a concept name, P be an atomic role, d be an abstract individual name, $f_1, \ldots, f_n$ be feature names, and $P_r \in \mathsf{pred}(\mathcal{D}_i)$ for some $i \in [1, k]$ be an n-ary predicate name. Concepts C, D and roles R, R' can be formed by means of the following syntax:*

$$\begin{array}{rl} C, D \longrightarrow & \top \mid A \mid C \sqcap D \mid \exists R.C \mid P_r(f_1, \ldots, f_n) \mid \\ & \{d\} \mid \ominus R\, R'\, C \mid \oplus R\, R' \mid \sqsupseteq R.C \end{array}$$

$$R, R' \longrightarrow \quad P \mid P^- \mid R \circ R' \mid \Theta\, R\, R'$$

Definition 7 (Semantics) *$\mathcal{I}$ is defined as for the definition 3. Additionally, $\mathcal{I}$ has to satisfy the following equations:*

$$\begin{array}{rcl} \top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\ (\exists R.C)^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \exists d' : (d, d') \in R^{\mathcal{I}} \wedge d' \in C^{\mathcal{I}}\} \\ P_r(f_1, \ldots, f_n)^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid (f_1^{\mathcal{I}}(d), \ldots, f_n^{\mathcal{I}}(d)) \in P_r^{\mathcal{D}}\} \\ \{d\}^{\mathcal{I}} & = & \{d^{\mathcal{I}}\} \\ (\ominus\, R\, R'\, C)^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid \exists d' \in C^{\mathcal{I}} \text{ such that} \\ & & \quad R^{\mathcal{I}}(d) \cap R'^{\mathcal{I}}(d') \neq \emptyset\} \\ (\oplus\, R\, R')^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(d) \subseteq R'^{\mathcal{I}}(d)\} \\ (\sqsupseteq R.C)^{\mathcal{I}} & = & \{d \in \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(d) \supseteq C^{\mathcal{I}}\} \\ (R \circ R')^{\mathcal{I}} & = & \{(d, d') \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists c \in \Delta^{\mathcal{I}} \text{ such that} \\ & & \quad (d, c) \in R^{\mathcal{I}} \wedge (c, d') \in R'^{\mathcal{I}}\} \\ (\Theta\, R\, R')^{\mathcal{I}} & = & \{(d, d') \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid R^{\mathcal{I}}(d) \subseteq R'^{\mathcal{I}}(d')\} \end{array}$$

Example 3 *Consider the schema fragment of Figure 1. The intended reading of the query:*

$$\begin{array}{r} \mathsf{Sequence} \sqcap \exists \mathsf{film} \circ \mathsf{directedBy}. =_{\text{"}\mathsf{Kevin\ Costner}\text{"}} (\mathsf{name}) \sqcap \\ \oplus \mathsf{film} \circ \mathsf{directedBy}\ \mathsf{actor} \end{array}$$

would be "Sequences of movies directed by Kevin Costner *in which he is also an actor". And the query*

$$\mathsf{Film} \sqcap \exists(\Theta\ \mathsf{directedBy}\ \mathsf{producedBy}).\top$$

denotes the set of movies whose directors are also producers of some films.

3 Related Work

In the framework of semantic modeling schemes for video retrieval by content, our work relates to several fields of research in databases and Artificial Intelligence. Semantic schemes attempt to model the meaning of video sequences, taking into account a number of aspects connected to what questions may be put, e.g., related to objects, spatial relationships between objects, events and actions involving objects, temporal relationships between events and actions, user interaction. In the following, we shortly discuss the relationship to some of them regarding database support for modeling and querying video data and knowledge representation for video data.

Database Support for Modeling and Querying Video Data. Oomoto and Tanaka [8] proposed a schema-less video-object data model. They focus on the capabilities of object-oriented database features (their extension) for supporting schema evolution and to provide a mechanism for sharing some descriptive data. Adali et al. [1] have developed a formal video data model, and they exploit spatial data structures for storing such data. Hjelsvold and Midtstraum [7] proposed a generic video data model. Their proposal combines ideas from the stratification and the segmentation approaches. Decleir *et al.* [5] proposed a data model and a rule-based constraint query language for video content based indexing and retrieval. The data model is designed around the object and first order constraint paradigms. The query language, with a clear declarative and operational semantics, is used to infer relationships about information represented in the model.

We believe that our work complements database approaches for modeling and querying video data in the sense that we provide powerful mechanisms for schema (and hence meta-data) level reasoning, i.e., reasoning at the intensional level of a video database.

Knowledge Representation for Video Data. Carrive *et al.* [3] addressed the problem of using a description logic for automatically building high-level descriptions from a collection of shots for annotating, indexing and accessing broadcast audiovisual documents. The proposed description logic is an extension of CLASSIC[2] with temporal operations (e.g., *meets*, *before*), which are taken into account by implication rules. Goble *et al.* [6] proposed a description logic, called, GRAIL, for describing the image and video semantic content. A set of dedicated constructors are used to capture the structural part of these media objects. The aim is to support the coherent and incremental development of a coarse index on the semantic annotations of media documents.

First, we provided more sophisticated constructors dedicated for navigation and searching. Second, these two proposals do not take into account predicate restrictions over concrete domains, which are extremely useful when querying multimedia repositories.

4 Conclusion

There is a growing interest in video databases. As video libraries proliferate, aids to browsing and filtering become increasingly important tools for managing such exponentially growing information resources and for dealing with access problems. One of the central problems in the creation of robust and scalable systems for manipulating video information lies in representing video content. We believe that formal settings will help understanding related modeling and querying problems. This will lead to the development of intelligent systems in order to effectively disseminate, integrate, retrieve, correlate, and visualize video information.

We have overviewed a formal design of a framework for capturing the structured abstractions and knowledge that are needed for video retrieval. We have merely laid a formal and flexible framework which is appropriate for modeling and reasoning about meta-data and queries in video databases. Expressiveness and services of the meta data schema are crucial for video database quality[3]. This framework is appropriate for supporting semantic indexing, conceptual and intensional queries.

There are many interesting directions to pursue: (1) Due to the visual nature of video data, a user may be interested in results that are similar to the query, thus, the query system should be able to perform exact as well as partial or fuzzy matching; (2) It seems attractive to extend this work such that it can accommodate space, time and actions, inherent to video data; (3) The interface between the abstract level and concrete domains can be exploited to elegantly consider *form queries*, i.e., queries addressing video sequences on the basis of color, objects' shapes, etc. In this case an additional layer intended to contain such features is required. This layer will be characterized by a set of techniques (algorithms) allowing to retrieve video sequences based on the similarity of physical features of objects they contain.

We are investigating these important research directions.

References

[1] Sibel Adali, Kasim S. Candan, Su-Shing Chen, Kutluhan Erol, and V. S. Subrahmanian. Advanced Video Information System: Data Structures and Query Processing. *ACM-Springer Multimedia System*, 4:172–186, 1996.

[2] Alexander Borgida, Ronald J. Brachman, Deborah L. McGuinness, and Lori A. Resnick. CLASSIC: A Structural Data Model for Objects. In *Proceedings of 1989 ACM SIGMOD International Conference on Management of Data (SIGMOD'89), Portland, Oregon, USA*, pages 59–67, June 1989.

[3] Jean Carrive, Francois Pachet, and Rémi Ronfard. Using Description Logics for Indexing Audiovisual Documents. In Enrico Franconi, Giuseppe De Giacomo, Robert M. MacGregor, Werner Nutt, and Christopher A. Welty, editors, *Proceedings of The International Workshop on Description Logics (DL'98), Povo-Trento, Italy*, pages 116–120, 1998.

[4] Tat-Seng Chua and Li-Qun Ruan. A Video Retrieval and Sequencing System. *ACM Transactions on Information Systems (TOIS)*, 13(4):373–407, October 1995.

[5] Cyril Decleir, Mohand-Said Hacid, and Jacques Kouloumdjian. A Database Approach for Modeling and Querying Video Data. In *Proceedings of the 15th International Conference on Data Engineering (ICDE'99), Sydney, Australia*, March 1999. (To appear).

[6] C. A. Goble, C. Haul, and S. Bechhofer. Describing and Classifying Multimedia Using the Description Logic GRAIL. In Ishwar K. Sethi and Ramesh C. Jain, editors, *Storage and retrieval for image and video database IV (SPIE'96), San Jose, California*, pages 132–143, February 1996.

[7] Rune Hjelsvold and Roger Midtstraum. Modelling and Querying Video Data. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB'94), Santiago, Chile*, 1994.

[8] Eitetsu Oomoto and Katsumi Tanaka. OVID: Design and Implementation of a Video-Object Database System. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):629–643, August 1993.

[9] ACM TOIS. Special Issue in Video Information Systems. *ACM Transactions on Information Systems (TOIS)*, 13(4), October 1995.

[10] Yoshinobu Tonomura. Video Handling Based on Structured Information for Hypermedia Systems. In *Proceedings of the International Conference on Multimedia Information Systems'91, Singapore*, pages 333–344. McGraw Hill, New York, 1991.

---

[3] Quality implies accessibility, optimization, validation, etc.