# Conceptual Modeling of Multimedia Databases

THÈSE N$^{O}$ 4344 (2009)

PRÉSENTÉE LE 24 AVRIL 2009

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

Laboratoire de bases de données

SECTION D'INFORMATIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Oleksandr DRUTSKYY

acceptée sur proposition du jury:

Prof. A. Ailamaki, présidente du jury
Prof. S. Spaccapietra , directeur de thèse
Prof. O. de Troyer, rapporteur
Prof. W. Klas, rapporteur
Prof. D. Thalmann, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2009

# Abstract

## Conceptual Modeling Of Multimedia Databases

The gap between the semantic content of multimedia data and its underlying physical representation is one of the main problems in the modern multimedia research in general, and, in particular, in the field of multimedia database modeling. We believe that one of the principal reasons of this problem is the attempt to conceptually represent multimedia data in a way, which is similar to its low-level representation by applications dealing with encoding standards, feature-based multimedia analysis, etc. In our opinion, such conceptual representation of multimedia contributes to the semantic gap by separating the representation of multimedia information from the representation of the universe of discourse of an application, to which the multimedia information pertains. In this research work we address the problem of conceptual modeling of multimedia data in a way to deal with the above-mentioned limitations.

First, we introduce two different paradigms of conceptual understanding of the essence of multimedia data, namely: *multimedia as data* and *multimedia as metadata*. The *multimedia as data* paradigm, which views multimedia data as the subject of modeling in its own right, is inherent to so-called multimedia-centric applications, where multimedia information itself represents the main part of the universe of discourse. The examples of such kind of applications are digital photo collections or digital movie archives. On the other hand, the *multimedia as metadata* paradigm, which is inherent to so-called multimedia-enhanced applications, views multimedia data as just another (optional) source of information about whatever universe of discourse that the application pertains to. An example of a multimedia-enhanced application is a human-resource database augmented with employee photos. Here the universe of discourse is the totality of company employees, while their photos simply represent an additional (possibly optional) kind of information describing the universe of discourse.

The multimedia conceptual modeling approach that we present in this work allows addressing multimedia-centric applications, as well as, in particular, multimedia-enhanced applications. The model that we propose builds upon MADS (Modeling Application Data with Spatio-temporal features), which is a rich conceptual model defined in our laboratory, and which is, in particular, characterized by structural completeness, spatio-temporal modeling capabilities, and multirepresentation support. The proposed multimedia model is provided in the form of a new modeling dimension of MADS, whose orthogonality principle allows to integrate the new multimedia modeling dimension with already existing modeling features of MADS. The following multimedia modeling constructs are provided: multi-

media datatypes, simple and complex representational constraints (relationships), a multimedia partitioning mechanism, and multimedia multirepresentation features.

Following the description of our conceptual multimedia modeling approach based on MADS, we present the peculiarities of logical multimedia modeling and of conceptual-to-logical inter-layer transformations. We provide a set of mapping guidelines intended to help the schema designer in coming up with rich logical multimedia document representations of the application domain, which conform with the conceptual multimedia schema.

The practical interest of our research is illustrated by a mock-up application, which has been developed to support the theoretical ideas described in this work. In particular, we show how the abstract conceptual set-based representations of multimedia data elements, as well as simple and complex multimedia representational relationships can be implemented using Oracle DBMS.

**Keywords:** multimedia, conceptual database modeling, multimedia databases, MADS.

# Résumé

## Modélisation conceptuelle des bases de données multimédias

Le décalage entre le contenu sémantique des données multimédias et leur représentation physique est l'un des problèmes principaux dans la recherche scientifique sur les données multimédias en général et sur la modélisation des bases de données multimédias en particulier. Nous pensons que l'une des causes essentielles de ce problème vient du fait d'essayer de représenter les données multimédias au niveau conceptuel d'une manière similaire à celle qui est utilisée par les applications qui traitent les données multimédias au bas niveau, comme par exemple encodages multimédias, analyse d'images basée sur les couleurs et les textures, etc. A notre avis, une telle représentation conceptuelle du multimédia aggrave le problème du décalage sémantique en séparant la représentation de l'information multimédia de la représentation de l'univers du discours d'une application à laquelle cette information multimédia s'adresse. Dans le présent travail de recherche nous abordons la problématique de la modélisation conceptuelle des données multimédias de manière à s'attaquer aux limitations mentionnées ci-dessus.

Nous introduisons tout d'abord deux paradigmes de l'interprétation conceptuelle de l'essence des données multimédias, à savoir: *multimédia en qualité de données* et *multimédia en qualité de metadonnées*. Pour le paradigme de *multimédia en qualité de données* les données multimédias représentent le sujet même de la modélisation. Une telle vision s'adresse aux applications dites orientées multimédia, pour lesquelles le multimédia représente la partie principale de l'univers du discours. Des exemples d'applications orientées multimédia sont les collections des images numériques ou encore les vidéothèques numériques. En ce qui concerne le paradigme de *multimédia en qualité de metadonnées*, qui s'appliques aux applications dites augmentées multimédia, les données multimédia jouent là un rôle d'une simple source supplémentaire d'information sur l'univers du discours, quel qu'il soit, de l'application. Un exemple d'une application augmentée multimédia est une base de données de ressources humaines enrichie par les photos des employés. Ici l'univers du discours est les employés d'une entreprise, tandis que leurs photos représentent simplement une source d'information supplémentaire (et peut-être optionnelle) sur cet univers du discours de l'application.

L'approche de la modélisation conceptuelle du multimédia que nous présentons ici s'applique tant aux applications orientées multimédia qu'aux applications augmentées multimédia. Le modèle que nous proposons s'appuie sur MADS (Modeling Application Data with Spatio-temporal features), qui est un riche modèle conceptuel défini au laboratoire et qui est en particulier caractérisé par une plénitude structurelle, des capacités de modélisation spatio-temporelle, ainsi que le support de

la multi-représentation. Notre modèle multimédia se présente sous la forme d'une nouvelle dimension du modèle MADS, dont le principe d'orthogonalité permet d'intégrer cette nouvelle dimension de modélisation avec les autres moyens de modélisation déjà existants dans MADS. Le modèle multimédia se compose des éléments suivants: les types de données multimédias, les contraintes (relations) de représentations basiques et complexes, un mécanisme de partitionnement des données multimédias, ainsi que les moyens de multi-représentation multimédia.

A la suite de la description de notre approche de modélisation conceptuelle des données multimédias basée sur MADS, nous présentons les particularités de la modélisation logique du multimédia ainsi que de la transformation d'un modèle conceptuel dans un modèle logique. Nous fournissons un ensemble de directives de transformation destinées à aider le concepteur du schéma à élaborer une riche représentation multimédia logique du domaine de l'application par des documents multimédias qui soient conformes au schéma multimédia conceptuel.

L'intérêt pratique de notre travail est illustré par une maquette d'application, qui a été développée dans le but d'appuyer les idées théoriques décrites dans ce travail de recherche. Nous démontrons en particulier comment les représentations conceptuelles abstraites des éléments multimédias basées sur la théorie des ensembles, ainsi que les relations de représentation basiques et complexes peuvent être implémentées dans une base de données Oracle.

**Mots clefs:** multimédia, modélisation conceptuelle des bases de données, bases de données multimédias, MADS.

# Acknowledgements

Obtaining a Ph.D. degree is a challenge, which demands significant personal efforts and implication. However, I'm certain that the thesis you are reading would not exist without the assistance of many people (colleagues, friends, family members), to whom I am sincerely grateful.

First of all I would like to thank my thesis director, Prof. Stefano Spaccapietra, for the wisdom he has shared and for the trust he had in me during all these years. He always knew exactly which questions to ask to give me the answers I was looking for.

I would also like to thank all the members of the database laboratory, and in particular Christine, Anastasiya, Christelle, Fabio, Abdel, Lina, Fabrice, Shijun, José, Marlyse, Chiara, for the helpful advices and comments they gave me, and simply for being wonderful colleagues and friends.

Prof. Anastasia Ailamaki has kindly accepted to preside the oral exam jury, which I greatly appreciate. My thanks also go to the jury members, Prof. Olga de Troyer, Prof. Wolfgang Klas, and Prof. Daniel Thalmann, for the time they spent and for the valuable remarks that have helped me improve my thesis.

Last but not least, my family and friends have also contributed a lot to this accomplishment. I am endlessly obliged to my lovely wife Ulyana and to my parents for always encouraging me in my endeavors, for cheering me up, and simply for always being there for me.

My relatives, my friends, my colleagues from Gland are as well among those who have been urging me to continue, for which I am very thankful.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

In the era of advanced computer and communication technologies multimedia data becomes more and more pervasive. With multimedia-recording equipment becoming increasingly affordable and compact, the number of digital multimedia sources (incl. photos, videos, and audios) constantly increases. This notable growth has in particular been intensified by popularization of camera-enabled cell phones and other types of mobile devices combined with a quasi-general connectivity via cellular and WiFi networks. The emergence of social networking phenomena on the Web has also contributed to the ubiquitous presence of multimedia due to on-line services for photo sharing, pod casting, mobile video blogging, etc.

Nevertheless, despite the simplicity of taking photos with one's digital camera and storing them in an Web repository, performing an automated semantic search in a photo collection still often represents a challenging task. One of the reasons of this problem, which is generally characterized as the semantic gap problem, is the fact that the major emphasis in the field of multimedia research has been for a long time placed on low-level multimedia characteristics like storage, encoding, etc., while paying little attention at semantic aspects of multimedia information.

In the context described above, we believe that the issue of conceptual modeling of multimedia-enhanced databases represents an important and interesting research direction, which could help bridge the semantic gap between the low-level content and the high-level semantic meaning of multimedia data.

## 1.2 Multimedia Semantics Research Background

Research in the field of multimedia information systems began in the early 80s and mostly focused on digital image retrieval systems. Although emerging as a new

scientific community at the time, the area of multimedia image retrieval was based on strong theoretical foundations of other already existing research areas like artificial intelligence, pattern recognition, optimization theory, etc. At the same time sciences and studies like psychology, ergonomics, etc. have provided inspiration to researchers in the field of human-computer interaction. Moreover, some limited digital-image-based search systems such as OCR (optical character recognition) applications or robotic-related applications have also already existed by that time.

### 1.2.1 Low-Level Content-Based Retrieval

The early efforts in multimedia information retrieval were mostly dealing with computer vision algorithms, which focused on feature-based similarity search within the multimedia recordings. Among the most representative examples of that kind of systems are QBIC [AFH+95] and Virage [BFG+96]. Basically, the user has to present a sample image to the system in order to find images in the collection that are visually similar to the presented one. The presented image is processed and low-level features like color histograms or image patterns are extracted. By comparing the extracted low-level features of the sample image with the corresponding features of the images stored in the collection, the system is able to return the images that visually correspond to the presented sample with some degree of certainty. A scenario described above is also known as query by example (QBE).

By the late 90s the general concept of similarity search has found its implementation in the Internet, and several image search engines like WebSeer [FSA96] and WebSeek [SC97] have appeared. In WebSeek autonomous Web agents traverse the Web by following hyperlinks between documents and detect images and videos, which are retrieved, processed, and described in the catalog. The system provides two methods for content-based searching: by color histograms and by spatial locations and arrangements of color regions. In the area of video data retrieval, the focus had been initially on automatic real-time temporal segmentation algorithms for shots- and scene-detection. The most common approach consisted in measuring discrepancies in color histograms of adjacent video frames. Other algorithms of video shot detection included the ones based on motion detection within the video sequence [Lie01]. At about the same time the feature-based similarity search has also found its adoption in commercial enterprise-level DBMS like Oracle and DB2, which have elaborated extensions to implement the similarity search of the multimedia data stored within the database.

Nevertheless, the feature-based similarity search algorithms were essentially user-unfriendly, and due to their underlying complexity they were mostly meant for multimedia retrieval professionals and researches.

### 1.2.2 From Low-Level Features to Semantics

Starting from the end of the 90s the new trend in multimedia research was to devise multimedia systems that would be user-friendly and that would allow public access to various multimedia archives and collections. Contrary to feature-based search systems, where information was retrieved using highly unintuitive and cryptic low-level characteristics of the multimedia data, the new multimedia systems were aiming at allowing the users to formulate their queries using human-understandable categories and concepts relating to the semantic content of the multimedia data. The problem of bringing together the semantic meaning of multimedia with its underlying low-level form has been commonly named as "bridging the semantic gap".

The early efforts in bridging the semantic gap simply consisted in correlating the automatically computed low-level media features (e.g. similar-pattern or similar-color regions of an image) with high-level semantic concepts that the end-user is interested in. An examples of a system trying to tackle the semantic gap problem in the area of human face detection is described in [RBK96]. One of the early image content-based retrieval (CBR) systems addressing the semantic gap problem in query interfaces is the ImageScape search engine [LSDJ06], where users can make queries for visual objects such as trees, sky, water, etc. using spatially positioned icons in an index containing over $10^7$ images. The system uses the notion of entropy from the information theory to determine the best features in order to minimize uncertainty in the classification of the returned images.

It has been generally recognized that it was not feasible to bridge the semantic gap by uniquely low-level approaches. According to [LSDJ06], the research topics that have the potential for improving multimedia retrieval by bridging the semantic gap can generally be characterized as human-centric computing. The main idea behind the human-centric computing is to allow the user make the queries in his/her own terminology. Due to the complexity of the problem of automatic multimedia content extraction, an alternative class of annotation-based approaches, which make multimedia image retrieval applications more user-centric, have emerged. Keyword-based approaches have been adapted by all the existing search engines, which rely on a substantial manual component. In order to increase their efficiency, the keyword-based systems would often address only some limited specific area and try to use taxonomy- or ontology-based approaches in order to organize their keyword corpora.

## 1.3 Thesis Objectives

Taking into account the multimedia semantics research background presented above, we believe that one of the reasons, which hampers the solution of the semantic gap problem, is the fact that even the semantics-oriented multimedia systems based on annotations and tagging, still attempt to conceptually represent multimedia data

in a way, which is similar to its low-level representation. Thus, such systems would traditionally try to provide annotations per-photo or per-spatial-portion of a photo. We call this approach multimedia-centric. While the multimedia-centric approach may be perfectly suitable for a wide range of applications, for which the photos in a collection represent the very subject of modeling, there exists a class of so-called multimedia-enhanced applications, where multimedia data would simply represent an additional source of information about the universe of discourse, whichever it is, that the application pertains to. In a context like this, the traditional multimedia-centric representation of multimedia semantics would contribute to the semantic gap by separating the representation of multimedia information from the representation of the rest of the universe of discourse of an application, to which the multimedia information pertains.

Taking into account what precedes, the goal of our research work is to address the problem of conceptual modeling of multimedia data in a way, which would be suitable for multimedia-centric, as well as, in particular, multimedia-enhanced applications. The conceptual model to propose should, in particular, allow to represent multimedia data of various sensorial types and to express relationships between multimedia data elements. Since, as it has been mentioned above, in the case of multimedia-enhanced applications their universe of discourse is in general independent of the existence of any multimedia data, this implies that the conceptual modeling technique, which we want to devise, should allow to represent any aspect relative to the universe of discourse of the application, no matter multimedia or not.

## 1.4 Thesis Roadmap

The rest of the dissertation is organized as follows:

**Chapter 2** (Metadata and Annotations in Multimedia Information) introduces what the multimedia data is and the ways we perceive it. First, we provide a state-of-the-art overview of research on semantic descriptions of multimedia. We then introduce the duality principle of multimedia data and accentuate the importance of an alternative vision of multimedia data as the source of semantic descriptions in its own right.

**Chapter 3** (Conceptual Multimedia Modeling: MADS Multimedia Extension) introduces a novel multimedia-enhanced conceptual modeling approach, which, according to the multimedia duality principle, allows to model multimedia information perceived as either data or metadata. Our modeling approach is presented in the form of an extension to a conceptual model MADS, which is also introduced in the chapter.

**Chapter 4** (Logical Multimedia Modeling) discusses the peculiarities of logical multimedia modeling. We start by presenting an overview of several existing logical multimedia document models. We then describe the peculiarities of conceptual-to-logical mappings, and address this issue by providing a set of mapping guidelines.

**Chapter 5** (Experimental Implementation Results) demonstrates the feasibility of the research ideas described in this thesis by the example of a mock-up application implementing a number of multimedia modeling concepts.

**Chapter 6** (Conclusion and Future Directions) concludes the thesis and outlines the future research possibilities.

# Chapter 2

# Metadata and Annotations in Multimedia Information

In chapter 1 we have introduced the idea of devising a rich conceptual multimedia-enhanced database modeling technique. In order to efficiently tackle this problem it is first of all important to comprehend what the multimedia information is and what are its peculiarities.

In our opinion, multimedia information is manifested by its two general aspects: 1) sensorial information, i.e. a purely visual or aural part of the multimedia data; and 2) additional information that the multimedia data conveys. While the first aspect barely deals with media-oriented concepts like image bitmaps or sound waves, the second one is about the knowledge that the multimedia data provides, like the fact that Switzerland has picturesque mountain views, as can be seen on a picture, or that a song we hear is in Italian. The latter kind of information can be classified as metadata on multimedia data, or, more specifically, annotations of multimedia data.

Metadata can be naturally defined as *data about data* [Mar82]. Although the idea of metadata is widely used and is not peculiar to the domain of multimedia information, in our opinion it is with multimedia that metadata becomes particularly important. We believe that the main reasons to this are:

1. The computational intricacy of such highly unstructured data as digital images, videos, music, etc., which often prohibits us from easily recalculating on the fly even such basic information as the number of paragraphs in a RTF[1] document each time it is read;

2. A high level of subjectivity proper to interpretation of multimedia data, which leads to existence of a multitude of possibly very different (or even strictly

[1]RTF (Rich Text Format) is a free document file format developed by Microsoft for cross-platform document interchange. Most word processors are able to read and write RTF documents.

opposite) user-dependent descriptions within the metadata.

The two factors mentioned above clarify the importance of providing a way of associating metadata descriptions alongside the underlying multimedia information.

It is important to note that metadata on multimedia data can be divided into two main categories: 1) information that is not related to the semantic content of multimedia data, e.g. encoding-related or authoring information; and 2) user-provided descriptions of the semantic content of multimedia data.

The former class of metadata deals with basic tags like the MIME type of a particular piece of multimedia data (e.g. the MIME type of a GIF image file is `image/gif`), the run-length of a video or audio stream, the author of the multimedia document, etc. Metadata of this kind is often referred to as simply *metadata*, and it can be generally characterized by its context-free veracity and relative constancy (e.g. the author of a video clip is independent of the personal preferences of the video clip viewer and it is a time-constant value). The metadata of this type generally tries to follow the generally adopted description formats like Dublin Core [PNN+07] [NPJN08], Exif [JEI02], ID3 [ID3], etc.

The second class of metadata is generally referred to as *annotations*. Unlike the first class of metadata, annotations are generally highly context- and user-dependent, and tend to change and evolve. Users can tag multimedia data with different kinds of annotations, like the places, objects, and people that can be perceived in the multimedia data and the relationships between them (e.g. a photo annotated by "A weekend in Prague with friends"), or could also relate to some ratings or measurements, like "interesting" or "****", etc., or else provide some sort of categorization, e.g. "travel photos", "movies->comedy", and so on.

The annotation systems nowadays can be roughly classified as either: 1) organized (controlled) annotations, or 2) free-form annotations.

## 2.1 Controlled and Ontology-Driven Annotations

Organized annotation systems impose some control on what annotations can be entered and how. This is generally achieved by imposing some sort of a controlled vocabulary, taxonomy, or ontology to be used by the annotators. In particular, the ontology-based systems are being backed up in the last few years by an overgrowing Semantic Web community. We provide hereby an overview of the existing ontology-driven annotation systems.

In the last few years a great attention of the computer science research community has been drawn to ontologies and their applications in information and

data semantics. In philosophy, ontology is the study of being or existence. It seeks to describe or posit the basic categories and relationships of being or existence to define entities and types of entities within its framework. Ontology can thus be said to study conceptions of reality. One of the most important applications of ontologies lies in the area of Semantic Web [W3Cc], [BLHL01].

Semantic Web (also referred to as Web v.3.0) is an evolving extension of the World Wide Web in which web content can be expressed not only in natural language, but also in a form that can be understood, interpreted and used by software agents, thus permitting them to find, share and integrate information more easily [Car07]. It derives from W3C director Tim Berners-Lee's vision of the Web as a universal medium for data, information, and knowledge exchange. At its core, the Semantic Web comprises a philosophy [CS06], a set of design principles [DOS03], collaborative working groups, and a variety of enabling technologies. Some elements of the Semantic Web are expressed as prospective future possibilities that have yet to be implemented or realized. Other elements of the Semantic Web are expressed in formal specifications, which include, in particular, Resource Description Framework (RDF), a variety of data interchange formats (e.g. RDF/XML), and notations such as RDF Schema (RDFS) and the Web Ontology Language (OWL). All of these are intended to formally describe concepts, terms, and relationships within a given knowledge domain.

The main driving idea behind the Semantic Web is a wish to make information on the Internet easily "understandable" by computers without the need for any human intervention. This would in its turn allow for a fully automatic inter-computer communication and data sharing (data discovery) between computers on the Internet.

One of the emerging domains of ontological and Semantic Web research is semantic multimedia annotations. Combining existing multimedia annotation assets with Semantic Web could significantly enrich the latter by converting existing annotations and vocabularies to ontological annotation formats like RDF or OWL, while the multimedia applications could benefit from extra functionality offered by Semantic Web languages and tools. The expected applications that could emerge from such collaborations include on-line and on-demand media contents delivery (e.g. TV on demand), automatic content-aware multimedia discovery and trading, etc.

Among the steering institutions of ontological multimedia research are aceMedia (acemedia.org) and the SemanticWeb community (multimedia.semanticweb.org), which regroup a dozen of academia and industry partners.

In the next subsection we provide a brief state-of-the-art overview of ontology-driven multimedia systems.

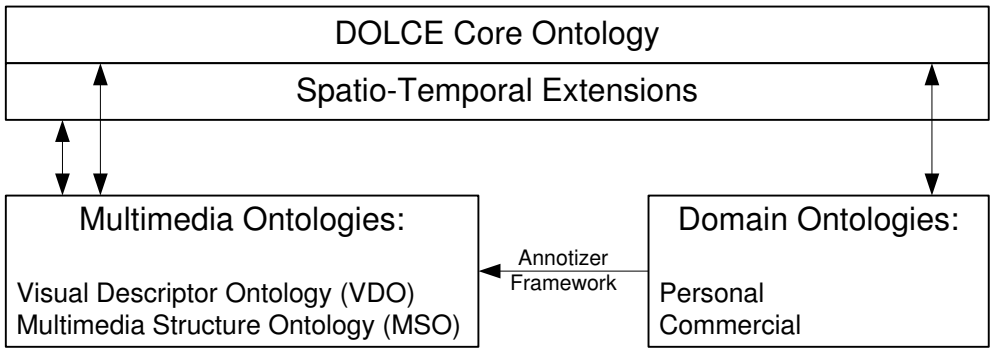



**Figure 2.1:** AceMedia ontology structure overview.

### 2.1.1 Multimedia Ontologies, State of the Art

#### aceMedia Multimedia Ontology Infrastructure

Developed at the National Technical University of Athens, aceMedia Multimedia Ontology Infrastructure [ATP+05] consists of: (i) a domain specific ontology that provides the necessary conceptualizations of a specific domain like e.g. tennis sports events, (ii) multimedia ontologies that model the multimedia layer data in terms of low level features and media structure descriptors, and (iii) a core ontology that bridges the previous ontologies in a single architecture. Additionally, a semantic annotation tool, M-OntoMat Annotizer, is provided, which is capable of eliciting and representing knowledge both about content domain and the visual characteristics of multimedia data itself. The aceMedia Infrastructure is RDFS-based and it can be characterized as modular, since it couples domain-specific and low-level description vocabularies.

The aceMedia ontology structure is shown in fig. 2.1. The role of the Core Ontology in the framework in fig. 2.1 is to serve as a starting point for the construction of new ontologies, to provide a reference point for comparisons among different ontological approaches and to serve as a bridge between existing ontologies. For this purpose aceMedia uses DOLCE (a Descriptive Ontology for Linguistic and Cognitive Engineering) [GGM+02]. DOLCE was explicitly chosen as a minimal and rigorously documented ontology, which includes only the most reusable and widely applicable upper-level categories.

The Multimedia Ontologies in fig. 2.1 model the domain of multimedia data, especially the visualizations in still images and videos in terms of low-level features and media structure descriptions. Structure and semantics are modeled to be consistent with existing multimedia description standards like MPEG-7. Based on MPEG-7's Visual Part and Multimedia Description Scheme, the following two ontologies make up Multimedia Ontologies in aceMedia:

- The Visual Descriptor Ontology (VDO) contains the representations of the MPEG-7 visual descriptors and models Concepts and Properties that describe visual characteristics of objects. Descriptors are specific representations of a visual feature (color, shape, texture, etc.) that define the syntax and the semantics of a specific aspect of the feature (dominant color, region shape, etc). The structure of the VDO is tightly coupled with the specification of the MPEG-7 Visual Part, with several modifications made to adapt to the XML Schema provided by MPEG-7 to ontology and the data type representations available in RDF Schema.

- The Multimedia Structure Ontology (MSO) models basic multimedia entities from the MPEG-7 Multimedia Description Scheme and mutual relations like decomposition. MPEG-7 provides a number of tools for describing the structure of multimedia content in time and space. The Segment DS describes a spatial and/or temporal fragment of multimedia content and a number of specialized subclasses are derived from that. These subclasses, which describe specific types of multimedia segments (such as video segments, moving regions, still regions and mosaics), along with their relations, are modeled inside the MSO.

Finally, Domain Ontologies, in fig. 2.1 are meant to model the content layer of multimedia information with respect to specific real-world domains, such as sports events, etc. All domain ontologies are explicitly based on or aligned to the DOLCE core ontology, and thus connected by high-level concepts. The core ontology in its turn assures interoperability between different domain ontologies at a later stage. Domain Ontologies are defined in a way to provide a general model of the domain, with focus on the users' specific point of view. In general, the domain ontology needs to model the domain in such a way that the concepts should be recognizable by automatic analysis methods, however remain comprehensible by a human.

### BOEMIE

The BOEMIE project [KPP+06] proposes a bootstrapping approach to knowledge acquisition, which uses multimedia ontologies for fused extraction of semantics from multiple modalities, and feeds back the extracted information, aiming to automate the ontology evolution process.

BOEMIE advocates a synergistic approach that combines multimedia extraction and ontology evolution in a bootstrapping process involving, on one hand, the continuous extraction of semantic information from multimedia content in order to populate and enrich the ontologies and, on the other hand, the deployment of these ontologies to enhance the robustness of the extraction system.

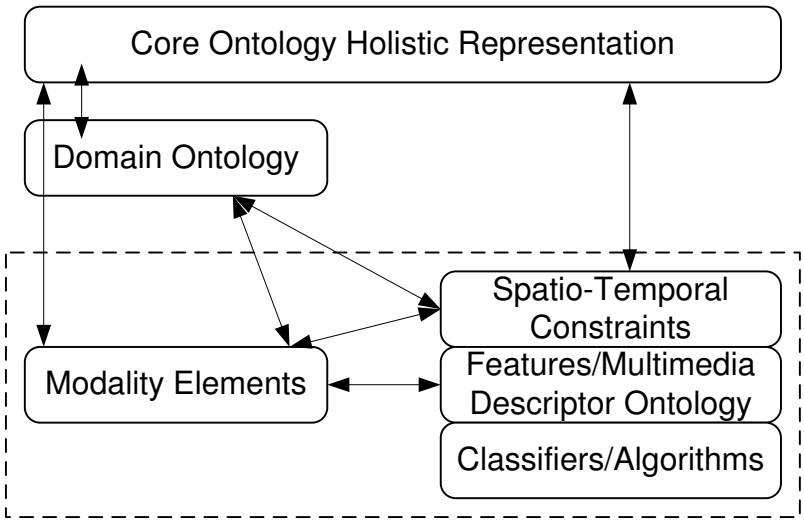



**Figure 2.2:** BOEMIE knowledge framework.

The knowledge framework of BOEMIE is represented in fig. 2.2. As one might notice, the framework in fig. 2.2 shares a number of similar concepts with aceMedia ontology structure shown in fig. 2.1. Indeed, the BOEMIE Knowledge Framework relies on a concept of core ontology. A core ontology is a very basic and minimal ontology consisting only of the minimal concepts required to understand the other concepts. Moreover, the Modality Elements in fig. 2.2 also heavily depend on MPEG-7 descriptors, since MPEG-7 is used to model low-level and structural aspects of multimedia documents, while domain-specific ontologies model high-level semantics.

### Multimedia Ontology System Analysis

In the subsections above we have presented an overview of two sample multimedia ontology systems. We proceed our survey by briefly presenting several other systems and then concluding on the general characteristics of existing multimedia ontology approaches.

Tsinaraki et al. [TPC04] describe a framework for extending MPEG-7 and TV-Anytime [PS00] with domain-specific ontologies. They express the semantic part of MPEG-7 Multimedia Description Schemes (MDS) in OWL and domain-specific ontologies extend this core ontology to fully describe the concepts of application domains. Taking the example of soccer domain, the `FootballTeam` concept from the domain-specific ontology extends the `OrganizationType` of the core ontology that was designed based on MPEG-7.

In [GC05] an attempt to completely move MPEG-7 to the Semantic Web world

is described by Garcia and Celma. The authors automate the entire conversion of MPEG-7 standard to OWL as envisaged by Hunter [Hun01] using XML Schema to OWL mappings. However, the authors try to map the domain-specific vocabularies to the ones provided by MPEG-7. For example, the concept `Artist` from a music ontology is mapped to the MPEG-7's concept `CreatorType` by sub-classing. The scalability of such approaches is questionable, since it tightly couples the domain-specific concepts with the semantic part of MPEG-7 by specialisation. Concept mapping based on semantic relations of equivalence or inclusion in the ontologies may not be feasible for domains with rich semantics.

Troncy [Tro03] asserts the need to infer multimedia documents at both their structural as well as conceptual aspects. MPEG-7/XML Schema is used to express the structural meaning of multimedia documents, and OWL is used to model their semantic aspects using a domain-specific vocabulary. A transformation mechanism from OWL to XML Schema results in XML Schema descriptors of the domain-specific constructs that could be linked with existing MPEG-7 types by specialisation. Isaac and Troncy [IT05] describe a case study in the medical domain and show that the combination of several ontologies results in better description and retrieval of audio-visual sequences. Bloehdorn et al. [BPS+05] describe an ontological framework and a software environment that allows linking low-level MPEG-7 visual descriptors to concepts in domain-specific ontologies based on a prototype approach.

#### Multimedia Ontologies: Summary

Summarizing the several multimedia ontology systems presented above, we can conclude on a number of common characteristics inherent to all of the above-mentioned approaches. These characteristics are:

1. The existing approaches do not try to devise a single pan-domain ontology that would cover all the aspects of the universe of discourse. Instead, they rather focus on integrating together several ontologies, corresponding each to some conceptual sub-domain (e.g. multimedia descriptor ontologies, spatio-temporal organization ontologies, application domain ontologies, etc.).

2. The existing approaches rely on a minimalistic high-level core ontology, which on one hand acts as a connecting link between different ontologies composing the overall framework, and on the other hand allows providing mappings to other systems' ontology frameworks. It should be noted that this latter property becomes particularly important in the Semantic Web environment, since it allows for system interoperability and automatic discovery of multimedia information on the Internet.

3. MPEG-7 is the de facto standard for multimedia structure descriptions in the presented systems. While domain-specific ontologies model high-level semantics, MPEG-7 is used to model low-level and structural aspects of multimedia

documents. Popularity of MPEG-7 in ontology-enhanced systems is further demonstrated by creation of the W3C Multimedia Semantics Incubator Group on the Apr. 25, 2006 [W3Ca].

### 2.1.2 Section Summary

Having established the general characteristics of the ontology-based multimedia systems, let's try to summarize the pros and cons of controlled multimedia annotation approaches on the example of the systems described above.

The advantages of ontology-based multimedia annotation approaches are mostly similar to the general advantages of using ontologies in any other domain. Namely, ontology-based annotations of multimedia information become machine-readable and machine-understandable. This in its turn implies better semantic interoperability between various systems in a Semantic Web environment.

However, despite the obvious advantages, the ontology-based approaches also have a number of disadvantages.

First of all, due to their unwieldy design, which is inherent to ontologies in general, and, even more, their increased complexity resulting from the use of several different ontologies combined in a single framework (e.g. domain ontology and multimedia descriptor ontologies), ontological descriptions of multimedia data easily become way too intricate to be understood by a human. This in fact undermines the very foundations of ontologies and Semantic Web, where, by definition, the content should be understandable by both humans and intelligent agents (computers). Putting it simply: too much ontology kills ontology.

Another disadvantage of the examined ontology-based multimedia systems is their raw-media-orientation. As will be demonstrated in sect. 2.4, the presented multimedia systems can be classified as adhering to the *multimedia as data* viewpoint, and hence providing a sort of ontology-enhanced semantic extraction systems for multimedia data. Although such an approach is perfectly appropriate in many cases, there exists a large class of applications (*multimedia as metadata* viewpoint), where such a methodology would be inadequate. Instead, we would like to be able to provide information about some real world object or phenomenon (information possibly organized through an ontlolgy) without necessarily having this object or phenomenon depicted on a multimedia support. Moreover, in above-described systems like aceMedia, changing some underlying multimedia sources by semantically equal (or similar) ones would probably require to revise the entire ontology-based annotations, which is a costly process possibly requiring substantial human intervention and entailing referential integrity problems.

## 2.2 Free-Form Annotations

In sect. 2.1 we have described a number of ontology-based multimedia annotation approaches. Despite the obvious benefits of controlled annotation approaches, such as enhanced semantics, stronger interoperability, etc., their advantages are sometimes questioned (see e.g. [Shi06]).

Indeed, the systems imposing a particular taxonomy or ontology to be used by the annotators do not work well in heterogeneous multi-user environments like Internet, where obliging the users to stick to some centralized dictionary is hardly possible.

With overgrowing collections of multimedia data becoming more and more available on the Internet, the tendency that we witness nowadays with Web 2.0 is that multimedia-oriented systems like Flickr[2], Facebook[3], del.icio.us[4], etc. prefer free-form tags and folksonomies to controlled taxonomies or ontologies. Although this fact is often believed (especially by the followers of the Semantic Web paradigm) to be due to the lack of sufficient technical advancements in the field of semantic interoperability, which manifests itself through the adoption of less science- and implementation-intensive approaches like free-text tagging, we believe that such an argumentation is only partly true.

As argued in [Shi06], ontology- or taxonomy-based approaches don't work well in environments with a domain characterized by a large corpus, having no formal categories, having unrestricted or unstable entities, without clear edges, and where the users of the system are uncoordinated, are not governed by an authority, and are not professional annotators. The major example of such kind of environment is Web 2.0. For example, with applications like Flickr, one can take part in annotating the community uploaded photos by simply entering a set of keyword tags, or even complete sentences.

Obviously, the ease of adding free-form uncontrolled annotations in environments like Web 2.0 comes at a price of obtaining annotations from participants who are generally neither domain specialists, nor professional annotators obliged by their job duties to take all the necessary time and efforts to produce the best annotations possible. This, in particular, often results in annotations that are incomplete and semantically imprecise. Moreover, as already mentioned above, annotations are to a large extent user-specific and depend on a particular participant's perception of things and user's personal viewpoints. For this reason, the same object can receive

[2]Flickr.com is an image and video hosting website, web services suite, and online community platform. It was one of the earliest Web 2.0 applications.

[3]Facebook.com is a social networking website.

[4]del.icio.us is a social bookmarking web service for storing, sharing, and discovering web bookmarks.

absolutely valid however possibly very different annotations simply because they are being produced by different annotators. Another problem peculiar to free-form annotations is that although very easy to enter, free-form annotations are difficult to search against, since unlike the controlled annotations (e.g. ontology-based), the natural-language-like form of uncontrolled annotations makes them difficult to be processed by machines.

Although lacking a rigid organizational structure, free-form annotations still represent the favorite annotation mechanism on the Internet. The strong community orientation of the uncontrolled annotation paradigm is also what has to a large extent helped resolving the problems peculiar to free-form annotations (see above).

Thus, a low suitability of free-form annotations for automatic intelligent search techniques that rely on controlled vocabularies has been overcome by making the search more user-intensive. As a consequence, instead of letting the system "improve" the query string based on relationships between the terms as defined in a taxonomy or ontology, it becomes up to the user to enter a set of keywords that describes close enough the information the user is looking for. One of the reasons this approach works is because in general the kind of keywords users enter as a search string are the same kind of keywords the same user community employs to produce the free-form annotations.

Also the problem of a high level of subjectivity of uncontrolled annotations resulting from a vast and heterogeneous user (annotator) base has found a solution in Web 2.0. While the latter problem does hamper the global mutual understanding of user-provided annotations, which could be achieved if controlled annotations techniques were used, it has been argued that in the context of Web 2.0 this problematic has brought a beneficial side effect, which is in particular used in the blogosphere and social networking services on Web 2.0, like Facebook, LiveJournal[5], etc. As a matter of fact, the lack of ubiquitous understanding of free-form annotations leads to a kind of partitioning of the user base, since the users that are likely to belong to the same community according to their interests, cultural background, age, etc. tend to show the same annotation habits and trends as their community peers, thus making the user-provided free-form annotations easier to use within the same user community. This in its turn helps adding a certain amount of personalization in the universally accessible Internet-based environments, on which many Web 2.0 applications, e.g. social networks, rely.

This above-described user community partitioning phenomenon, as well as the associated information browsing and searching technique based on the likeliness of annotation habits of users with similar interests is sometimes called *pivot browsing*, which is a technique that is inherent to *folksonomies*.

[5]Livejournal.com is a popular blogging service with some social networking features.

### 2.2.1 Folksonomies

Dealing with problems inherent to user-generated annotations (e.g. see several problems discussed above) is the area of *folksonomies*. Folksonomy (from "folk" and "taxonomy") is the practice and method of collaboratively creating and managing tags to annotate and categorize content [Vos07]. The term folksonomy is also known as collaborative tagging or social tagging, since it is in the context of social networking applications like digg.com[6] or Flickr in a Web 2.0 environment that folksonomies have gained large attention.

Folksonomies usually deal with free-form user-generated tags instead of controlled vocabularies. In a certain way folksonomies can be seen as an attempt to bring some control into the world of free-form annotations without however compromising the fundamental ideas of free-form tagging, i.e. no imposed dictionary or taxonomy, no mandatory standards to follow, potentially unlimited annotator base, etc.

While preserving the above-mentioned advantages of free-form annotation approaches, folksonomies try to bring answers to the problems that are inherent to free-form tagging. Thus, in order to facilitate browsing vast and uncontrolled tag collections, taxonomies provide approaches like pivot browsing (see above) or tag clouds, which help ease the navigation within the free-form tag collections.

Tag clouds, like the one shown in fig. 2.3, represent tags (usually single keywords) with their importance (weight) shown with font size or color. The tags are usually hyperlinks that lead to a collection of items that are associated with a tag. Tags in a tag cloud are often clustered semantically so that related tags appear closer to each other.

## 2.3 Annotea Annotation Standard

At the beginning of this chapter we have discussed the role that metadata and annotations play for multimedia data sources. Two major types of annotation approaches (i.e. controlled and free-form) have been introduced, and the advantages and disadvantages of each of the approach have been discussed. Throughout this introduction we have presented examples of different systems and paradigms pertaining to either of the annotation approaches.

In this section we would like to introduce an extensible annotation standard from W3C called Annotea [KKS02], as well as to introduce some already existing extensions of this system, which allow, in particular, embracing not only free-form

[6] digg.com is a social networking Web site with a story submission and voting system.

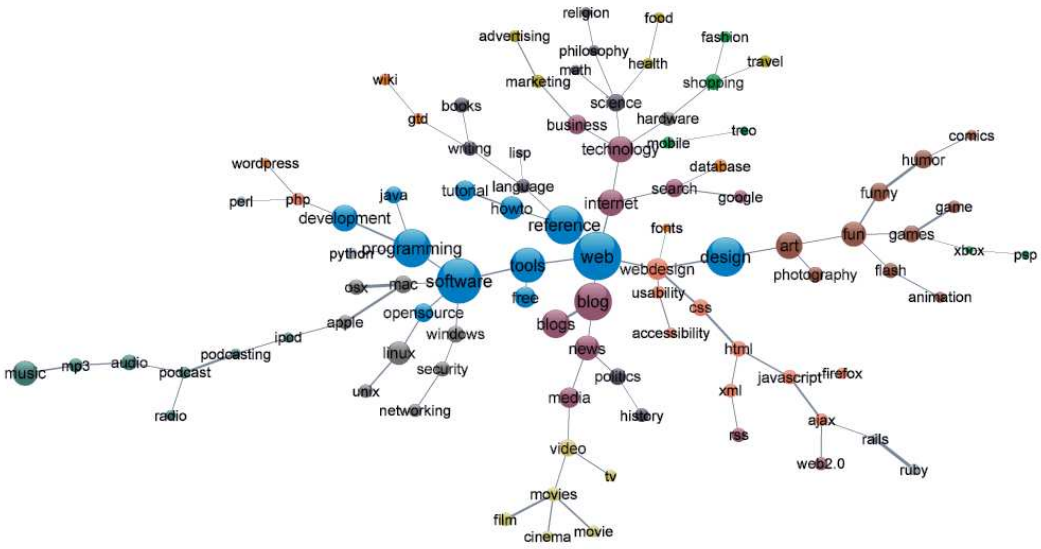



**Figure 2.3:** Example of a tag cloud.

annotations, but also the controlled ones.

In its original form Annotea [KKS02] is a Web-based annotation system. Annotea is open and it uses W3C standards when possible. For instance, RDF (Resource Description Framework) is used to describe annotations as metadata, and XPointer is used for locating the annotations in the annotated document. Using RDF, annotations are modeled as a set of "who - said what - about what?" triplets. The annotations are then stored on a Web-based server, which enables collaborative querying and editing of annotations, bookmarks, and their combinations. Annotations in Annotea could mean comments, notes, explanations, or other types of external remarks. They can be attached to a Web document or a selected part of the document without actually needing to modify the document itself. When the user gets the document, he or she can also load the annotations attached to it from a selected annotation server or several servers.

The RDF Schema model of the annotations is freely accessible on the Web[7]. The general annotation super-class in Annotea is called `Annotation`, and its precise URI name is `http://www.w3.org/2000/10/annotation-ns#Annotation`. Fig. 2.4 provides a graphical illustration of Annotea RDFS.

Annotea is an extensible format and developers are encouraged to create new

[7]http://www.w3.org/2000/10/annotation-ns

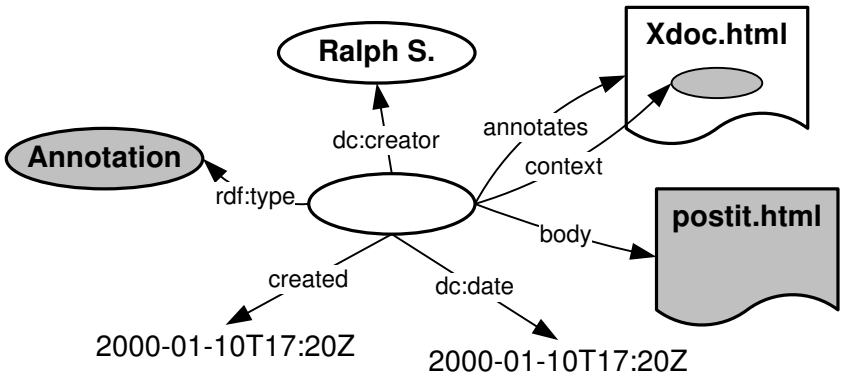



**Figure 2.4:** The RDF model of an Annotation.

types of annotations by sub-classing from the `Annotation` class. Due to extensive use of open XML-based standards like RDF/S, developers can easily integrate description formats like Dublin Core or FOAF into their Annotea metadata.

In fig. 2.5 we demonstrate an extension of the Annotea model proposed by Schroeter and Hunter [SHN07].

The RDFS schema of Annotea in the upper part of fig. 2.5 has been extended with a new class `Comment`, which is a sub-class of `Annotation`, as well as a new property `body`. The lower part of fig. 2.5 demonstrates an instantiation of the extended Annotea RDFS by basically describing an annotation of a specific part (`context`) of a structured Web document (`annotates`), which is specified using an XPointer expression. The `body` of the `Comment` specifies the free-text annotation itself. Finally, the annotator (author of the annotation) is represented by a `foaf:maker` property of the `Comment` object. In that way, this sample Annotea `Comment` object carries information of the kind "who - said what - about what?"

Another important Annotea extension described in [SHN07] provides a way of entering controlled annotations based on controlled vocabularies or ontologies. In this way, annotations can come in the form of controlled terms (e.g. terms from an ontology), or even entire statements composed of controlled terms. The fig. 2.6 illustrates such kind of annotation.

As can be seen on the upper part of fig. 2.6, authors have extended the Annotea RDFS with an annotation subclass called `FormalStatement`. Also the property `related` has been specialized into a new sub-property `states` of the subclass `FormalStatement`, which has a range of `rdf:Statement`. The bottom part of the fig. 2.6 presents an instance of the extended Annotea RDFS, which models a statement-type annotation "lion eats gazelle". All terms of the statement (i.e. subject, predicate, and object) come from an OWL wildlife ontology.

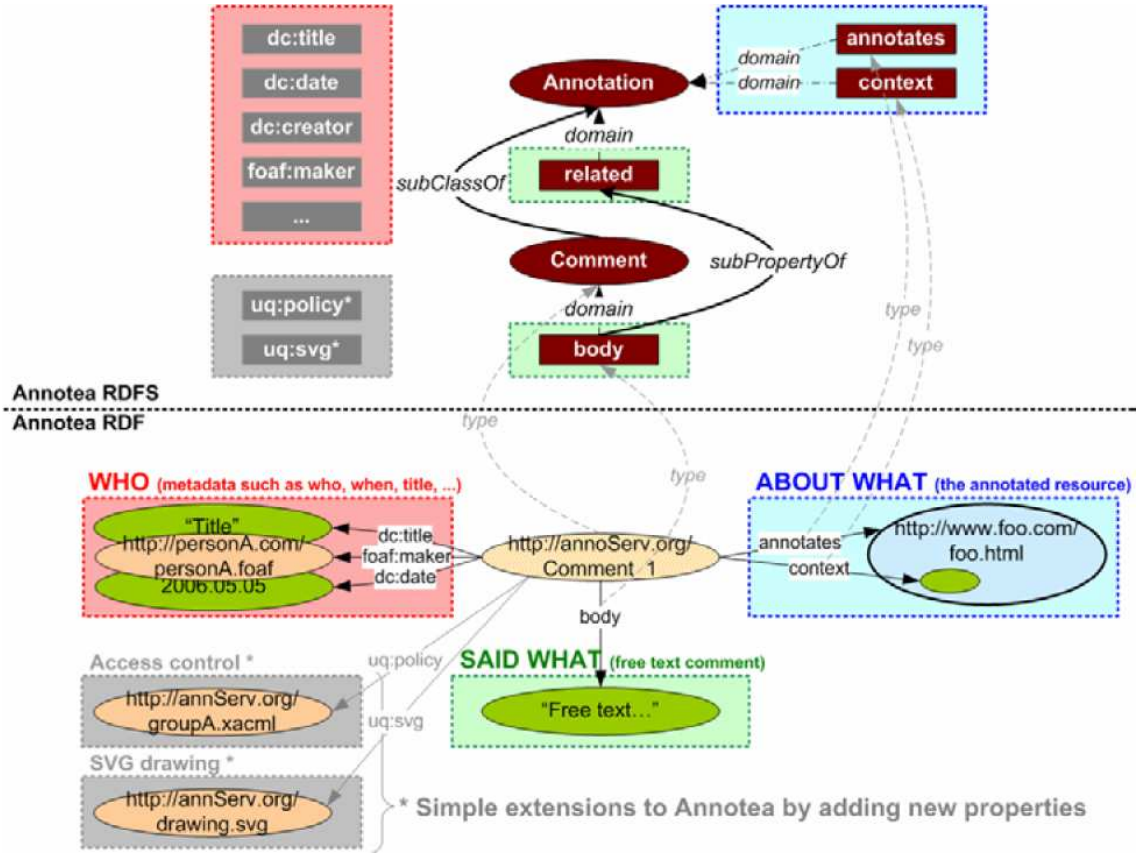



**Figure 2.5:** An extended Annotea schema and instance.

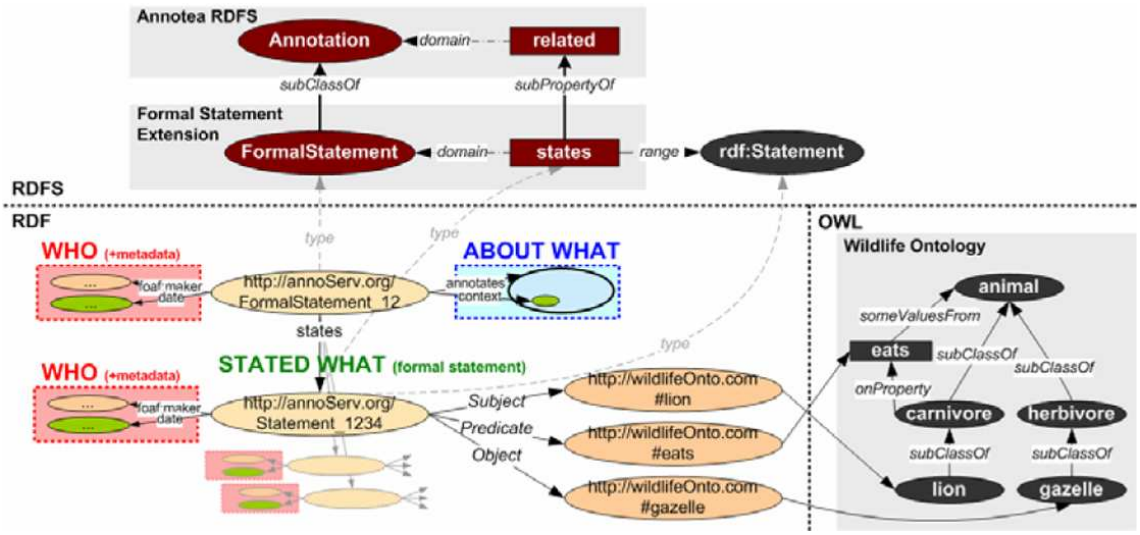



**Figure 2.6:** Extended Annotea schema and instance with a formal statement.

### 2.3.1 Annotea: Summary

The Annotea system presented in this section is an open and extensible annotation standard supported by the W3C community. Although initially aimed at free-text annotations of classical textual HTML sources, due to the various extensions proposed in particular by Schroeter and Hunter [SHN07] Annotea can be used to provide free-form as well as controlled (including ontology-based) annotations of a diverse content, including text, images, audio, and video. The extended Annotea also provides the ability to compare multiple resources (files) and annotate links between them.

## 2.4 Duality of Multimedia Data

In the previous subsections we have discussed the role of various kinds of metadata annotations of multimedia data. In this subsection we would like to demonstrate how the multimedia information can itself become metadata and what are the implications of this.

Due to its nature of providing information, data of any kind is characterized by a certain duality, namely that of *data vs. metadata*.

Take the example of textual data, which is one of the main forms of information in the cyberspace, providing, in particular, descriptions of various real-world phenomena. For instance, an article on a news agency's Web site reporting on results of municipal elections in Lausanne is textual data. At the same time, textual data can also itself be a target of textual descriptions. For instance, a historical novel (i.e. textual data) can be accompanied by (textual) comments from literary critics. These examples demonstrate the fact that one object's metadata can simultaneously become another object's data [GS00].

In a similar manner as the textual data, the other kinds of data also manifest the *data vs. metadata* duality. For instance, in the case of geographic data, a map can act both as metadata showing breakdown of vote results by Lausanne districts, and as a data source itself extended by a number of descriptions (e.g. map legends).

Clearly, audio-visual (multimedia) data are no exception to the *data vs. metadata* paradigm. On one hand, multimedia recordings can be seen as raw data, while on the other hand, they can be perceived as metadata providing audio-visual representation of phenomena they portray. Let's describe these two different visions of multimedia in more detail.

### 2.4.1 Multimedia as Data

The majority of existing multimedia-oriented applications perceive multimedia as the source data, meaning that multimedia recordings and their depicted content be-

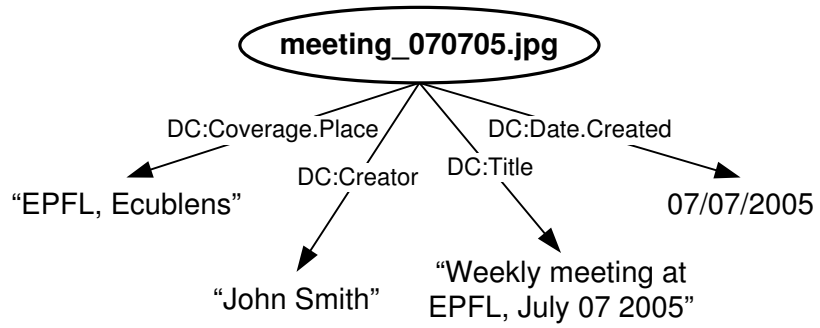



**Figure 2.7:** RDF and Dublin Core metadata descriptions of a JPEG file.

come the subject of modeling. An example of such applications are image or video archives, which are essentially composed of collections of digitized photos or movies stored on file servers or inside a database, and which are augmented with metadata corpora describing the media files as well as their visual contents. The multimedia recordings are thus becoming *data*, which are in their turn augmented by *metadata* like, for example, physical characteristics of image and video files, digitizing process parameters, contents access rights, etc. Tagging standards like Exif, XMP, ID3, etc. are among the most often used here. Moreover, metadata on multimedia data may also include descriptions of the semantic content of multimedia files, their represented objects and events. These metadata are often organized in more complicated structures stored outside the annotated multimedia files. Standards like Dublin Core, RDF, MPEG-7, etc., backed by advancements in ontology research, are among the most important for the latter type of metadata descriptions (see sect. 2.1 and sect. 2.2). For example, on fig. 2.7 a combination of RDF and Dublin Core qualifiers (XML namespace "DC") are used to provide metadata descriptions of a picture file.

It is important to note that the perception of multimedia as the source data becomes particularly important in the context of computer-created media, e.g. VRML worlds, synthesized music, digital drawings, etc., since in this case multimedia becomes the primary source of data, instead of merely a digitized representation of its hardcopy real-world original.

For the kind of applications presented above, we say that they adhere to the *multimedia as data* view, whose main peculiarities can be summarized as follows:

1. Annotations in a database must relate to a multimedia database entry.

   Example: In order to enter information about Paul in a digital photo archive application, Paul must appear on at least one photo stored in the system.

2. Existence of annotations in a database is governed by existence of their

corresponding multimedia entries.

Example: If all pictures of Paul are deleted from a digital photo archive, then the annotations about Paul must also be deleted.

3. Information organization is media-centric, and not annotation-centric.

Example: Instead of entering information about Paul first, and then eventually attaching to this data entry a list of photos where Paul appears, we rather annotate each photo in the system with a list of persons they represent (incl. Paul).

### 2.4.2 Multimedia as Metadata

As mentioned in sect. 2.4.1 , audio-visual and other types of multimedia are most often regarded as source data in their own right. We call this *multimedia as data* view.

It should be noted that the majority of multimedia data is obtained using commodity devices like photo- and video-cameras, microphones, etc. With media-recording equipment becoming increasingly affordable and compact, the number of digital media recordings (incl. photo, video, and audio) constantly increases. This notable growth has in particular been intensified by popularization of camera-enabled cell phones and other types of mobile devices combined with a quasi-ubiquitous connectivity via cellular and WiFi networks. In this context, emergence of systems and phenomena like Flickr, pod casting, mobile video blogging, etc. should be noted, just to mention a few.

Although the *multimedia as data* approach still prevails in the scenarios cited above, nonetheless the perception of multimedia as *metadata*, which visually (aurally) annotates recorded real-world phenomena constituting *data*, becomes more and more important. The need for this second vision of the essence of multimedia data comes primarily from applications, where multimedia recordings simply act as visual representations of real-world phenomena they portray. We call this second view *multimedia as metadata*.

According to *multimedia as metadata* view, multimedia data should be seen as a special kind of annotations, which adds to annotations of other types, i.e. alphanumeric, spatio-temporal, etc. Hence, integrating multimedia data in applications adhering to the second view of *multimedia as metadata* should pursue the goal of mapping multimedia information with other types (e.g. alphanumeric) of information about the same real-world phenomena that are visually/aurally represented by these multimedia recordings. For example, in a database for football games a video recording of a particular game should be considered not as the object of modeling, but rather as a special kind of annotation augmenting to traditional

alphanumeric annotations in the database (e.g. names of rival teams, the score, number of spectators on the game, etc.). The modeled object in this case should be the real game itself, rather than its multimedia depiction.

One of the examples of applications adhering to the second view of *multimedia as metadata* is virtual museums. Virtual museums are often regarded as web representations of real museums, which provide access to digitized collections of their real-world peers' exhibits. The examples of such projects are: Virtual Museum of Canada[8] and Lin Hsin Hsin Art Museum[9], just to mention a few. The textual descriptions of virtual exhibits provided by museum web sites (e.g. painter's name, epoch, etc.) actually describe the real-world exhibits rather than their virtual counterparts. Indeed, judging some historic artifact only by its picture on a museum's web site might not allow the spectator to appreciate such intrinsic properties of the exhibit as its physical measurements, the material it is made of, the pattern of its surface, or other characteristics describing the real artifact exhibited in the real museum. The artifact's digital photo on the web site thus acts not as the subject of description, but it becomes itself a part of metadata describing the visual appearance of the real-world artifact, just as classical alphanumeric metadata would describe artifact's place of origin, epoch, legend, etc.

The term *virtual museum* is also sometimes used by Virtual Reality researchers to refer to a number of projects aiming at recreating a 3D visualization of real-world places of interest (possibly not existing anymore) using techniques of virtual and augmented reality. Examples of such projects include ERATO [TCU+04] and CAHRISMA [PFMT03], which deal with virtual restoration of ancient theaters and mosques. Applications of this type also adhere to the *multimedia as metadata* viewpoint, since the multimedia data (in a special form of Virtual Reality worlds) is meant for describing the real-world objects (e.g. ancient edifices), which constitute the central point of interest of the applications.

Considering a more general case of Virtual Reality systems adhering to the *multimedia as metadata* viewpoint, we would like to mention a framework for providing semantic annotations for Virtual Reality applications presented in [KTCP07]. The approach described in [KTCP07] allows to add semantic annotations to Virtual Environments. The annotations can be provided not only in the text form, but may also be multimedia, i.e. images, videos, sounds. This approach correlates with the paradigm of *multimedia as metadata* in that it considers multimedia as simply one of the possible data sources providing information about the universe of discourse of the application.

[8] http://www.virtualmuseum.ca

[9] http://www.lhham.com.sg

Another example of applications adhering to the viewpoint of *multimedia as metadata* is multimedia meetings, which we describe below.

### Multimedia Meeting Framework

Multimedia meetings usually represent audio-visual transcriptions of talks among a number of participants discussing some topics either according to a pre-established schedule or in an unregulated manner. The multimedia recordings obtained during the meetings are stored in the data management system together with traditional text-based annotations, and can later be used to render meetings stored in the system, as well as to answer various media-related queries like "show Mike's reaction to John's proposition in the second part of the meeting", etc. It should be noted though that meeting data management systems store information about real-world meetings themselves, independently of the presence of any related multimedia recordings. In its turn, multimedia data, if present, simply provides additional annotations of audio-visual nature about the real-world meetings. In this connection, multimedia meeting applications view multimedia recordings not as the source data establishing the universe of discourse, but as a special kind of metadata about real-world phenomena.

Among the research projects dealing with multimedia meetings is IM2 Swiss national research project[10]. One of the objectives of IM2 is to devise a flexible annotation management framework for a multimedia database system applied to meeting recordings [BDJS04]. The notion of multimedia meetings in IM2 is associated with a Smart Meeting Room application dealing with interfaces and supporting facilities to store and retrieve both the raw media data produced at the meetings (e.g. video and audio recordings of the meetings), and the corresponding metadata produced after the meetings (namely, various annotations to describe, in particular, relevant segmentations of the audio and video files and, as far as possible, their semantic content).

Two typical examples of multimedia meetings in IM2 are shown in fig. 2.8 and fig. 2.9. Interaction between multimedia meeting participants can take place either in the form of monologues (a sequence of talks given by participants), or discussions (questions-and-answers, debates). Participants are also free to use a projection board or other visualization tools for demonstrating slides, diagrams, etc. The totality of multimedia meetings is recorded by a set of audio-visual recording equipment, which could be fixed-position, fixed-trajectory, or free-trajectory (cameraman-driven). In fig. 2.8 two wall-fixed cameras are filming 2 of the 4 meeting participants on each side of the desk, and a third camera is filming a projection screen on the

[10]National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2), supported by the Swiss National Science Foundation. http://www.im2.ch

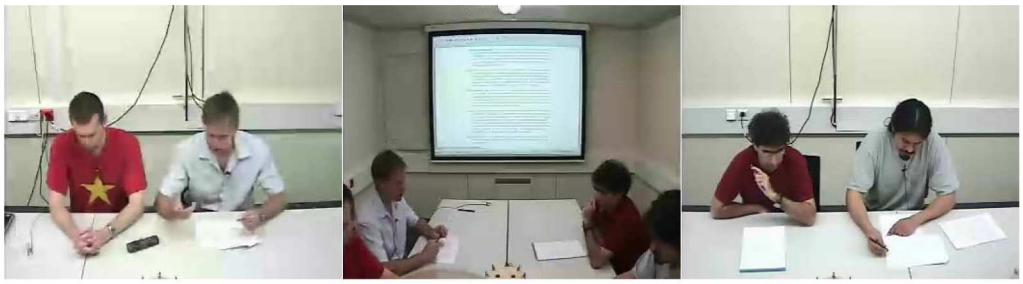


**Figure 2.8:** Multimedia meeting scenario #1.

**Figure 2.9:** Multimedia meeting scenario #2.

wall. In fig. 2.9 each meeting participant is filmed by a personal dedicated fixed camera.

Compared to traditional multimedia applications mostly adhering to the first view of *multimedia as data* (see sect. 2.4.1), multimedia meetings are characterized by a number of peculiarities that make traditional modeling approaches not quite suitable for meeting scenarios. Thus, for example, a great number of existing video information management systems are based on dividing video sequences into temporal components, namely: frames (single elementary pictures forming a video recording), shots (a single sequence of frames shot by one camera without interruption), and scenes (sequences of shots with the same time and locale). It has been argued that shots represent the finest level of descriptive granularity for motion pictures [VW03]. This reasoning, however, does not necessarily hold for the case of multimedia meetings. Indeed, in a setting like the one presented in fig. 2.9, where each participant is recorded by a personal video camera during the entire meeting, a shot-level division would be seriously hampered by a highly static pattern of video recordings, since except for some lip movements the picture we see in the recordings virtually does not change for the entire playtime. As for frame level division, this type of segmentation is way too fine and removes temporal aspects of video content [Dav95].

Another important peculiarity of multimedia meetings is the multitude of physical media sources. For example, each meeting participant can be represented by: a set of personal video files, parts of the video file for the projection screen, as well as parts of the common sound file. This markedly differs from the majority of classical single-media systems, often meant for video archives, where each media

file is considered as single, and is generally treated (i.e. annotated, queried, etc.) independently of the other files in the collection. It should be noted that unlike the motion pictures context, where montage is used to produce a single media file out of a series of independent footages, this solution is not appropriate to multimedia meeting scenarios. Unlike motion pictures, multimedia meeting applications do not seek to provide a unified multimedia view of the domain as seen by a single person or group of persons (e.g. movie director). Quite the contrary, in multi-user environments like that of the IM2, users/annotators are many, each having their own points of interest. It is thus important to preserve all the multimedia recordings, and make them available for multiple access and reuse by different users, thus prohibiting any possible semantic losses.

Further investigating the problem of multitude of physical media sources, we go on to yet another important characteristic of multimedia meetings, namely that of clearly separating semantic and physical aspects of multimedia. This, in our opinion, is one of the major requirements that a powerful multimedia modeling technique should meet. This requirement becomes all the more important when multimedia serves as a representation of real world entities (physical objects, relationships, events, etc.), while in this case the multimedia semantics actually reflects the semantics of the real world entities behind the multimedia recordings. For example, having John filmed during the meeting with a camera A, or a camera B, or not having him filmed at all, does not change the fact of John's taking part in the meeting. This means that we would like to be able to represent as much semantic information about this recorded meeting, as if we were doing so for the real meeting itself, and not just for its multimedia representation.

### 2.4.3 Section Summary

Summarizing the examples of applications adhering to the view of *multimedia as metadata* (see sect. 2.4.2), let's sum up the main peculiarities of this second perception of multimedia as compared to the classical view of *multimedia as data* (see sect. 2.4.1):

1. Application domain entities may not always have a multimedia representation.

   Example: Despite the fact of being out of sight of video recording equipment installed in a smart meeting room, John has nevertheless taken part in a meeting. The fact of John's participation should hence be reflected in the meeting database, even if this fact cannot be deduced from the meeting video recordings.

   This differs a lot from the classical approach of *multimedia as data*, as seen, for example, in digital movie archive applications, where only movies that are

part of the managed digital collection would be represented in the database.

2. Multimedia recordings do not always directly correspond to the concepts of the application domain.

   Example: Multimedia depiction of a meeting is composed of a sequence of several video files, which altogether as a group correspond to the visual representation of the meeting in question in the meeting database. However, each of these media files alone does not directly map to any of the modeling domain concepts, since breaking the video recording into files was simply dictated by the video tape length, and not by some domain-related concept like schedule, agenda, etc. Moreover, the video recordings that are used in multimedia representation of a meeting might also be reused to visually represent meeting participants, debate sessions, or other application domain concepts.

   Again, a scenario like this differs a lot from classical applications like digital image archives, where each multimedia recording represents a unit of modeling (e.g. each JPEG file represents one picture, and a "picture" is a unary concept in the application database schema).

3. Multimedia data are not always consistent with their corresponding real-world data.

   Example: For technical reasons related to recording, analysis, encoding, and other processing involved in multimedia data production, the information perceived through a multimedia recording may become inconsistent with the real-world information it portrays.

   Let's illustrate this with an example of a multimedia-enhanced HR database storing information about company employees along with their photos. The two photos shown in fig. 2.10 correspond to the same employee. However due to technical factors like lossy compression, color scheme shifts between various digitizing and visualization devices, etc., information perceived through these two photographs can become contradictory. Indeed, the person's eyes are hazel on the right picture, although in reality this person has grey eyes, which is also correctly represented on the left picture.

   The example above clearly shows that the classical *multimedia as data* approach (i.e. describing the person by its picture) would be unsuitable in this case, since it would lead to erroneous information in the system.

4. Multimedia representation tends to change.

**Figure 2.10:** Inconsistency in multimedia metadata.

Example: Multimedia data can continue to accrue during the life of an information object or system. Having entered the information about a last-week meeting in the database, including, in particular, its video summary, we might decide to change the corresponding video files with better quality ones possibly taken from a different spot. Obviously, this must not imply needing to delete and then reenter the meeting in question in the database.

On the contrary, in the case of digital movie archives, inserting a remake of an already existing movie in the system will create a new record in the database, since the data that the system is aimed at is the visualization of things (e.g. "Titanic" movie versions 1953 and 1997), and not the things themselves (e.g. the real story of RMS Titanic wrecked in 1912).

5. Multiple multimedia representations.

Example: Evolving the previous example of changing multimedia representations, it might also be required to store multiple multimedia representations of the same information object. Such a requirement may arise to deal with different user preferences and user profiles, to provide versioning support, to allow for multi-representations, etc.

Again, this example differs a lot from the digital movie archive scenarios, where multimedia data (i.e. movies), being themselves the subject of modeling, provide a unified multimedia view of the domain as seen by a single person or group of persons (usually, a movie director).

## 2.5 Chapter Summary

In this chapter we have described what the multimedia data is and the way we perceive it.

Besides a purely sensorial visual or aural aspect, multimedia data is also characterized by its semantic content, i.e. the semantic information that can be perceived via its multimedia representation. The latter kind of information is generally provided in the form of metadata annotations.

We have described two different approaches to representing annotations of multimedia data, namely controlled vs. free-form. While controlled annotations are relying on a centralized controlled vocabulary (e.g. taxonomy or ontology), the free-form annotation approaches can be described as democratic in that they do not impose any restrictions on the annotations, which can be produced by a potentially unlimited annotator base. Both approaches have their pros and cons, and examples of applications using either of the two approaches have been presented. We have also introduced a W3C annotation standard Annotea, which allows providing annotations of both types.

To better understand different possible ways of apprehending multimedia data, we have further introduced the duality principle, which allows perceiving multimedia information as either data or metadata. While the majority of existing multimedia information systems adhere to the viewpoint of *multimedia as data*, we have demonstrated the importance of the alternative viewpoint of *multimedia as metadata* for applications like those of the multimedia meeting framework presented within the chapter.

In the next chapter we introduce a novel multimedia-enhanced conceptual model, which allows efficiently considering not only the classical *multimedia as data* representation, but also the *multimedia as metadata* representation, making the model suitable for various kinds of multimedia applications.

# Chapter 3

# Conceptual Multimedia Modeling: MADS Multimedia Extension

In the previous chapter we have described the various aspects of multimedia data and the ways of perceiving it. In particular, we have pointed out the importance of the duality principle, which allows regarding multimedia information as either data or metadata. In this chapter we introduce a novel multimedia-enhanced conceptual model, which, in particular, is able to deal with the duality principle of multimedia data.

## 3.1 Requirement Analysis

In accordance with the first principle of conceptual modeling (also known as the 100% principle) [vG82], a conceptual schema should conform to the user's perception of the data domain by allowing to formally express the entire set of user requirements. Thus, in order to devise a conceptual model for multimedia applications we must first understand the requirements that such a model should meet.

Since we want to be able to cover not just the *multimedia as data* perception, but also the *multimedia as metadata* perception, the sought conceptual model should not be multimedia-centric, but rather multimedia-enhanced, allowing to represent not only multimedia-related information, but also the other types of information that possibly have no multimedia background. Hence, with a multimedia-enhanced conceptual model one should be able to express various application domain-related facts, like that of a person having a photo, but also the fact of a person having a phone number. These two facts are conceptually similar, and the only major difference between these two types of information is that the latter would most probably be represented in a database using alphanumeric data (a phone number), while representing the former would require recurring to multimedia data (a photo).

Being multimedia-enhanced instead of multimedia-centric implies, for example, that the lack of a photo of a person should not prohibit entering other kinds of infor-

mation about the person into the database. To draw a parallel between multimedia data and classical alphanumeric data, not knowing the phone number of a person must not hinder information about the person from being entered into the database[1].

According to the second principle of conceptual modeling (also known as the conceptualization principle) [vG82], a conceptual data model is by definition (and by design) not influenced by any implementation-related issues. Being a universal principle, it also has to apply to multimedia data. This implies that a multimedia-enhanced conceptual model should not be concerned with particular multimedia file formats (e.g. JPEG, AVI, MP3), compression standards (e.g. MPEG-1, JPEG 2000, etc.), multimedia network streaming configurations, or other implementation-related issues[2]. Instead, the model should allow depicting various multimedia-related facts pertinent to the universe of discourse. A conceptual schema designer should thus be able to represent facts like "a person has a photo", to distinguish among various types of multimedia information (e.g. picture, text, video, audio, etc.) including composite types (e.g. picture-audio-text), to specify various kinds of relationships between visual/aural/etc. contents of multimedia data, to specify multimedia-related integrity constraints (e.g. "each person has a mandatory passport photo, and can optionally have one additional photo"), and so on.

Another important requirement for a multimedia-enhanced conceptual model is multi-dimensionality. As mentioned in sect. 2.4.2, in applications adhering to the *multimedia as metadata* viewpoint, the multimedia data simply provides a special kind of descriptions of the universe of discourse, rather than makes up the universe of discourse itself. In particular, we have demonstrated that the presence or the absence of multimedia data should not govern the availability of other data-domain-related information in the database. Speaking the language of conceptual data modeling, the presence of multimedia-related concepts should not hamper any other concepts related to the universe of discourse from being represented in the conceptual schema. This requirement corresponds exactly to the idea pursued by the *orthogonality principle* [PSZ06].

Orthogonality applies whenever a conceptual schema should simultaneously consider several (possibly many) design issues. The way of handling this multitude of issues is by decomposing them into dimensions according to various modeling aspects. This allows handling each individual aspect in its own dimension independently from the other aspects, and to eventually combine all the partial solutions thus forming a global one. Multimedia-related concepts could thus be seen as one of the modeling

[1]For the sake of simplicity we assume here that the phone number and picture are not part of mandatory characteristics of a person, which would otherwise be characterized as NOT NULL in DDL SQL syntax

[2]For references on cited multimedia formats and standards, see [BK97] and [RBM02]

aspects, and form a separate independent multimedia modeling dimension.

The main advantage of the orthogonality principle is its weak susceptibility to the level of complexity, which becomes particularly important in multidimensional modeling (e.g. multimedia-extended modeling). Orthogonality also helps to simplify the model or make the model backward-compatible by providing a "trimmed" model, which discards some of the modeling dimensions that an application is not interested in or is not capable of treating. For example, discarding the multimedia dimension of a global conceptual model makes the model suitable for multimedia-disabled environments.

Having summarized in this section the requirements towards a multimedia conceptual model, in the next section we present MADS (Modeling Application Data with Spatio-temporal features), which in our opinion represents a good candidate model to be extended with multimedia modeling capabilities.

## 3.2 Introduction to MADS Conceptual Model

In the previous section we have presented a set of requirements to be met by our multimedia conceptual model. We have shown the importance of a multimedia-enhanced representation of the universe of discourse instead of a multimedia-centric representation. These requirements argue in favor of adopting a powerful extensible conceptual model suitable for representing information of non-multimedia types and further extending it with multimedia modeling capabilities. In this section we present an overview of MADS [PSZ06], which, as we will show, fully meets the requirements for such an extensible base model.

MADS (Modeling of Application Data with Spatio-temporal features) is a conceptual data model, focusing on taking into account modeling requirements of real-word applications. It provides a rich set of constructs in four complementary modeling dimensions, i.e., for modeling data structures, spatial features, temporal features, and multirepresentation features. MADS adopts an orthogonal perspective among the different modeling dimensions in order to achieve maximal expressive power. In the following subsections we describe the four existing modeling dimensions of MADS.

### 3.2.1 Structural Modeling Dimension

Structurally, MADS is an object-relationship data model. It allows schema designers to represent basic concepts from extended entity-relationship modeling, e.g., object types, relationship or association types, `IS_A` links, attributes, and methods. Fig. 3.1 shows MADS structural notation. Objects and relationships bear an identity and may have attributes. The attributes in MADS can be mono-valued

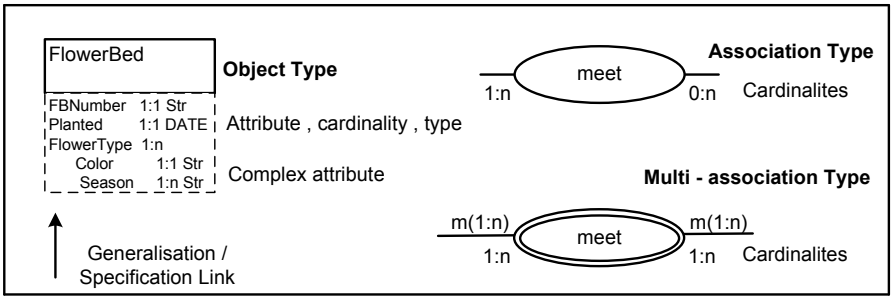



**Figure 3.1:** Samples of MADS structural notation.

or multi-valued, simple or complex (i.e. composed of other attributes). MADS relationship types are $n$-ary ($n \geq 2$), i.e., they have two or more roles, where a role is a link between the relationship type and a given object type. The relationship types may be cyclic with two or more roles linking the same object type. Two kinds of relationships provide the basic constructs to link objects together: *associations* and *multi-associations* [PSZ06].

*Association* is the most universally-known kind of relationship type. An association type links two or more object instances without imposing any specific semantics on the link. Associations are non-directed links and may have any number ($\geq 2$) of non-pending roles. For each association type role, its minimum and maximum cardinalities define the number of relationship instances that may link an instance of the object type linked by the role.

*Multi-association* is the most general kind of linking construct. Whereas association relationships limit to one instance per role the number of instances linked by the relationship, each role of the multi-association relationship links a non-empty set of instances of the linked object type. Consequently, each role bears two pairs of (minimum, maximum) cardinalities. A first pair is the traditional one that defines for each object instance, how many relationship instances it can be linked to via the role. The second, additional, pair defines for each relationship instance, how many object instances it can link with this role. The minimum cardinality on the relationship side is 1, which is required to avoid pending roles, and in the case of the maximum cardinality also equal to 1 the multi-association simply becomes an association.

The fig. 3.2 shows an example of a MADS schema, which uses only the structural dimension of MADS. The object types `RoadSection` and `CrossRoad` linked by a relationship type `meet` model the universe of discourse, which considers road sections, some of which meet crossroads. According to the cardinalities of the `meet` relationship, a road section can begin or end with 0 to $n$ crossroads and a crossroad can be composed of $m$ to $n$ road sections, where $m \geq 2$.

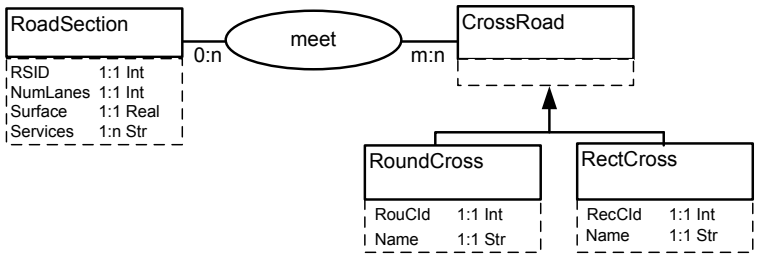



**Figure 3.2:** Structural dimension of MADS.

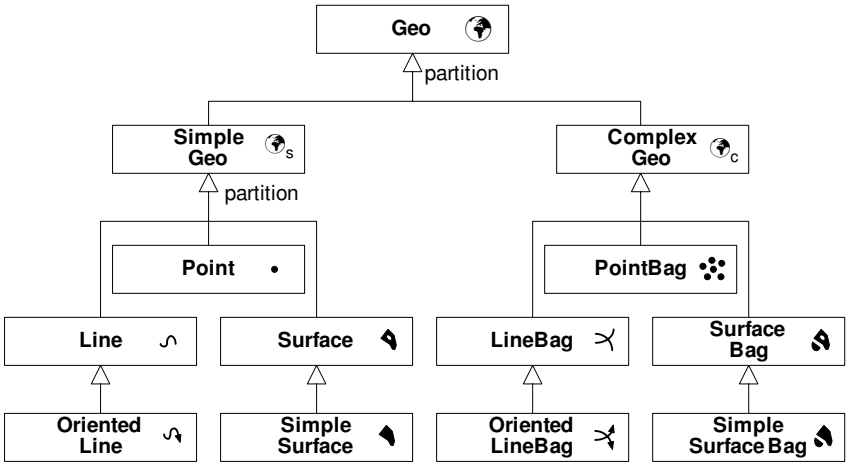



**Figure 3.3:** Basic MADS spatial datatype hierarchy.

### 3.2.2 Spatial Modeling Dimension

The spatial dimension of MADS allows to include the description of the spatial properties of real-world phenomena represented in a database schema [PSZ+98]. Objects and relationships are spatial if they have an associated spatial extent. In MADS a predefined set of spatial data types with associated operations and predicates provides the domains of values for the spatial extents. These datatypes are: `Point`, `Line`, `OrientedLine`, `Surface`, `SimpleSurface`, `SimpleGeo`, `PointBag`, `LineBag`, `OrientedLineBag`, `SurfaceBag`, `SimpleSurfaceBag`, `ComplexGeo`, `Geo`. The fig. 3.3 shows the hierarchy of MADS spatial datatypes as well as the icons denoting each datatype. The most generic spatial datatype `Geo` generalizes the `SimpleGeo` and the `ComplexGeo` datatypes with the semantics: "this element has a spatial extent" and without any commitment to a specific spatial datatype. The three spatial datatypes mentioned above are abstract and are never instantiated. The spatiality of an element may either be defined precisely e.g., `Point`, `OrientedLine`, or left undetermined, e.g., `Geo`.

Let's consider the following example. The MADS schema in fig. 3.2 can be enriched with the spatial semantics. As shown in fig. 3.4, a road section can be

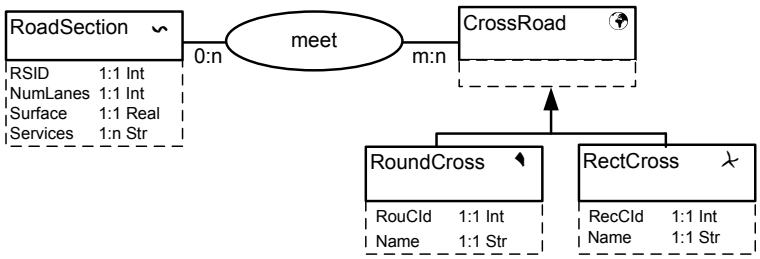



**Figure 3.4:** MADS sample schema: structural and spatial.

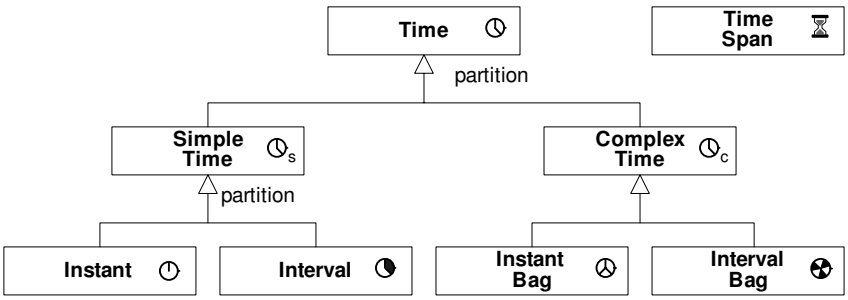



**Figure 3.5:** Basic MADS temporal datatype hierarchy.

modeled with a geographic domain of the spatial type `Line`, and a crossroad with the geographic domain of the spatial type `Geo`. Obeying the hierarchy in fig. 3.3, `RoundCross` and `RectCross` are modeled with the spatial datatypes `SimpleSurface` and `LineBag` respectively.

### 3.2.3 Temporal Modeling Dimension

The temporal dimension of MADS allows to include the description of the temporal properties of real-world phenomena represented in a database schema using the mechanism of time stamping. Time stamping is the traditional way of modeling temporal information. Time stamped attribute values allow expressing when an attribute value was, is, or will be holding in the real world as perceived by the application (valid time) or as per when it was known in the database (transaction time). Time stamps of objects and relationships convey their life cycle information: when an object or relationship was created, suspended, reactivated, or deleted. Object and relationship time stamps are also based on either valid time or transaction time. Currently, MADS supports valid time. It is also important to note that the spatiality/temporality of an application is reflected by the existence of spatial/temporal entities, but also by the existence of space- or time-related relationships between these entities. Fig. 3.5 shows the hierarchy of temporal data types in MADS.

In fig. 3.6 we expand the sample road section schema from the previous subsec-

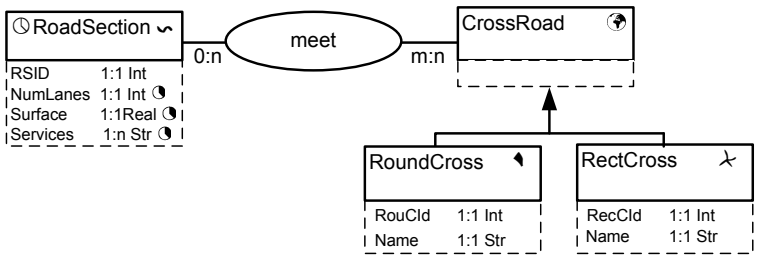



**Figure 3.6:** MADS sample schema: structural and spatial and temporal.

tions by expressing temporal properties of the elements of the schema in fig. 3.4. The entity type `RoadSection` is modeled as a temporal element of type `Time`, meaning that it has a temporal extent with no commitment to a specific temporal datatype. The attributes of `RoadSection` are modeled with the `Interval` temporal semantics.

### 3.2.4 Constraining Relationships

In MADS *constraining relationship type* is a kind of relationship type that bears a specific spatial or temporal predicate on the geometries or life cycles of the linked object types. MADS includes topological and synchronization relationships as built-in constrained relationship types. For example, a topological relationship type `TopoTouch` may be defined to link object types `CrossRoad` and `RoadSection`, expressing that according to the application semantics the geometry of a road section does not intersect with but is adjacent to the geometry of a crossroad. The list of predefined topological relationship types in MADS and their associated icons are shown in table 3.1.

| *Topological Type* | *Icon* | *Topological Type* | *Icon* |
|---|---|---|---|
| TopoDisjoint | | TopoTouch | |
| TopoOverlap | | TopoCross | |
| TopoWithin | | TopoEqual | |
| TopoGeneric | | | |

**Table 3.1:** MADS topological relationships.

Similarly to topological relationships, the synchronization relationships allow specifying constraints on the life cycles of their participating objects. They allow in particular, to express constraints on schedules of processes. MADS predefined synchronization relationship types are shown in table 3.2.

| *Synchronization Type* | *Icon* | *Synchronization Type* | *Icon* |
|---|---|---|---|
| SyncPrecede | | SyncMeet | |
| SyncWithin | | SyncOverlap | |
| SyncStart | | SyncFinish | |
| SyncEqual | | SyncDisjoint | |
| SyncGeneric | | | |

**Table 3.2:** MADS synchronization relationships.

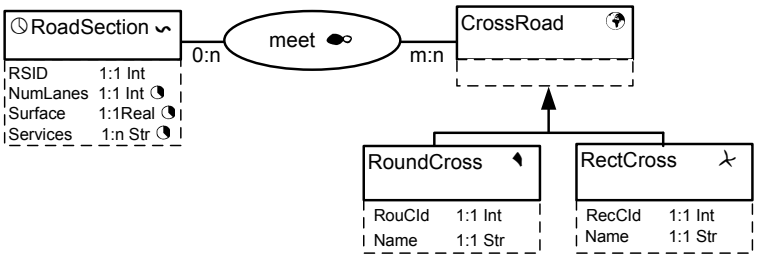



**Figure 3.7:** MADS sample schema: topological relationships.

Let's consider the following example. In fig. 3.7 the relationship type `meet` is enriched with the semantics of the the topological relationship `TopoTouch`. As for the synchronization relationships, in this example we cannot add any type of synchronization semantics to the `meet` relationship, since the object type `CroosRoad` has no temporal extent.

### 3.2.5 Multiple Representations and Multiple Perceptions

Multiple representations allow to define in the same MADS schema several different representations of the same real world object [Van04]. These different representations may be the consequence of diverging requirements during the database design phase or, in the case of spatial data, of the description of data at various levels of detail. The support for multiple representation has been added in MADS via an additional orthogonal dimension.

To allow users to retrieve specific representations from the set of all existing representations, representations have to be distinguishable and denotable. In this regard, perception stamps are placed on data, be it object type instances, attribute values, metadata, object or relationship type definitions, or attribute definitions. Stamps are vectors of values characterizing the context of each perception, e.g., spatial resolution, viewpoint. Object and relationship types may be perception-varying types and thus have a different set of attributes depending on the considered perception.

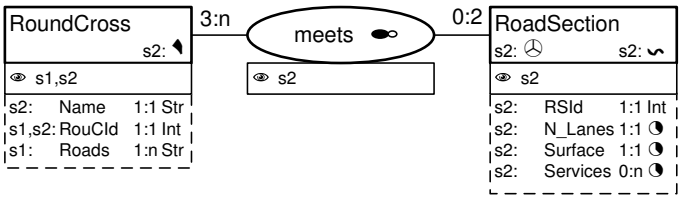



**Figure 3.8:** Perception-varying object type `RoundCross`.

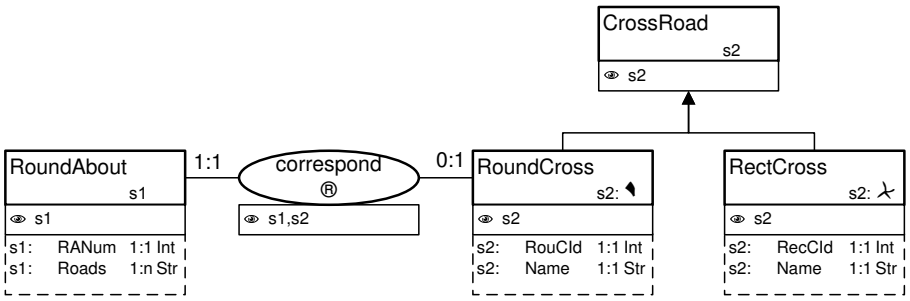



**Figure 3.9:** Two mono-perception object types related by the `correspond` inter-representation link.

Let's illustrate the multirepresentation features of MADS with the following example. In fig. 3.8 the object type `RoundCross` is a multirepresentation type with two definitions, one for stamp $s_1$ with the attribute `Roads`, and one for stamp $s_2$ with the attribute `Name`. The attribute `RouCId` exists for both stamps $s_1$ and $s_2$. As we have mentioned above, the perception stamps can be added on the level of attributes as well as on the object or relationship levels. For example, in fig. 3.8 the object type `RoundCross` is described by only the structural dimension in the stamp $s_1$, however becomes spatial in the stamp $s_2$ with `SimpleSurface` as its spatial type. In the perception with the stamp $s_2$, the `RoundCross` object type is related through a topological relationship `meets` to another spatial object type `RoadSection`. The relationship `meets` in this schema is a constrained relationship, with one of the constraints being the spatiality of the related object types. In general, relationship types may hold several different semantics depending on the representation and, for instance, be a topological relationship in one representation and a synchronization relationship in another.

Also pertaining to the multirepresentation dimension of MADS is a specific inter-representation semantics that may be applied to both associations and multi-associations to denote that the linked object types describe instances that are different representations of the same real world object. In fig. 3.9 the same real world objects are modeled by an object type `RoundAbout` in a schema with the stamp $s_1$ and by an object type `RoundCross` in a schema with the stamp $s_2$. The inter-

representation relationship type `correspond` links the two mono-perception object types. The cardinality of this relationship states that for each instance from the class `RoundAbout` there exists exactly one instance of type `RoundCross` and that there may be at most one instance of type `RoundAbout` for each instance of type `RoundCross`. These cardinality constraints of the `correspond` relationship convey the information on how the populations of the related object types are related. According to the cardinalities in fig. 3.9, the population of `RoundAbout` is included in the population of `RoundCross`. Besides the relationship between the sets of instances, or populations, the inter-representation semantics does not induce any constraints between the linked objects.

### 3.2.6 On Extending MADS with Multimedia Semantics

In the previous subsections we have introduced the MADS model and have pointed out its major strengths like structural completeness, orthogonality, spatio-temporal modeling, multi-representation, etc. We believe that MADS constitutes a powerful conceptual modeling technique and propose to use it as the base model for multimedia-enhanced systems.

In order to extend a regular MADS conceptual model with multimedia information pertinent to the universe of discourse, we propose to introduce a new *multimedia modeling dimension* [DS06]. As described in previous subsections, orthogonality principle of MADS substantially simplifies multidimensional modeling by decomposing various modeling aspects into a number of dimensions, and treating each dimension independently of the others. Adhering to this principle, we introduce a new multimedia modeling dimension, which is orthogonal to already existing thematic, spatial, temporal, and multi-representation dimensions. This, in particular, allows extending initially multimedia unaware applications with multimedia semantics. Also the applications that are not interested in or are not capable of dealing with multimedia information can work with a pruned version of the conceptual schema ignoring the multimedia dimension. Due to the orthogonality principle, trimming the multimedia dimension does not influence the other modeling dimensions and allows keeping the schema backward-compatible for legacy applications.

In spite of independence between multimedia and other modeling dimensions, modeling multimedia information should follow the same principles as, for example, modeling spatial or temporal information. As stated in [PSZ06], similarity represents a very important factor in multidimensional modeling. Making the modeling process similar across various dimensions, a user can easily grasp the general idea using one dimension and easily apply his skills onto other dimensions. Thanks to this approach, increasing the number and diversifying the focal points of modeling dimensions only marginally complicates the modeling task in general

and subsequent user retraining in particular. Therefore, the multimedia modeling concepts that we introduce try to follow the spirit of other modeling dimensions in MADS.

To make a regular MADS schema account for multimedia semantics we should enrich it with information of multimedia nature pertaining to the universe of discourse. For instance, we can specify that besides a `name`, `SSID`, `address`, and `e-mail` an object type `Professor` is also characterized by a `picture`. Associating a picture to a `Professor` makes it a multimedia-enhanced object type. Furthermore, knowing that the object type `Faculty` is also characterized by a `picture`, we might require that the picture of a professor makes part of the picture of his respective faculty. Presence of application domain-related multimedia information allows answering queries like "what does the database professor look like?", or "show me the pictures of his colleagues from the faculty". Due to orthogonality of MADS modeling dimensions, the `Professor.picture` attribute might additionally be defined as time-varying, whose value changes each year (e.g. due to the university's internal policy of annually updating its staff's photos). Furthermore, in a multi-representation environment we might specify `Professor.picture` as having different values in different representations (e.g. a passport-format photo to use in official administrative settings, and an informal photo taken at the recent scientific conference to use in research- and academic-related environments).

In the sections to follow we describe how we have addressed the multimedia modeling requirements depicted in these examples.

## 3.3 Multimedia Datatypes

In order to conceptually represent facts like "a professor has a picture", new multimedia data types must be introduced. Indeed, classical data types such as float, char, integer, etc. are not suitable to represent highly unstructured multimedia data like images, videos, or sound. It is important to note that although the majority of traditional DBMS provides data types for storing chunks of binary data, which are particularly meant to store raw multimedia data like images and videos within a database (e.g. data types `RAW`, `BLOB`[3], etc. available in Oracle DBMS), using these data types at conceptual level is inappropriate. Indeed, due to their physical raw media orientation, BLOB-like data types are not suitable in situations where multimedia information does not come from a set of linear files on a hard disk, but is instead organized in complex hypermedia documents (see sect. 4.1). Moreover, with the same BLOB-like variable being able to store an image, a video clip, or even whatever non-multimedia binary data, the BLOB-like data types cannot convey the

[3]A binary large object (BLOB), is a collection of binary data stored as a single entity in a database.

semantic meaning of the particular type of multimedia information they represent.

A number of attempts to define a set of multimedia data types have been made (see e.g. [RKN96] and [AN97]). In M data model [DC98] a multimedia type is defined as an entity type whose contents is most meaningful when displayed or presented in a way that is directly perceivable by the human senses. A structure of six basic multimedia types (`Image`, `Image Stack`, `Sound`, `Speech`, `Video`, and `Long Text`) is proposed, which can be expanded if necessary by either adding new types or by extending the already existing ones. Despite their internal complexity, multimedia types are thought to be seen by users as encapsulated black boxes, which can be manipulated via operations and methods, just like any other traditional alphanumeric data types are. A similar idea is implemented in Oracle interMedia feature of Oracle DBMS [Cor07], which provides a set of predefined multimedia object types. The most significant of them are `ORDAudio`, `ORDImage`, and `ORDVideo`, which represent respectively audio, image, and video data originating from a variety of sources including `BLOB` within the same database, external files on a local file system or another server, or from an external HTTP server. Each of the ORD* multimedia data types provides a set of methods for extracting metadata and attributes from multimedia data, reading and saving multimedia data from/to external sources, performing some manipulation operations.

It should be noted that in the above examples of systems introducing multimedia value domains, almost no attention is given to complex data types. Most of the provided multimedia data types only allow representing information of just one single nature (e.g. solely a picture, or solely a video clip, etc.). Although working with only simple multimedia data types suffices for a fairly large number of existing applications (e.g. digital image collections), this approach is not suitable for hypermedia-oriented applications working with composite multimedia documents that combine a multitude of media sources of possibly different nature (see sect. 4.1). Although the `ImageStack` data type, introduced in the M data model [DC98], allows representing sets of logically related images, the model does not generally provide data types for representing sets of multiple (possibly heterogeneous) media sources. To deal with complex multi-type multimedia elements in M, a notion of multistreams is used. A multistream is an aggregation of streams, which are combined and synchronized to form a new composite stream. Thus, for example, a regular audio-video clip would be represented in M by a multistream consisting of one video stream synchronized with one audio stream. Nevertheless, an approach like this is not quite suitable at the conceptual design stage, since the idea of streams and multistreams is rather physical-level-oriented and conveys the idea of physical organization of multi-track streaming media data. This approach is furthermore particularly unsuitable in *multimedia as metadata* environments, since it would be absolutely inappropriate, for example, to have to conceptually represent a participant of a meeting by a synchronized set of two semantically equal objects

types, with one of them bearing the participant's video representation, and the other one bearing his audio representation.

The reasoning above clearly demonstrates the need for means of representing directly at the conceptual level the existence of application-related multimedia information of complex heterogeneous types. In this connection, we propose to introduce a set of complex multimedia datatypes, which together with simple datatypes like `Image`, `Video`, `Audio`, etc. will be able to fulfill the requirements of a broad range of multimedia applications, including hypermedia-enabled ones. Some examples of such datatypes could be, e.g. `ImageText`, which would fit to represent multimedia documents corresponding to classical Web pages, or `ImageSound`, which would correspond to digitized music recordings accompanied by pictures of their CD covers, etc. A complete list of multimedia data types in MADS can be found further in this section.

Besides providing a rich set of various simple and complex multimedia data types it is also important to organize them in a hierarchical structure much as it is done for spatial and temporal data types in MADS (see sect. 3.2). Hierarchical organization becomes particularly important in situations where the exact type of a multimedia element is not known a priori and can generally be one of a set of several multimedia data types (e.g. multimedia representation of a meeting could be a still picture or a video). In this case, having all the data types organized in a hierarchy, we can specify the value domain of the multimedia element in question as the lowest common super type of the set. A more general example of a situation where having multimedia data types organized in a hierarchical structure becomes advantageous is when a user has no precise idea about the possible value domains of a multimedia element, or else expressly does not want to limit them to any particular data type or set thereof, thus allowing for whatever multimedia representation supported by the system. In this case, in order to still be able to specify that an object type bears a multimedia representation, the most general (top-most) multimedia data type in the hierarchy can be used.

An idea similar to that of using a multimedia super type to cover all its possible subtypes is implemented, for instance, in Oracle interMedia [Cor07]. The `ORDDoc` multimedia data type (see fig. 5.1) supports the storage and management of heterogeneous media data including image, audio, and video (i.e. types `ORDImage`, `ORDAudio`, and `ORDVideo` respectively). `ORDDoc` is generally meant for situations where the specific type of a multimedia column in a table can change from one row to another. From the point of view of relational database modeling concepts, using `ORDDoc` in such situations spares the database designer from having to resort to some clumsy modeling tricks like introducing a separate column for each possible data type and implementing a check-constraint or a trigger to make sure that at most one such column per table row is used.

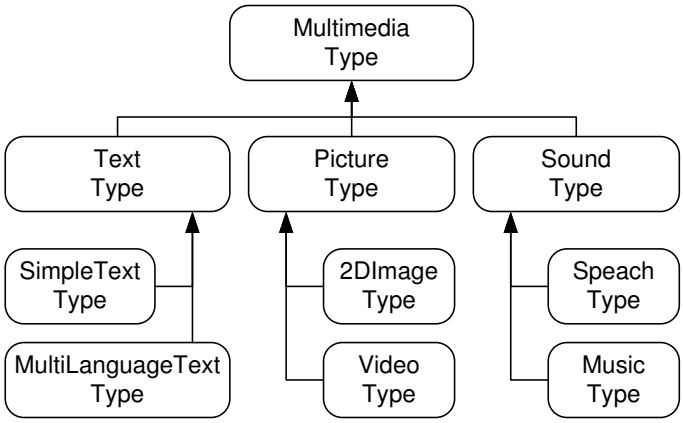



**Figure 3.10:** Example of a 2-level multimedia datatype hierarchy.

In addition to the above-described scenarios, employing multimedia super types can also be useful in situations where the data type hierarchy is a priori not statically defined, and can possibly evolve by being extended with new members. Using a multimedia super type to define the value domain of a multimedia entity would enable it to represent values of data types not known at the design time. In this context, it becomes particularly important to provide a multi-level data type hierarchy by refining upon granularity of multimedia super types. The latter should also carry sufficient semantic descriptions of criteria by which a group of subtypes is arranged under a certain super type. For example, instead of arranging all multimedia data types under the same and only one super type in the hierarchy, as it is e.g. the case with Oracle interMedia, we could provide a hierarchy like the one presented in fig. 3.10, where multimedia data types are arranged in a tree-like structure of depth 2 with leaves representing particular data types, grouped under 3 super types, which are, in their turn, grouped under the top-most super type (tree root).

Grouping data types in sub-trees under a certain super type is determined by grouping criteria associated with each super type (non-leaf vertex) in the hierarchy. In the structure shown on fig. 3.10 these criteria simply rely on the names of the 4 super types, which are presumed to be explicit enough to describe the type of information they represent (i.e. textual information, graphical information, and aural information). However, in more sophisticated environments some complex metadata descriptions using WordNet [Voo98] [MF07] or domain ontologies [JPA06] could be used to reproduce proper semantic meaning of grouping principles of each super type in the hierarchy. In general, such semantically enhanced multi-level hierarchical structures of multimedia data types can help the database designer to make the model more precise by using the most specific super types instead of simply using the topmost one. Moreover, it becomes easier to add a new data type to the hierarchy by putting it under a super type whose semantic description corresponds the most to the new type that is being inserted. For example, if a new multimedia data type `Hypertext Type` were to be added to the hierarchy on fig. 3.10, it would

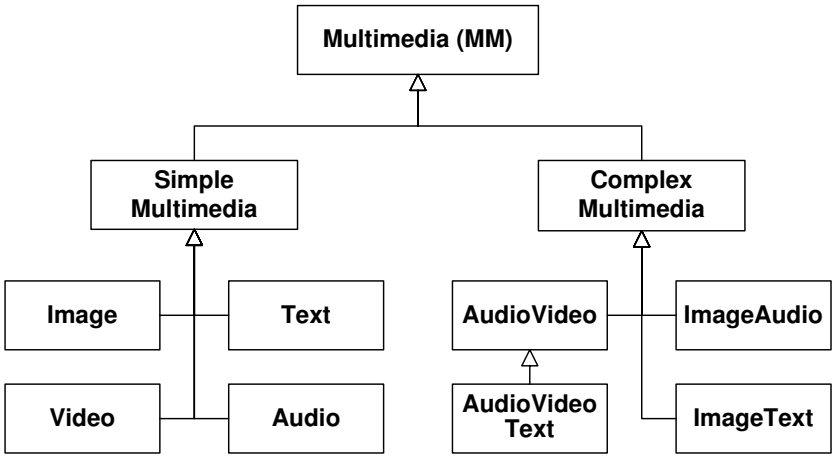



**Figure 3.11:** MADS multimedia datatype hierarchy.

logically be attached to the sub-tree beneath the `Text Type` super type. In this case using a multi-level hierarchy instead of a single-level one enables only the multimedia elements of type `Text Type` to represent data of the new type `Hypertext Type`, while, for instance, the elements of type `Sound Type` are, naturally, not able to store hypertext information.

### 3.3.1 MADS Multimedia Datatype Hierarchy

Taking into account the above argumentation, the fig. 3.11 introduces a multi-level hierarchy of multimedia data types to be used in MADS multimedia dimension.

The hierarchy in fig. 3.11 is divided in two sub-hierarchies, namely simple data types and complex data types. Simple data types correspond to mono-media data like picture, sound, text, etc., while complex data types correspond to composite media, i.e. truly multi-media data, like audio-video, picture-text, etc. Due to the multilevel structure of the hierarchy in fig. 3.11, users can choose among several super types in situations when the exact data type is not known a priori, or when a certain level of generalization is expressly required.

Let's semantically define the multimedia datatypes depicted in the hierarchy in fig. 3.11.

`Image` denotes two-dimensional images portraying some existing or imaginary objects. Digital images are either pixel-based, or geometrical-primitives-based (vector graphics). Images are usually taken with some specialized devices like photo cameras, scanners, graphic tablets, etc., or are else produced either automatically or semi-automatically by image editing software. Examples of images include photo camera shots, document scans, stills from computer animation, etc.

`Text` denotes character-based information generally expressed in some natural language. Besides the useful information itself, a text may also contain some overhead information usually relating to text layout formatting. Examples of text data are ASCII and Unicode unformatted text documents, as well as layout-enhanced text documents like RTF, PDF, HTML, etc.

`Video` denotes data representing moving pictures, and which is primarily intended for viewing on monitors or projection screens. Video is usually seen as a sequence of images (frames) displayed at a certain frequency (frame rate). Consequently video data has a temporal dimension to it, i.e. a timeline. This means that in order for a video sequence to be entirely displayed on a visualization device, some amount of time is required, which at least corresponds to the duration of the video. It must be noted that `Video` is a simple data type and it does not have any sound component incorporated into it. Examples of video data are video tracks of DVDs, video recordings from surveillance cameras, etc.

`Audio` denotes data representing sound waves. Audio information is perceived by hearing organs and is usually reproduced by sound speaker devices. Like video data, audio information also has a temporal dimension, i.e. it is characterized by duration and speed. Examples of audio data are speech samples, digitized music, etc.

`SimpleMultimedia` is a super type generalizing all simple media data types, i.e. data types `Image`, `Text`, `Video`, and `Audio` in case of the particular hierarchy in fig. 3.11. This super type is used for variables whose values can represent information of several (or all) simple data types. As such, the `SimpleMultimedia` is totally abstract and hence cannot be directly valued. Any value of this super type must be a value of one of its underlying subtypes. For example, `SimpleMultimedia` could be used to define the value domain of a variable, whose values could either be of type `Image` or of type `Text`.

Besides four simple data types, the hierarchy in fig. 3.11 also introduces four complex data types, which combine together different types of media.

`AudioVideo` denotes audiovisual data that combines both audio and video streams, which are usually timeline-synchronized. Examples of audiovisual data are movies, TV news, homemade recordings from a video camera with a microphone, etc.

`AudioVideoText` is a subtype of `AudioVideo`, and denotes audiovisual information superimposed with text. An example of this type of multimedia information is movies with subtitles.

`ImageText` denotes textual information combined with image data. Some

examples of this type of multimedia information are classical HTML pages, or PDF documents consisting of text and pictures.

`ImageAudio` denotes audio information enhanced with an image. Examples of this type of multimedia information include digitally distributed musical recordings with a picture of their corresponding CD cover, or TV news coming from a reporter telephoning to the studio without a video footage available.

`ComplexMultimedia` is a super type generalizing all complex data types, i.e. data types `AudioVideo`, `AudioVideoText`, `ImageText`, and `ImageAudio`. Just like `SimpleMultimedia`, this super type is totally abstract and hence cannot be directly valued. Any value of this data type must be a value of one of its underlying complex subtypes.

Finally, `Multimedia` is the most generic multimedia super type, which does not provide any specific details about the particular type of information it represents, but rather simply affirms its multimedia affiliation. Thus, `Multimedia` is a generalization of types `SimpleMultimedia` and `ComplexMultimedia`, and just like the latter two it is totally abstract and hence cannot be directly valued. Any value of this super type must be a value of either one the four simple types, or one of the four complex types. For example, a variable of datatype `Multimedia` could have values of types `Image` or `AudioVideo`.

The hierarchy in fig. 3.11 is definitely not exhaustive, and simply represents the most used, in author's point of view, types of multimedia information relating to the following two basic human senses: vision and hearing. In order to better adapt to specific user requirements, any additional multimedia data types can be introduced into the hierarchy. In that case, new data types either refine upon already existing hierarchy members (whether final types or super types), and thus become added to the hierarchy as subtypes (e.g. subtypes `Music` and `Speech` that could be introduced into the hierarchy in fig. 3.11 as subtypes of `Audio`), or else introduce totally new types of media (e.g. sensor information, smell, etc.), thus forming completely new branches in the hierarchy tree. Should additional multimedia datatypes be introduced, a semantic description of their essence can be provided. In order for such descriptions to be correctly and equally understood by various participating parties (e.g. schema designers, end users, etc.), they could be based on commonly accessible ontological descriptions.

Besides extending the hierarchy in fig. 3.11, a trimmed version of the hierarchy could also be elaborated in order to better adapt to some specific application requirements. This particularly makes sense for applications that only support some limited set of media types, e.g. only textual and image data. In that case, simple and complex data types not relating to either image or text (e.g. video and

audio) could be removed from the data type hierarchy in order to not make the model too heavy.

Using various particularized data types increases semantic precision of the multimedia-enhanced conceptual schema, since instead of using one generic "multimedia data type" we specify which particular kind of multimedia information is in question. Moreover, multimedia data types also provide some general metadata about the multimedia information they represent. This metadata may include, for instance, the MIME type, the video resolution, the sound wavetable scheme, etc. Nevertheless, it should be noted that individual multimedia data types only provide general semantic specifications about the kind of data they represent, and it would be absolutely inappropriate at the conceptual level to differentiate between different multimedia file formats by providing, for example, separate data types for GIF images and TIFF images. Indeed, this latter characteristic pertains to logical and physical organization of image information, and hence, according to the second principle of conceptual modeling (see sect. 3.1), should not be considered at the conceptual modeling stage.

### 3.3.2 Application of Multimedia Datatypes in MADS

In general, in order to be able to represent the broadest range of information of multimedia nature pertinent to the universe of discourse, any phenomenon reflected in a conceptual model must be able to manifest its multimedia-related semantics regardless of the way this particular phenomenon is represented in the model. Fortunately, the orthogonality principle peculiar to MADS modeling dimensions allows fulfilling this requirement. Considering various data structures available in MADS thematic dimension, the potential multimedia extensions must be applicable to objects, relationships, and attributes. From this point of view, the area of applicability of multimedia semantics within a MADS schema is the same as for spatial or temporal semantics, namely: object and relationship types, and attributes.

A *multimedia attribute* is a simple attribute (either mono- or multi-valued), whose value domain is one of the multimedia data types. Any object or relationship type is free to contain whatever number of multimedia attributes. For example, in a HR conceptual schema, an object type `Employee` can have two multimedia attributes: `Employee.picture` of type `Image` and cardinality `1..2` to store one or two photos of the employee, and `Employee.voice_sample` of type `Audio` to store a recorded sample of employee's voice to be used in the recognition module of the automated access control system throughout the company premises.

A *multimedia object type* is an object type, which as a whole is characterized by some multimedia information, whether this multimedia information conditions our knowledge of the object existence or simply provides an optional multimedia

depiction thereof (i.e. regardless of the *multimedia as data* or the *multimedia as metadata* approach that is used). The determining characteristic here is affiliation of the multimedia data with the object as a whole, and not just some of its constituents. Should this condition not be met, such modeling constructs as multimedia attributes and multimedia relationship types (see below) ought to be used instead.

It must be noted that the decision whether the multimedia information is intrinsic to the object type as a whole, or simply pertains to some constituents thereof, is highly subjective and is generally taken by the schema designer according to his personal preferences. Thus, referring to the example above, the employee's voice sample might be considered not as simply one of the many attributes of the employee (`Employee.voice_sample`), but rather as the primary and essential multimedia characteristic of an employee (this would most probably be the case for a conceptual schema of a voice-recognition-oriented application). In that case, the voice sample should be modeled not by a multimedia attribute within the object type `Employee`, but rather be associated with the entire object type. Nevertheless, similarly to spatial and temporal dimensions, multimedia information at the object type level is stored in a dedicated attribute with the reserved name `multimedia`, whose value domain is the appropriate multimedia data type.

A *multimedia relationship type* is a relationship type that, similarly to a multimedia object type, is as a whole characterized by some multimedia information. At the relationship level, multimedia semantics can also be alternatively conveyed by multimedia attributes in relationship types. Again, it is up to the schema designer to choose whatever approach is the most appropriate. Just like with multimedia object types, the relationship type level multimedia information is stored in a dedicated attribute with the reserved name `multimedia`.

It should be noted that a single object or relationship type can simultaneously bear any combination of spatial, temporal, and multimedia semantics. For example, an object type `Employee` can be at the same time spatial and multimedia, or e.g. temporal and multimedia. The latter case would naturally correspond to a situation when besides a multimedia representation, the object type `Employee` has a `lifecycle` to differentiate between employee's various statuses like `scheduled` (e.g. a new employee is hired and comes to work next week), `active` (e.g. an employee is currently active), `suspended` (e.g. an employee is temporary on leave until the end of the year), or `disabled` (e.g. an ex-employee that quit the company a year ago). Moreover, an object that is at the same time temporal and multimedia can alternatively be characterized by a time-varying multimedia representation. An example of such situation is a picture of a merchandise on an on-line shop web site. Depending on the availability of stock, which can be characterized by altering statuses of the merchandise (e.g. "active" for items in stock, "suspended" for items temporarily out of stock, "disabled" for items not sold anymore by the shop, etc.), the picture representation of goods also changes accordingly (i.e. it


State="active" State="suspended" State="disabled"

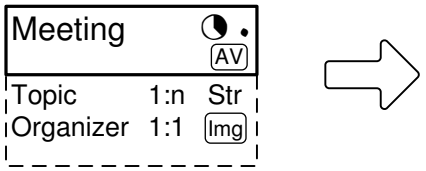
**Figure 3.12:** A time-varying multimedia element.

is time-varying). For example, a regular photo of the merchandize would be used for "active" commodities, a washed-out picture for "suspended" (i.e. temporary unavailable) commodities, and a washed-out picture with an inscription stating that the merchandise is not sold anymore for "disabled" goods (see fig. 3.12).

Nevertheless, from the standpoint of concurrent multimedia and temporal semantics, it is important to stress the fundamental difference between time-varying multimedia data and static timeline-based multimedia data like video or audio. Let's consider an example of a multimedia object type $O_1$ with a time-varying image representation `Image(t)` vs. a multimedia object type $O_2$ of type `Video`. Indeed, a video file is often perceived as a sequence of images (frames) that supersede themselves at a certain frequency (frame rate), e.g. 25 fps PAL, or 30 fps NTSC. The duration of a video clip can then be calculated as the total number of frames divided by the frame rate. Despite the seeming similarity of video data to a time-varying image data, the two are different, and the duration of a video should not be confused with the lifecycle of an object having an image representation.

Firstly, at each particular point in time, multimedia representation of an object of type $O_1$ is characterized by a still image that provides the appropriate multimedia depiction of the object at the given moment. At the same time, an object of type $O_2$ is always characterized by the selfsame video clip no matter the point in time. Thus, at any given point in time $T$ the multimedia representation of the object $O_1$ is of multimedia type `Image`, while the multimedia representation of the object $O_2$ is always of type `Video`.

Secondly, the lifecycle of an object conceptually conveys the temporal nature of the real-world object itself, while a video timeline relates to the logical organization of a video file or stream representing the real-world object. Although the two might sometimes match either fully (e.g. the entirety of a meeting $M$ is being filmed providing its multimedia representation, whose duration corresponds to the lifecycle of $M$), or partially (e.g. the multimedia representation of a two-hour meeting $M$ is a five-minute video resume of $M$'s milestones), nevertheless they are generally different (e.g. the multimedia representation of $M$ is a promotional video clip showing the conference venue filmed a year ago).




```
Object Meeting {
  Lifecycle Interval
  Geometry Point
  Multimedia AudioVideo

  Topic [1,n] String
  Organizer Image
}
```

**Figure 3.13:** Multimedia semantics in a MADS schema and an object type definition.

Besides time-varying multimedia data, the orthogonality of MADS modeling dimensions also allows for space-dependent multimedia data. Similarly to time-varying multimedia, the space-varying multimedia is also not static. Although the data type is fixed, the multimedia value changes throughout object's geometry. For example, a spatio-multimedial object type `Meeting` with a multimedia extent of type `Video` and a spatial extent of type `SimpleSurfaceBag` (see fig. 3.3) can have its multimedia representation vary depending on the meeting geometry. With the meeting venue being split between several rooms, where each room is filmed with its own video equipment, the multimedia representation of a meeting varies according to its geometry (i.e. according to a particular room of the meeting venue).

Needless to say, the multimedia information could also depend on both space and time together. An example of space-and-time-varying multimedia is an object type `Meeting`, whose multimedia representation of type `Image` depends on meeting's lifecycle and geometry. With a different photo camera shooting each of the meeting rooms every 3 minutes, the multimedia representation of a meeting varies in time (the picture changes every 3 minutes), but also in space (different pictures for different meeting rooms).

Although this example illustrates quite well the case of space-and-time-varying multimedia, we believe that due to its relative complexity the area of its applicability is rather limited.

The fig. 3.13 illustrates the use of multimedia data types in MADS. The object type `Meeting`, as well as one of its attributes `Meeting.Organizer` are specified as multimedia (`AudioVideo` and `Image`, respectively), which on the object type definition level means having an attribute `multimedia` of type `AudioVideo`, and an attribute `Organizer` of type `Image`. Note as well, that the object type `Meeting` combines spatial, temporal, and multimedia characteristics.

### 3.3.3 Abstract Set-Based Definition of Multimedia Datatypes

As mentioned in sect. 3.3.1, multimedia datatypes from the hierarchy in fig. 3.11 are defined abstractly in the sense that we do not privilege any particular perception of the multimedia data and of its underlying structure. In this subsection we define the conceptual-level understanding of the essence of multimedia data pertaining to the multimedia datatypes introduced in sect. 3.3.1.

As a matter of fact, there exist a lot of various types of multimedia data, which are often heterogeneous in their nature and have little things in common (e.g. image data vs. sound). This heterogeneity relates to the fact that perceiving multimedia data involves a multitude of human senses, e.g. sight, hearing, etc., as well as a substantial participation of the brain. For example, although a photo we see on a computer screen and a text we see in a text editor window are both perceived with our eyes, it is our brain that makes the difference between the two by recognizing text characters and interpreting their meaning. It is also the brain that plays the leading role in recognizing the objects and events we see on a photo. Nevertheless, the various ways of representing multimedia data, which exist nowadays, are far from conveying the knowledge that human brain is able to extract, and are usually mimicking the low-level hardware-oriented view of raw media files (e.g. bitmap pixel representation of digital images caused by pixel matrix organization of computer screens, etc.). Thus, a regular pixel-based representation of an image that is used, for example, in BMP, GIF, TIFF, and other formats, fails at conveying the geometry of objects it depicts, while vector-based image formats like EMF can provide the geometrical description of objects, however cannot convey semantic description of objects they represent. Even with such classical type of media data as text, still a lot of problems in the field of natural language processing are left unsolved. In this context, multimedia data differs drastically from, e.g. spatial data, where geometrical shapes can be fully described either discretely by sets of points in a discrete space, or continuously using geometrical equations.

As mentioned in sect. 2.4.2, besides conveying a purely sensorial information such as vision or sound, multimedia data is also manifested by the semantic information it conveys, which is generally provided in the form of annotations (controlled or free-form) associated with the multimedia data. In our opinion, a multimedia data element is really characterized by both its sensorial as well as semantic parts. Unlike the sensorial information, which, as we have just described above, is difficult to represent in a unified way due to the heterogeneity of various types of multimedia and its digitalization techniques, on the contrary, the semantic annotation-based information is distinguished by its independence from the particular type of multimedia data it describes. Indeed, semantic annotations of e.g. two multimedia elements, of types `AudioVideo` and `Image` respectively, can all be provided in an Annotea format (see sect. 2.3) no matter the particular datatypes

of the multimedia elements behind the annotations. Hence, the semantic aspect of multimedia information, which is conveyed by a collection of annotations associated with a multimedia element, represents one of the candidate interpretations of the essence of multimedia information and can serve as a basis for the conceptual-level interpretation of multimedia information.

Although semantic annotations can act as a universal substance of the multimedia information at the conceptual level, they cannot fully convey the sensorial side of the multimedia data. No matter how rich the annotations of the digital photo are, the annotations alone will not suffice to convey the information that can be obtained by actually looking at the photo. From this standpoint, it is important to still be able to allow for alternative interpretations of multimedia information that pertain to its sensorial aspect, and also possibly to allow for complex interpretations that mix several different interpretations pertaining to either sensorial or semantic aspects of multimedia.

Taking into consideration the above-given argumentation, we have decided not to limit ourselves by any particular interpretation of the structure of multimedia data, but rather to adapt a generic formal abstract interpretation based on the formalisms of the set theory.

Set theory is one of the axiomatic foundations for mathematics, allowing abstract objects to be constructed formally from the undefined terms of "set" and "set membership" [HJ99]. Following this approach, a multimedia element is conceptually defined as an abstract set, whose elements and membership depend on a particular implementation model that is chosen by the user. For example, an image could be defined as a set of pixels forming the image bitmap, or as a set of semantic annotations (e.g. a set of Annotea RDF triplets) characterizing the image, or even as a combination of several models, e.g. a combination of both image bitmap pixels and annotation objects of the image.

Having multimedia elements defined as abstract sets allows in particular defining abstract operations on such sets. For instance, we could define a method function `magnitude()` within the multimedia datatype `Image`, which calculates the magnitude of an image as the number of elements in its set-based representation. The particular implementation of this method function will depend on the particular image representation approach that we choose. Thus, if images are regarded as sets of triplets $\{(x_i, y_i, c_i)\}$ describing the coordinates and the color of image pixels, then `magnitude()` returns the number of pixels in the image. If on the other hand we opt for a vector graphics representation, with images regarded as sets of geometrical primitives used in the image $\{(g_i)\}$, then `magnitude()` returns the number of all geometrical objects in the image. Finally, when represented as a set of associated sematic annotations $\{(a_i)\}$, the method function `magnitude()` of the datatype `Image` returns the number of all semantic annotations $a_i$ associated with the image.

It should also be noted that some methods may only make sense with some particular implementation models, but not the others. For example, a method function `max_area()`, which calculates the area of the largest geometrical figure within an image, would only make sense for a vector-based representation of an image, and would be inappropriate for pixel-based images. As for the annotation-based image representation, the `max_area()` function could also return the area value for the largest geometrical figure, provided that each figure within the image is characterized by a collection of annotations among which is an annotation on the figure measurements. Thus, the list of specific methods provided by abstract multimedia data types is to a large extent dependent on the particular implementation model that is chosen by the user.

In sect. 3.4.2 and sect. 3.4.4 we will provide several different examples of implementing the abstract set-based representation of multimedia data.

### 3.3.4 Section Summary

In this section we have established a basis of the new multimedia extension of MADS conceptual model by introducing a set of novel multimedia datatypes.

The datatypes that have been proposed correspond to various sensorial and perceptional varieties of multimedia data, and allow to take into account the peculiarities of each specific kind of multimedia. In a MADS schema, just like the datatypes of other data varieties, the multimedia datatypes can be used to characterize attributes, entity types, and relationship types. The importance of organizing the newly introduced datatypes in a complex hierarchical structure according to a certain criteria has also been stressed.

A generic conceptual-level definition of the essence of multimedia datatypes in MADS has been provided using abstract concepts of the set theory. The abstract definition allows us to deal with potentially very different representations of the same datatypes, like, for example, pixel-based or annotation-based representations of pictures.

The abstract set-based definition of multimedia data finds its further application in the next section, where we describe multimedia representational relationships, which provide a powerful mechanism of introducing multimedia-related constraints into a MADS multimedia schema.

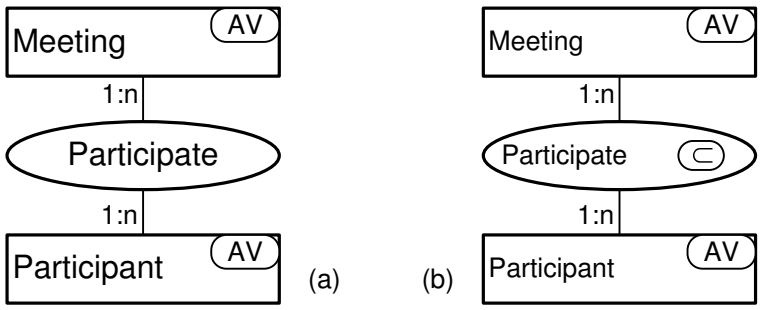



**Figure 3.14:** Representational relationships in MADS.

## 3.4 Multimedia Representational Relationships

As mentioned in the sect. 3.3.2, relationship types in MADS can be specified as multimedia relationship types by being associated with one of the abstract multimedia datatypes defined in sect. 3.3.1. In addition to that, we introduce in this section a special kind of relationships called *multimedia representational relationships*, in which multimedia object types can participate. Enriching a binary relationship with representational semantics allows to impose additional multimedia-related constraints on `multimedia` attributes of instances linked by the relationship.

### 3.4.1 Simple Multimedia Representational Relationships

In the example illustrated in fig. 3.14a object types `Meeting` and `Participant` are linked by a relationship type `Participate`. Both `Meeting` and `Participant` are said to be of multimedia type `AudioVideo` (`AV`). Suppose that we would like to constrain the relationship `Participate` by imposing that only then does it hold when the assumed meeting participants can be seen and/or heard on the multimedia representation of the corresponding meeting. To enforce this condition, the relationship type `Participate` is further constrained by being assigned as *multimedia inclusion* type (fig. 3.14b), meaning that according to the schema in fig. 3.14b any participant, who is said to participate in a meeting, must additionally have his multimedia representation included in the multimedia representation of the corresponding meeting. This would in particular signify (taking into account that both `Meeting` and `Participant` are of `AudioVideo` (`AV`) multimedia type) that the audio-visual representation of the `Participant`, which is contained in his `multimedia` attribute, is included in the audio-visual representation of the corresponding `Meeting`, contained in the Meeting's `multimedia` attribute.

We define the following four generic types of representational relationships:

1) *multimedia inclusion* ($\subset$): multimedia representation of one linked instance is semantically included into multimedia representation of the other linked instance (see e.g. fig. 3.14b);

2) *multimedia intersection* ($\cap$): multimedia representations of two linked instances share some common semantics, however neither of the two completely includes the other one;

3) *multimedia equality* ($=$): multimedia representations of two linked instances are semantically equal, meaning that multimedia representation of neither linked instances semantically provides more information than the other one.

4) *multimedia disjointness* ($\bar{\cap}$): multimedia representations of two linked instances are semantically disjoint, i.e. share no information in common.

It is important to emphasize that associating relationship types in MADS with any of the representational relationship types described above only imposes additional constraints of merely conceptual nature, and does not necessarily imply similar restrictions on physical multimedia data sources (e.g. files, streams, etc.) behind the multimedia instances linked by the relationship in question. For example, turning back to fig. 3.14b, the fact of the relationship type `Participate` being of the representational type *multimedia inclusion* does not necessarily imply that e.g. MPEG files containing video portrayal of meetings should be physically composed of MPEG files with video representations of the meeting participants.

The above-given descriptions of the four multimedia representational relationship types only provide their general conceptual meaning. The formal definitions of the four relationship types are given using the formalisms of the set theory, which appeal to the abstract set-based representations of multimedia elements participating in the relationship, and are based on set membership.

If $A$ and $B$ are homogeneous multimedia elements, then:

$$\begin{array}{lll} A \subset B & \text{iff} & \forall x \in A : x \in B \\ A \cap B & \text{iff} & \exists x : (x \in A) \wedge (x \in B) \\ A \bar{\cap} B & \text{iff} & \neg \exists x : (x \in A) \wedge (x \in B) \\ A = B & \text{iff} & (A \subset B) \wedge (B \subset A) \end{array} \tag{3.1}$$

Although the definitions above are formal, their exact interpretation remains abstract due to the abstract set-based definitions of multimedia elements $A$ and $B$. The particular meaning of these relationship types depends on the particular model that is chosen to represent the abstract sets. For example, if at the conceptual level the multimedia image elements $A_1$ and $B_1$ were represented by their bitmap interpretations of sets of pixels, then a multimedia equality $A_1 = B_1$ would signify equality of images $A_1$ and $B_1$ on the pixel bitmap level. On the other hand, if multimedia elements $A_2$ and $B_2$ were conceptually represented by sets of their associated semantic annotations, then a multimedia intersection $A_2 \cap B_2$ would mean the existence of common annotations between $A_2$ and $B_2$ (e.g. two digital

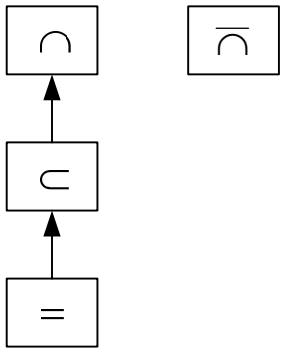



**Figure 3.15:** Hierarchy of basic multimedia representational relationships.

photos having some common tags).

It is important to note that the four multimedia representational relationships defined in expression 1.1 are not completely orthogonal. Indeed, it is easy to see that according to the set-based definitions of the representational relationships, the following fact can be deduced:

$$A{=}B{\Rightarrow}A{\subset}B{\Rightarrow}A{\cap}B,$$

i.e. the equality relationship implies inclusion relationship, which in its turn implies intersection. In fact, only the multimedia disjointness is completely orthogonal to the other three representational relationship types. This interdependency of multimedia representational relationships is depicted in a hierarchy in fig. 3.15.

Similarly to the data type hierarchies in MADS, the hierarchy of representational relationship types allows users and schema designers to choose the desired level of compromise between generality and specificity by using either the more specific representational relationships to keep the model more precise, or else the less specific (i.e. more general) ones to relax the multimedia constraints and allow taking into account a broader range of situations. However, it is important to note that unlike data type hierarchies, the non-leaf representational relationships in the hierarchy in fig. 3.15 (i.e. $\cap$ and $\subset$) do not describe abstract relationships which cannot be valued on their own, like it is, for instance, the case, with multimedia super types in the hierarchy in fig. 3.11. In this regard, the hierarchical organization of multimedia representational relationships simply conveys the specialization-generalization semantics between various representational relationships, and not the abstraction-instantiation semantics like in data type hierarchies.


**Figure 3.16:** An image fragment of a city plan.


**Figure 3.17:** A fragment of a city plan conceptual schema.

### 3.4.2 Examples of Simple Multimedia Representational Relationships

In this subsection we provide several examples of multimedia-enhanced applications and the way that multimedia data and multimedia representational relationships are defined using the formalism of the set theory.

#### Example #1

Let's consider a sample application, which models a town plan represented in fig. 3.16.

A fragment of the application conceptual schema in fig. 3.17 shows two spatial entity types `Building` and `Block` and a relationship type `IsLocatedIn` reinforced with a topological constraint of type `Within` expressing geographical affiliation of a building to a city block.

To take into consideration the multimedia depiction of the concepts pertinent to the universe of discourse, we want to provide each instance of a building and of a city block with a visual representation derived from the town plan on fig. 3.16. For that we enhance the conceptual schema in fig. 3.17 with multimedia semantics by specifying that both entity types `Building` and `Block` are characterized by mul-

timedia representations of type `Image`. Besides the topological constraint `Within`, the relationship type `IsLocatedIn` further introduces an additional constraint of multimedia nature imposing that the multimedia representation of a building is included in the multimedia representation of its corresponding city block. The multimedia inclusion relationship between a building and a block means that according to the town plan on fig. 3.16 the image of a building can be obtained from the image of its corresponding block by cropping[4].

Note that since according to the hierarchy in fig. 3.15, the *multimedia equality* representational relationship is a subtype of *multimedia inclusion*, then the buildings that occupy an entire block would be among those that comply with the schema in fig. 3.17. However, if only that type of builings was to be considered for the `IsLocatedIn` relationships type, then the *multimedia equality* representational relationship must have been used instead of the *multimedia inclusion*.

Using the set theory approach described in sect. 3.3.3 we view image representations of map images (incl. buildings and blocks) as bitmaps, i.e. sets of all pixels of the image. Clearly, the image of the entire city map shown in fig. 3.16 corresponds to the universal set ($U$). Supposing that $U$ is $X_U$ pixels wide by $Y_U$ pixels high, we can define an image of a building or a city block as a set of coordinate pairs on the image bitmap, namely:

$\{(x,y) : x{\in}[X_0;X_N]; y{\in}[Y_0;Y_N]\}$, where $X_0{<}X_N{\in}[0;X_U]$, and $Y_0{<}Y_N{\in}[0;Y_U]$.

Suppose 2 images $A$ and $B$ defined as follows:
$A{=}\{(x,y) : x{\in}[X_{A_0};X_{A_N}]; y{\in}[Y_{A_0};Y_{A_N}]\}, X_{A_0}{<}X_{A_N}{\in}[0;X_U], Y_{A_0}{<}Y_{A_N}{\in}[0;Y_U]$
$B{=}\{(x,y) : x{\in}[X_{B_0};X_{B_N}]; y{\in}[Y_{B_0};Y_{B_N}]\}, X_{B_0}{<}X_{B_N}{\in}[0;X_U], Y_{B_0}{<}Y_{B_N}{\in}[0;Y_U]$
Then the 4 multimedia relationship types between $A$ and $B$ described by expression 1.1 are formally defined as follows:

$$\begin{aligned} A{\subset}B &\quad \text{iff} \quad \forall(x,y){\in}A : (x,y){\in}B \\ A{\cap}B &\quad \text{iff} \quad \exists(x,y) : (x,y){\in}A \wedge (x,y){\in}B \\ A{\bar{\cap}}B &\quad \text{iff} \quad \neg\exists(x,y) : (x,y){\in}A \wedge (x,y){\in}B \\ A{=}B &\quad \text{iff} \quad A{\subset}B \wedge B{\subset}A \end{aligned}$$

If $A$ is an image of a building and $B$ is an image of a block, then the 4 multimedia relationships above can be semantically described as follows:

$A{\subset}B$: The image of a building is completely inside the image of a block (i.e. the image of a building can be obtained by cropping the image of a block).

$A{\cap}B$: A part of the image of the building is inside the image of a block (i.e. a part of the image of a building can be obtained by cropping the image of a block, and vice versa).

[4]We refer here to "cropping" in the sense of one of the most common operations provided by the majority of image-editing software like GIMP, Adobe Photoshop, etc.

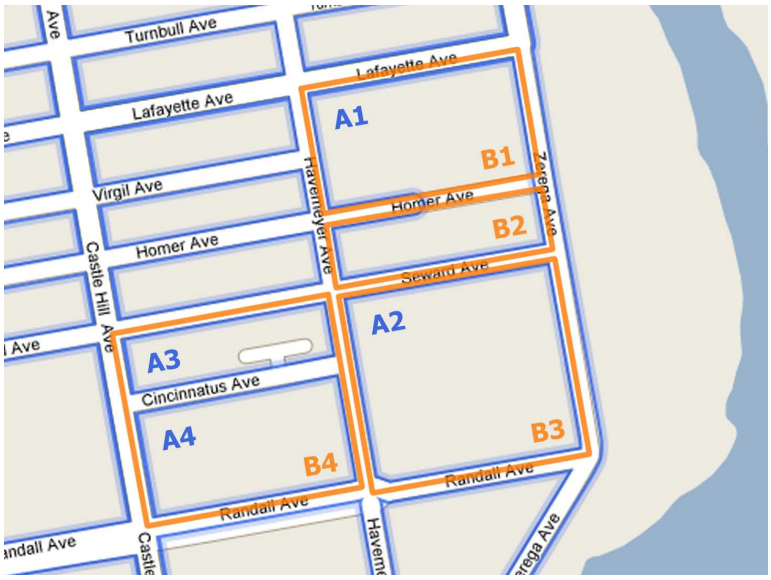



**Figure 3.18:** Buildings and blocks on an image fragment of a city plan.


**Figure 3.19:** Human resources database.

$A \bar{\cap} B$: The images of the building and the block have no common parts (i.e. no part of the image of a building can be obtained by cropping the image of a block, and vice versa).

$A{=}B$: The images of the building and the block are the same.

In fig. 3.18 rectangles A1, A2, A3, and A4 represent images of buildings within the global town plan U (fig. 3.16), while B1, B2, B3, and B4 represent images of four town blocks within U.

It is then easy to see that the following multimedia relationships hold between $A_i$ and $B_j$:

| | B1 | B2 | B3 | B4 |
|---|---|---|---|---|
| A1 | $\cap$ | $\cap$ | $\bar{\cap}$ | $\bar{\cap}$ |
| A2 | $\bar{\cap}$ | $\bar{\cap}$ | $=$ | $\bar{\cap}$ |
| A3 | $\bar{\cap}$ | $\bar{\cap}$ | $\bar{\cap}$ | $\subset$ |
| A4 | $\bar{\cap}$ | $\bar{\cap}$ | $\bar{\cap}$ | $\subset$ |

### Example #2

Let's consider a fragment of a HR database conceptual schema on fig. 3.19, where each company employee is attached to one or several departments.

The application is also multimedia-enhanced, with each employee record comprising a photo, and each department also characterized by a group photo of all its employees. To make sure that a new group picture is taken each time a new employee is joining a department, we enforce the relationship type `WorksAt` with an additional constraint of multimedia nature stating that every department employee must be seen on the department group photo.

Using the set theory approach described in sect. 3.3.3, at the conceptual level we view department and employee pictures as sets of semantic annotations of the people that these pictures represent. Thus, an individual employee picture corresponds to a singleton set (there's only one person annotated), while a group picture is described by a set with several elements, e.g.:

$$John\_photo = \{John\},$$
$$HR\_Dept\_photo = \{John, Paul, Mary, Chris, Mike\}^5.$$

With this approach in place, the 4 multimedia relationships between multimedia entities $A$ and $B$ are defined as follows:

$$\begin{array}{lll} A \subset B & \text{iff} & \forall x \in A : x \in B \\ A \cap B & \text{iff} & \exists x : x \in A \wedge x \in B \\ A \bar{\cap} B & \text{iff} & \neg \exists x : x \in A \wedge x \in B \\ A = B & \text{iff} & A \subset B \wedge B \subset A \end{array}$$

where $x$ is a single annotation of a person seen on the photo. For example, $x = John$ and $x \in A$, meaning that John can be seen on the photo $A$.

If $A$ is a photo of a company employee and $B1$, $B2$ are group photos of two company departments, then multimedia relationships between $A$ and $Bi$ can be semantically described as follows:

$A \subset B1$: the employee from the personal photo $A$ can be seen on the group photo $B1$ (i.e., the employee $A$ is apparently attached to the department $B1$).

$B1 \subset B2$: all the employees from the group photo $B1$ are also seen on the group photo $B2$ (i.e., the department $B1$ is apparently a sub-department of department $B2$).

$B1 \cap B2$: there are employees that are seen on both group photos $B1$ and $B2$ (i.e. there are employees that are apparently attached to both departments $B1$ and $B2$).

[5]For the sake of simplicity we use first names to identify employees in this example. Obviously, to avoid possible semantic conflicts between employees with the same name, some globally unique identifier like SSID could be used instead.


**Figure 3.20:** Surveillance database.

$B1 \bar{\cap} B2$: there are no employees seen at the same time at group photos $B1$ and $B2$ (i.e. there are no employees in the company that are simultaneously attached to both departments $B1$ and $B2$).

$B1 = B2$: employees seen on both group photos $B1$ and $B2$ are the same people (i.e. all the employees attached to the department $B1$ are also attached to the department $B2$, and vice versa).

It is interesting to note, that although not directly recurring to low-level features of image data like color distribution, patterns, pixel bitmaps, etc., the vision of multimedia data as a set of annotated semantic objects represented in the media recordings as presented in this example also fits into the general set-theory-based approach described in sect. 3.3.3. Moreover, from the image content analysis point of view, this annotation-based representation of multimedia data allows, for example, matching two photos of the same person despite different clothing, haircut, or even age of the person on the photos. The latter still remains very hard to achieve with automatic low-level image content analysis approaches.

### Example #3

Let's consider a video surveillance application, where a video camera is used to record a patrolled area twenty-four hours a day. The surveillance rota is organized on a daily basis, with several shifts every day (see a fragment of the conceptual schema on fig. 3.20).

The relationship type `ConsistsOf` between entity types `DailyWatch` and `WatchShift` is enforced with a synchronization constraint of type `Within` to specify that every watch shift is scheduled within its corresponding daily watch rota. From the multimedia point of view, each daily watch as well as each particular shift are characterized by a corresponding video footage from the surveillance camera. To provide an additional level of security and to control that no shift video footage is corrupted, we impose that shift videos must be included in their corresponding daily footages (note the multimedia inclusion semantics of the relationship type `ConsistsOf`).

Using the set theory approach described in sect. 3.3.3 we view each video footage as a continuous sequence of frames with their temporal binding:

$$\{(F_i, t_i) | t_i \in [t_0, t_N]\},$$

where $F_i$ is a video frame, and $t_i$ is a global timestamp of the frame $F_i$.

The four multimedia relationship types between video footages $A$ and $B$ can then be described as follows:

$$\begin{aligned} A{\subset}B &\text{ iff } \forall(F,t){\in}A : (F,t){\in}B \\ A{\cap}B &\text{ iff } \exists(F,t) : (F,t){\in}A \wedge (F,t){\in}B \\ A\bar{\cap}B &\text{ iff } \neg\exists(F,t) : (F,t){\in}A \wedge (F,t){\in}B \\ A{=}B &\text{ iff } A{\subset}B \wedge B{\subset}A \end{aligned}$$

If $A$ and $B$ are video footages, then the 4 multimedia relationship types above can be semantically described as follows:

$A{\subset}B$: the video clip A is represented in the video clip B (i.e. the footage $A$ can be obtained from the footage $B$ by cutting).

$A{\cap}B$: a part of the video clip $A$ is represented in the video clip $B$ (i.e. a part of the footage $A$ can be obtained from the footage $B$ by cutting, and vice versa).

$A\bar{\cap}B$: video clips $A$ and $B$ represent nothing in common (i.e. no part of the footage $A$ can be obtained by cutting the footage $B$, and vice versa).

$A{=}B$: video clips $A$ and $B$ are the same.

### Example #4

In example 3 each video clip is conceptually represented as a set of pairs $(F_i, t_i)$, where $F_i$ is a video frame, and $t_i$ is a global timestamp of the frame $F_i$. In this setting, a multimedia inclusion relationship between videos $A$ and $B$ yields:

$$A{\subset}B \text{ iff } \forall(F,t){\in}A : (F,t){\in}B,$$

i.e. all the frame-timestamp pairs of video $A$ also belong to video $B$. This representation, however, would not allow, for example, noting multimedia inclusion of video $V1$ in video $V2$, where $V1$ and $V2$ are timeline-synchronized and frames of $V1$ are obtained from corresponding frames of $V2$ by image cropping (see fig. 3.21).

The generic set theory approach described in sect. 3.3.3 also allows taking into account situations like the one shown in fig. 3.21 by using the notion of Cartesian product.

According to the set theory, if $A$ and $B$ are sets, then the Cartesian product of $A$ and $B$ is defined as:

$$A{\times}B = \{(a,b)|a{\in}A \wedge b{\in}B\}.$$

That is, $A{\times}B$ is the set of all ordered pairs whose first coordinate is an element of $A$ and whose second coordinate is an element of $B$. Although Cartesian product was initially defined by René Descartes to provide the formulation of analytic geometry (e.g. definition of Euclidean space as a product: $R^3{=}R{\times}R{\times}R$), the

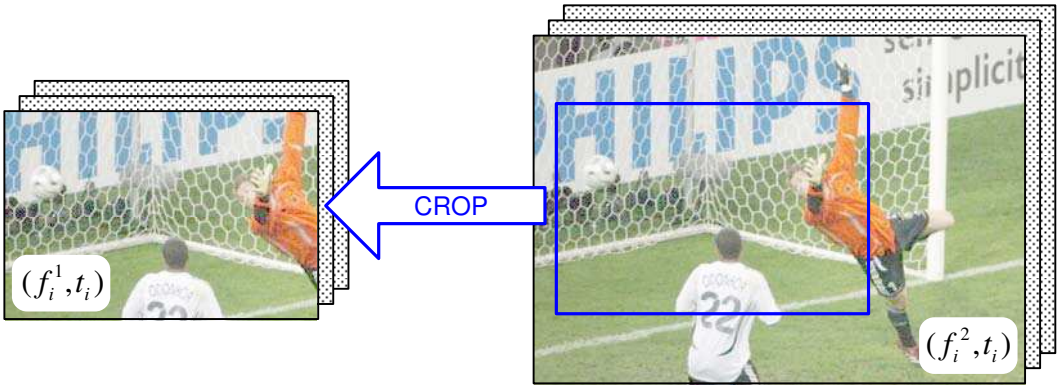



**Figure 3.21:** Video cropping example.

notion of a cartesian product $A \times B$ is generally applicable to any abstract sets $A$ and $B$, with multimedia data being no exception.

To define a video clip using the Cartesian product approach, let's consider a universal set of video clips $V$, such that:

$$V = F \times T,$$

where $F$ is a universal set of all video frames, and $T$ is a universal set of all video timelines. Then a particular video clip $A$ can be defined as follows:

$$A = \{(f_i, t_i) | f_i \in F, t_i \in T\},$$

where $f_i$ is a video frame, and $t_i$ is a global timestamp of the frame $f_i$. With this approach the definition of four multimedia relationship types stays very much the same as in the example 3. However, using Cartesian products introduces orthogonality principle in definition of multimedia elements. In the example of $V = F \times T$ we can work with the timeline component ($T$) of video data independently of the video frame component $F$. Moreover, having the universal set of video frames $F$ defined abstractly allows remaining generic with regard to a particular vision of the conceptual essence of video frames in $F$. From a software engineering point of view, the Cartesian-based approach allows modularizing the set-based definition of a multimedia (video) elements, and any particular representation of image frame space $F$ whatsoever would comply with the general definition of $V = F \times T$.

For example, to fulfill the requirements of an application like the one shown in fig. 3.21, the video frames can be regarded in the way similar to that of the example 1 above, namely:

$$F = X \times Y \times C,$$

i.e. the set $F$ is itself represented as a Cartesian product, where $X$ and $Y$ describe frame pixel coordinates, and $C$ describes the pixel color space (e.g. RGB, HSB, CMYK, etc.).

The universal video set $V$ can then be redefined as:

$$V = F \times T = X \times Y \times C \times T,$$

and a particular video clip $A$ can be defined as follows:

$$A = \{(x_i, y_i, c_i, t_i) | x_i \in X, y_i \in Y, c_i \in C, t_i \in T\}.$$

If $A$ and $B$ are video clips, then the 4 multimedia relationship types are defined as follows:

$$\begin{array}{lll} A \subset B & \text{iff} & \forall (x,y,c,t) \in A : (x,y,c,t) \in B \\ A \cap B & \text{iff} & \exists (x,y,c,t) : (x,y,c,t) \in A \wedge (x,y,c,t) \in B \\ A \bar{\cap} B & \text{iff} & \neg \exists (x,y,c,t) : (x,y,c,t) \in A \wedge (x,y,c,t) \in B \\ A = B & \text{iff} & A \subset B \wedge B \subset A \end{array}$$

Contrary to multimedia relationships in example 3, the multimedia inclusion, intersection, and disjointness relationship types above can be semantically defined as:

$A \subset B$: the footage $A$ can be obtained from the footage $B$ by cutting in both timeline and video frame dimensions.

$A \cap B$: a part of the footage $A$ can be obtained from the footage $B$ by cutting in both timeline and video frame dimensions, and vice versa.

$A \bar{\cap} B$: no part of the footage $A$ can be obtained by cutting the footage $B$ in timeline or video frame dimension, or vice versa.

An alternative view of the space of video frames $F$ as a set of semantic objects (e.g. people) seen on these video frames and provided via annotations (see example 2 above) would yield the following representation of $V$:

$$V = F \times T = P \times T,$$

where $P$ is the set of all people seen on the video recordings from $V$.

This approach allows, in particular, matching video footages that are timeline-synchronized (e.g. using globally universal time-stamping), however, can hardly be matched on the frame level using low-level image feature analysis. An example of such scenario is a news coverage of an event with several available video footages taken by video cameras of different news agencies. Due to different video recording equipment being positioned and oriented differently, the produced recordings can be very hard to match automatically on the frame level (see fig. 3.22), however, using the representation of $V = P \times T$, we can note the video footage $V_1$ on fig. 3.22 being included into the footage $V_2$ ($V_1 \subseteq V_2$), since both $V_1$ and $V_2$ are timeline-synchronized, and the people we see in $V_1$ are also shown in $V_2$.

### 3.4.3 Complex Multimedia Representational Relationships

In the examples 1 through 4 in sect. 3.4.2 we have demonstrated the applicability of the general abstract set-based conceptual understanding of the content of

**Figure 3.22:** Timeline-synchronized video footages $V_1$ (left) and $V_2$ (right) depicting the same event.

multimedia data (see sect. 3.3.3) to different particular perceptions of multimedia. Nevertheless, it is clear that the four general binary relationships, which compare the entire set-based representations of multimedia elements taken in the lump, can turn out to be too coarse for a number of applications that would require a finer level of detail. Thus, for example, one might be interested in comparing only the video components of two complex type multimedia elements, ignoring the media components of other types (e.g. audio, text, etc.). Another requirement could consist in distinguishing not simply intersections of image bitmaps, but rather intersections of the upper-right corner of one image with the bottom-left corner of the other image, etc. To meet these possible requirements and to optionally provide a way of refining the four general multimedia relationship types described in sect. 3.3.1, we propose to provide a way of refining the set-based representations of multimedia elements themselves.

Our approach to refining multimedia relationships consists in providing a mechanism for partitioning the set-based representations of multimedia elements participating in the relationships into subsets according to a certain proposition (criterion). Generally, for any set $A$, partitioning of $A$ into $N$ subsets is provided, such that:

$$A = \bigcup_{i=1}^{N} A_i, \text{ where } \forall i, j \in [1, N], i \neq j : A_i \cap A_j = \emptyset.$$

Having sets $A$ and $B$ partitioned into subsets in this manner allows comparing the subsets of $A$ and $B$ pairwise, namely:

$$A_i R_k B_j,$$

where $R_k$ is one of the four basic relationships defined in sect. 3.4.1. For example, having $A$ and $B$ partitioned each in two subsets ($A_1$, $A_2$, $B_1$, $B_2$) allows comparing 4 different pairs of subsets of $A$ and $B$:

$$A_1 R_{k1} B_1,\ A_1 R_{k2} B_2,\ A_2 x R_x k 3 B_1, \text{ and } A_2 R_{k4} B_2 x.$$

Simultaneously considering combinations of various subsets of $A$ and $B$ allows us introducing additional relationships of the following form between $A$ and $B$:

$$AR_iB : (A_1R_{k1}B_1)\wedge(A_1R_{k2}B_2)\wedge(A_2R_{k3}B_1)\wedge(A_2R_{k4}B_2).$$

Having introduced the principle of obtaining additional multimedia representational relationships based on set partitioning, let's count the number of various binary representational relationship types that can be obtained in this way.

Seemingly, since partitioning multimedia elements $A$ and $B$ into $K$ subsets each yields $K^2$ possible combinations of pairs $A_iR_XB_j$, where $R_X$ is one of the four basic representational relationships, then a total of $4^{K^2}$ various binary relationships could be possible. However, due to the interdependence of basic representational relationships as described in fig. 3.15, not all of the $4^{K^2}$ potential combinations are valid.

Indeed, it is easy to notice that if a subset $B_j$ of $B$ includes a subset $A_i$ of $A$ then no other subset of $B$ can intersect with $A_i$, i.e.:

$$A_i{\subset}B_j \Rightarrow \neg\exists l{\neq}j : A_i{\cap}B_l,$$

which is easy to prove by contradiction.

Proof:

Let's assume that there exist two different subsets of $B$, such that $A_i$ is included in one of them and intersects with the other one, i.e.:

$$\exists p, q : (A_i{\subset}B_p) \wedge (A_i{\cap}B_q).$$

It follows then that:

$$\exists x{\in}A_i : (x{\in}B_p) \wedge (x{\in}B_q).$$

Hence:

$$B_p{\cap}B_q{\neq}\emptyset,.$$

which contradicts by definition the set separation principle stating that subsets of a separation do not intersect.

□

Furthermore, taking into account that according to the hierarchy in fig.3.15 the multimedia inclusion is a subtype of multimedia intersection, and multimedia equality is a subtype of multimedia inclusion representational relationship, i.e.:

$$A{=}B \Rightarrow A{\subset}B \Rightarrow A{\cap}B,$$

it follows that every subset $B_j$ of $B$ can include or be equal to at most one subset $A_i$ of $A$, i.e.:

$$\begin{aligned}
(A_i{\subset}B_j) \Rightarrow \quad & \nexists l{\neq}j : (A_i{\cap}B_l) \vee (A_i{\subset}B_l) \vee (A_i{=}B_l), and \\
(A_i{=}B_j) \Rightarrow \quad & \nexists l{\neq}j, \nexists m{\neq}i : \\
& (A_i{\cap}B_l){\vee}(A_i{\subset}B_l){\vee}(A_i{=}B_l){\vee}(A_m{\cap}B_j){\vee}(A_m{\subset}B_j){\vee}(A_m{=}B_j).
\end{aligned}$$

In other words, if $A_i \subset B_j$, then for the rest $(K-1)$ pairs $A_i R^? B_f$, where $f \neq i$, the representational relationship $R^?$ can only be a multimedia disjointness ($\bar{\cap}$), i.e.:

$$\begin{aligned}(A_i \subset B_j) \Rightarrow \quad & \forall l \neq j : (A_i \bar{\cap} B_l), and \\ (A_i = B_j) \Rightarrow \quad & \forall l \neq j, \forall m \neq i : (A_i \bar{\cap} B_l) \wedge (A_m \bar{\cap} B_j).\end{aligned}$$

Taking into account the above argumentation, we have derived an iterative formula for calculating the maximum number of possible set-partitioning-based multimedia representational relationships depending on the number $K$ of partitions, which we give here without proof:

$$N(K) = (2^K + K)^K + N(K-1) \cdot K^2 - (K!)^2 \cdot \sum_{i=2}^{K} \frac{N(K-i)}{i! \cdot ((K-i)!)^2} \tag{3.2}$$

where $N(0)=1$ and $N(1)=4$.

Table 3.3 lists the values of $N(K)$ for several different values of $K$:

| K | $N(K)$ |
|---|---|
| 2 | 50 |
| 3 | 1703 |
| 4 | 183240 |
| 5 | 73551837 |
| 10 | $\approx 1.4 \cdot 10^{30}$ |

**Table 3.3:** Maximum possible number of complex representational relationships.

As one can see from table 3.3, in the case of a simplest nontrivial partitioning (i.e. $K=2$) we obtain 50 possible representational relationship types, with this number growing to 1703 for $K=3$, and continuing on growing exponentially for more complex separations.

Obviously, defining such a big number of multimedia relationships is hardly practicable, and is in general rarely required. Indeed, the set-partitioning approach described here merely provides a way of detailing the 4 basic multimedia relationship types described in sect. 3.4.1. However, no obligation of using all the possible $N(K)$ combinations is imposed. It is up to the user to arbitrary choose a subset of relationships that he/she is interested in, i.e. the relationships that correspond at most to the user's vision of the universe of discourse. Besides arbitrary choosing the relationship types to consider, the user could also employ some formal mechanisms of reducing the number of different possible relationship types. For example, instead of considering all the $K^2$ pairwise combinations between subsets of sets $A$ and $B$, one could consider only a limited subset of $M$ particular combinations, where $1 \leq M < K^2$, ignoring the rest $K^2 - M$ combinations. Thus, for

instance, having the sets $A$ and $B$ separated into 8 subsets each ($A_i$ and $B_j$, where $i, j \in [1;8]$), and considering just the following two combinations of subsets of $A$ and $B$: $A_1 R^X B_4$ and $A_2 R^Y B_2$, where $R^X$ and $R^Y$ are any 4 basic relationships defined in sect. 3.4.1, we only obtain $4^2$=16 different relationship types instead of the total $N(4)$=183240 possible combinations. Another formal way of limiting the number of possible relationship types between two set-based multimedia elements $A$ and $B$ is by limiting the number of different relationships to consider between pairs of subsets of $A$ and $B$. For example, by taking into account only 2 representational relationships between $A_i$ and $B_j$ out of generally 4 possible options, namely only $\cap$ and $\bar{\cap}$, we reduce the number of possible relationship types between $A$ and $B$ to $2^{K^2}$, which, for instance, only gives 16 representational relationship types instead of 50 for the case of a simplest nontrivial separation of $A$ and $B$ (i.e. $K = 2$). Naturally, the two above-described formal mechanisms of reducing the number of possible multimedia representational relationship types could also be combined together. However, it is important to note that despite using the formal reduction approaches described here, the final decision on keeping or discarding a particular representational relationship should belong to a domain expert, since some representational relationships could simply not make any sense within the application domain context (see examples below).

In sect. 3.4.1 the hierarchical organization of the four basic representational relationship types was presented. It has been shown, in particular, that according to the set-based definitions of the multimedia representational relationships, the following fact takes place:

$$A{=}B \Rightarrow A{\subset}B \Rightarrow A{\cap}B.$$

The fact that the basic multimedia relationships are not completely orthogonal implies that the set-partitioning-based complex representational relationships are not completely independent neither.

For example, if $A$ and $B$ are partitioned into two subsets each, then the relationship $R^*$:

$$AR^*B : (A_1{=}B_1){\wedge}(A_1\bar{\cap}B_2){\wedge}(A_2\bar{\cap}B_1){\wedge}(A_2{\subset}B_2)$$

implies, for instance, the relationship $R^{**}$ that follows:

$$AR^{**}B : (A_1{\subset}B_1){\wedge}(A_1\bar{\cap}B_2){\wedge}(A_2\bar{\cap}B_1){\wedge}(A_2{\cap}B_2).$$

In situations like that it is up to the schema designer to choose either the most specific relationship that corresponds the most to the universe of discourse (e.g. $R^*$), or to choose among the more generic relationships to allow for a broader range of possible interpretations (e.g. $R^{**}$).

Referring to the hierarchy of representational relationships presented in fig. 3.15, we extended this hierarchy to take into account the additional set-partitioning-based relationship types, which can be divided in two subfamilies: $AR^{\cap}B$ and $AR^{\subset}B$, where:

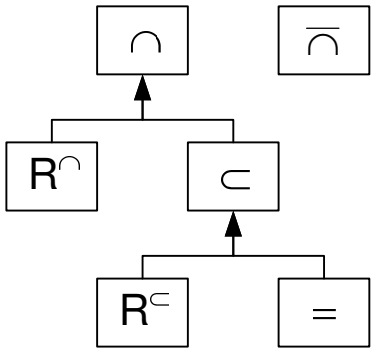



**Figure 3.23:** Hierarchy of additional multimedia representational relationships.

$$AR^{\cap}B : \exists i, j : A_i \cap B_j,$$
$$AR^{\subset}B : \forall i, \exists j : A_i \subset B_j.$$

It is easy to see that:

$$AR^{\cap}B \Rightarrow A \cap B, \text{ and}$$
$$AR^{\subset}B \Rightarrow A \subset B.$$

Hence, considering the representational relationships $AR^{\cap}B$ and $AR^{\subset}B$, we define an extended hierarchy of multimedia representational relationships, which is shown in fig. 3.23.

For instance the representational relationship $R^{**}$ above belongs to the family of relationships $R^{\cap}$, while $R^{*}$ belongs to the family of relationships $R^{\subset}$.

It is easy to notice that the basic disjointness representational relationship type can be alternatively expressed in a set-partitioning-based manner as:

$$A \bar{\cap} B \Leftrightarrow A_i \bar{\cap} B_j, \forall i, j.$$

Moreover, the disjointness relationship type can have no subtypes, since having at least one pair $A_i R^X B_j$, where $R^X \neq \bar{\cap}$ (i.e. is not disjoint), automatically breaks the overall disjointness constraint yielding a least a $R^{\cap}$-class relationship, i.e.:

$$\exists i, \exists j : A_i \cap B_j \Rightarrow AR^{\cap}B.$$

As for the multimedia equality relationship, it can also be alternatively expressed in a set-partitioning-based manner as:

$$A = B \Leftrightarrow \forall i : A_i = B_i.$$

It is important to note that unlike $R^{\cap}$-class representational relationships, in a set-partitioning-based form of the multimedia equality representational relationship the equal subsets of $A$ and $B$ must bear the same indexes (i.e. be homogeneous with respect to the set partitioning criterion), which can be proven by a contrary instance.

Proof:

Let's assume two multimedia elements $A$ and $B$ partitioned each in two subsets $A_1$, $A_2$, and $B_1$, $B_2$, respectively:

$$A = A_1 \cup A_2, \text{ where } A_1 \cap A_2 = \emptyset, \text{ and}$$

$$B = B_1 \cup B_2, \text{ where } B_1 \cap B_2 = \emptyset.$$

Partitioning $A$ and $B$ is governed by the selfsame partitioning criterion $P$ such that:

$$P(X)=X_1 \text{ and } \neg P(X)=X_2.$$

Hence:

$$(P(A)=A_1) \wedge (\neg P(A)=A_2), \text{ and}$$
$$(P(B)=B_1) \wedge (\neg P(B)=B_2).$$

Let's assume that:

$$(A_1=B_2) \wedge (A_2=B_1) \Rightarrow A=B,$$

meaning that $A$ and $B$ are equal if their heterogeneous subsets are pairwise equal.

It follows then that:

$$\left\{ \begin{array}{l} (P(A)=A_1) \wedge (\neg P(A)=A_2) \\ (A_1=B_2) \wedge (A_2=B_1) \\ A=B \end{array} \right| \Rightarrow (P(B)=B_2) \wedge (\neg P(B)=B_1),$$

which contradicts the separation criterion $P$.

□

Just like the multimedia disjointness, the multimedia equality cannot have any subtype representational relationships, since changing the equality by a different relationship in at least one pair $A_i=B_i$, would only weaken the overall relationship between A and B downgrading it to multimedia inclusion or intersection (see the hierarchy in fig. 3.23 above).

We believe that the main advantages of the set partitioning approach to introducing additional multimedia relationships as described in this chapter are: formality, simplicity, and extensibility.

The approach is formal because it is entirely built upon the formalism of the set theory, upon which the representation of multimedia elements is based.

The approach is simple because it provides a simple way of defining whatever number of additional multimedia relationship types without actually having to introduce any single new relationship type as such. Indeed, the definition of any new multimedia relationship type is entirely based on the definitions of the four predefined basic relationship types.

Finally, the approach is extensible because it allows introducing any arbitrary number of new more detailed multimedia relationship types, whose semantics is not restricted to any particular criteria.

Summarizing the above-said, to introduce new multimedia relationship types additional to the four basic ones, the user must simply specify the semantic criteria according to which set-based representations of multimedia elements should be partitioned into subsets. By comparing the obtained subsets pairwise using the four basic multimedia relationship types, the user then selects some particular relationship types that he is interested in out of the total of $N(K)$ combinations

(see equation 3.2), where $K$ is the number of subsets in a separation.

A similar idea of defining relationship types by considering pairwise combinations of subsets of domain members was used by Egenhofer and Franzosa [EF91] to provide a complete set of nine topological spatial relationships. Using the point-set topological model, they propose to partition each spatial object into its boundary ($\partial$) and interior ($\circ$). Considering four possible pairwise combinations of boundaries and interiors, authors come up with 16 resulting spatial relationships, only 9 of which are retained as such that make sense (see table 3.4).

| | $\partial \cap \partial$ | $\circ \cap \circ$ | $\partial \cap \circ$ | $\circ \cap \partial$ | |
|---|---|---|---|---|---|
| r0 | $\emptyset$ | $\emptyset$ | $\emptyset$ | $\emptyset$ | $A$ and $B$ are disjoint |
| r1 | $\neg\emptyset$ | $\emptyset$ | $\emptyset$ | $\emptyset$ | $A$ and $B$ touch |
| r3 | $\neg\emptyset$ | $\neg\emptyset$ | $\emptyset$ | $\emptyset$ | $A$ equals $B$ |
| r6 | $\emptyset$ | $\neg\emptyset$ | $\neg\emptyset$ | $\emptyset$ | $A$ is inside of $B$ or $B$ contains $A$ |
| r7 | $\neg\emptyset$ | $\neg\emptyset$ | $\neg\emptyset$ | $\emptyset$ | $A$ is covered by $B$ or $B$ covers $A$ |
| r10 | $\emptyset$ | $\neg\emptyset$ | $\emptyset$ | $\neg\emptyset$ | $A$ contains $B$ or $B$ is inside of $A$ |
| r11 | $\neg\emptyset$ | $\neg\emptyset$ | $\emptyset$ | $\neg\emptyset$ | $A$ covers $B$ or $B$ is covered by $A$ |
| r14 | $\emptyset$ | $\neg\emptyset$ | $\neg\emptyset$ | $\neg\emptyset$ | $A$ and $B$ overlap with disjoint boundaries |
| r15 | $\neg\emptyset$ | $\neg\emptyset$ | $\neg\emptyset$ | $\neg\emptyset$ | $A$ and $B$ overlap with intersecting boundaries |

**Table 3.4:** Egenhofer topological spatial relationships.

Independently of different application domains (namely, multimedia data vs. spatial data), our approach is more general and is characterized by a number of considerable differences as compared with the approach proposed by Egenhofer and Franzosa.

Firstly, we do not limit either the number or rationale of subset partitioning, thus allowing the users to employ any arbitrary criteria of dividing multimedia elements into clusters of subsets. Consequently, the number of resulting multimedia relationship types is also not a priori limited.

Secondly, due to inherent properties of connected topological spaces, as well as peculiarities of interior-boundary partitioning criteria, where the boundary fully and deterministically depends on the interior, only the two most generic mutually-orthogonal relationships (namely, disjointness and intersection) are used in [EF91] to perform pairwise comparison of subsets (i.e. interiors and boundaries). Yet, using only multimedia disjointness and multimedia intersection relationship types would be insufficient for the approach we propose, since no particular restrictions on the choice of subset partitioning criteria are imposed, and the resulting subsets (partitions) are generally mutually independent.

Let's illustrate this with an example. Suppose a bitmap space in fig. 3.24 repre-

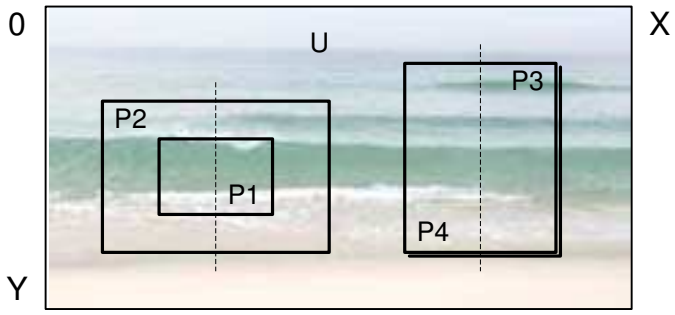



**Figure 3.24:** Demonstrating modeling differences between complex representational relationships and Egenhofer topological relationships.

senting a universal set of pixels $U$. A particular bitmap image $S$ ($S \subset U$) representing groups of neighboring pixels within $U$ is separated into $S^L$ and $S^R$ ($S^L \cup S^R$, and $S^L \cap S^R = \emptyset$), where $S^L$ is the left part of the image $S$, and $S^R$ is its right part.

Having 2 images $A$ and $B$ ($A \subset U$, $B \subset U$) it is easy to see that using only four following possible pairwise combinations of subsets of $A$ and $B$: $A^L R_X B^L$, $A^L R_X B^R$, $A^R R_X B^L$, $A^R R_X B^R$, where $R_X$ is either $\cap$ or $\bar{\cap}$, it is impossible to express even such basic relationship as equality, i.e. $A=B$. Indeed, the compound relationship that would correspond the most to the equality relationship would be the following one:

$$(A^L \cap B^L) \wedge (A^R \cap B^R) \wedge (A^L \bar{\cap} B^R) \wedge (A^R \bar{\cap} B^L), \tag{3.3}$$

which is also the combination that corresponds to equality relationship in Egenhofer's approach (i.e. interiors intersect, and boundaries also intersect, while the other two subset pairs do not intersect). However, the condition in equation 3.3 holds for both pairs $P1$, $P2$ and $P3$, $P4$ in fig. 3.24:

$$(P1^L \cap P2^L) \wedge (P1^R \cap P2^R) \wedge (P1^L \bar{\cap} P2^R) \wedge (P1^R \bar{\cap} P2^L),$$
$$(P3^L \cap P4^L) \wedge (P3^R \cap P4^R) \wedge (P3^L \bar{\cap} P4^R) \wedge (P3^R \bar{\cap} P4^L),$$

nevertheless, only $P3=P4$, and $P1 \neq P2$.

To be able to express the equality relationship in such a way, we should also be able to use the equality relationship between the separation subsets, namely:

$$P3=P4 \Leftrightarrow (P3^L=P4^L) \wedge (P3^R=P4^R) \wedge (P3^L \bar{\cap} P4^R) \wedge (P3^R \bar{\cap} P4^L).$$

As a matter of fact, the particular inclusion relationship between $P1$ and $P2$ can also be defined more precisely in the similar manner:

$$P1 \subset P2 \Leftrightarrow (P1^L \subset P2^L) \wedge (P1^R \subset P2^R) \wedge (P1^L \bar{\cap} P2^R) \wedge (P1^R \bar{\cap} P2^L).$$

### 3.4.4 Examples of Complex Multimedia Representational Relationships

Having introduced the general set-partitioning-based approach to definition of supplementary multimedia relationship types, let's demonstrate the applicability of

this approach with a series of examples.

### Example #1

The set-partitioning approach plays an extremely important role for multimedia elements of complex multimedia types by providing a mechanism for defining additional multimedia relationships limited to some particular types of media and ignoring the other media types pertinent to the complex nature of multimedia elements that take part in the relationship. Thus, for instance, comparing two multimedia elements of complex multimedia type `AudioVideoText` (`AVT`), it can be required to represent the multimedia inclusion relationship considering only the textual parts of the multimedia elements and ignoring any possible correlation between the audiovisual parts thereof.

The set-partitioning approach fits perfectly into this case. Indeed, as described in sect. 3.3, the complex multimedia datatypes from the hierarchy on fig. 3.11 provide each a number of standard method functions that return the single-media components of a composite multimedia element. For example, an `AudioVideoText` multimedia element has methods that return the element's audio part, its video part, and its textual part. As a matter of fact, these class methods provide exactly the mechanism of partitioning a complex multimedia element into a collection of single media type subsets. By comparing these single-media components of two multimedia elements pairwise, a series of new multimedia relationship types can be defined.

For example, having two multimedia elements $M_F$ and $M_D$ of datatype `AudioVideoText`, with the multimedia element $M_F$ representing a movie with French subtitles, and the second element $M_D$ representing the same movie with German subtitles, we can express the relationship of audiovisual equality of $M_F$ and $M_D$ even when their textual components are not equal. By separating $M_F$ and $M_D$ into audio, video, and textual parts:

$$M_i = M_i^A \cup M_i^V \cup M_i^T,$$
$$\text{where } \forall i \in \{F; D\}, \forall X, Y \in \{A, V, T\}, X \neq Y : M_i^X \cap M_i^Y = \emptyset,$$

we can define the sought relationship as follows:

$$M_F R_{=}^{AV} M_D : (M_F^A = M_D^A) \wedge (M_F^V = M_D^V).$$

Since no particular correlation between the textual parts $M_F^T$ and $M_D^T$ is imposed, the relationship $R_{=}^{AV}$ occurs, whichever of the four basic multimedia relationships holds between the textual parts of complex multimedia elements. It is easy to see that in the special case of equality of the textual components of $M_F$ and $M_D$, the $R_{=}^{AV}$ relationship becomes identical to the regular equality relationship:

$$M_F = M_D : (M_F^A = M_D^A) \wedge (M_F^V = M_D^V) \wedge (M_F^T = M_D^T).$$

### Example #2

In the example 1 in sect. 3.4.2 a town plan application was described. The multimedia elements of type `Image` were considered, with each image defined as a set of coordinate pairs on the image bitmap, i.e. $\{(x, y)\}$. Also the four basic multimedia relationship types were defined, describing pairwise intersection, inclusion, equality, and disjointness of image bitmaps within the global town plan image. For instance, on fig. 3.18 images $A1$ and $B1$, as well as $A1$ and $B2$ intersect: $A1 \cap B1$ and $A1 \cap B2$. Nevertheless, using only the set of four basic multimedia relationship types it is impossible to specify that $B1$ intersects with $A1$ in the upper right part of $B1$, and that $B2$ intersects with $A1$ in the upper left part of $B2$.

To accomplish this task, we propose to partition each image $P$ into top-left, top-right, bottom-left, and bottom-right quadrants:

$$P = P_{TL} \cup P_{TR} \cup P_{BL} \cup P_{BR},$$

where $\forall i, j : P_i \cap P_j = \emptyset$, so that:

$$\begin{aligned}
P_{TL} &= \{(x, y) : x \in [x_0; \frac{x_0 + x_N}{2}]; y \in [y_0; \frac{y_0 + y_N}{2}]\}, \\
P_{TR} &= \{(x, y) : x \in [\frac{x_0 + x_N}{2}; x_N]; y \in [y_0; \frac{y_0 + y_N}{2}]\}, \\
P_{BL} &= \{(x, y) : x \in [x_0; \frac{x_0 + x_N}{2}]; y \in [\frac{y_0 + y_N}{2}; y_N]\}, \\
P_{BL} &= \{(x, y) : x \in [\frac{x_0 + x_N}{2}; x_N]; y \in [\frac{y_0 + y_N}{2}; y_N]\}.
\end{aligned}$$

Using the previously defined formula it is easy to see that such a partitioning yields a maximum of $4^{4^2} = 4G$ new possible relationship types. However, it is not necessary to deal with all the possible combinations, and users can simply choose the subset of relationships they are interested in. Moreover, not all combinations have sense. E.g. for any pair of images $A$ and $B$ of minimal dimension $2 \times 2$ pixels, the following family of relationships could never hold:

$$(A_{TL} \cap B_{BR}) \wedge (A_{BR} \cap B_{TL}).$$

Partitioning the set representations of images into quadrants allows defining additional more detailed multimedia relationship types like the ones shown in fig. 3.25.

Relationships $R_1$, $R_2$, and $R_3$ in fig. 3.25 are formally defined as follows:

| | $A_{TR}$ vs. $B_{TL}$ | $A_{TR}$ vs. $B_{BL}$ | $A_{BR}$ vs. $B_{BL}$ | $A_X$ vs. $B_Y$ |
|---|---|---|---|---|
| $R_1$ | $\bar{\cap}$ | $\cap$ | $\bar{\cap}$ | $\bar{\cap}$ |
| $R_2$ | $\cap$ | $\bar{\cap}$ | $\cap$ | $\bar{\cap}$ |
| $R_3$ | $=$ | $\bar{\cap}$ | $=$ | $\bar{\cap}$ |

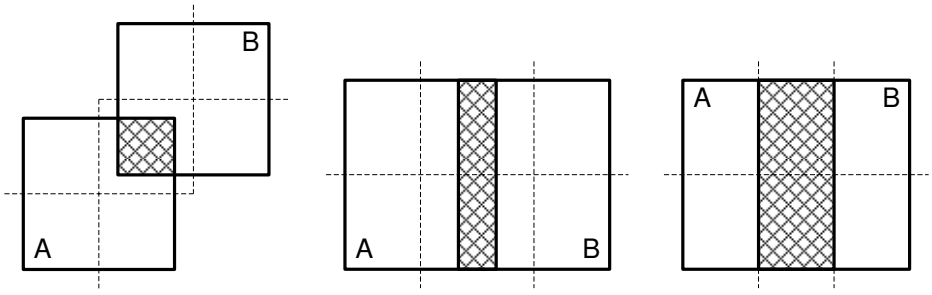



**Figure 3.25:** Example of complex representational relationships based on spatial image partitioning.

where "$A_X$ vs. $B_Y$" column represents the remaining 13 pairwise combinations of $A_X$ and $B_Y$.

Semantically $R_1$, $R_2$, and $R_3$ are described as:
$R_1$: images $A$ and $B$ intersect by the top-right corner of $A$ and the bottom-left corner of $B$.
$R_2$: images $A$ and $B$ overlap horizontally, with the image $B$ tiled at the right of the image $A$.
$R_3$: images $A$ and $B$ overlap horizontally, with the right part of $A$ equal to the left part of $B$.

It is interesting to note that since:

$$\forall R_i, i{\in}\{1,2,3\}, \exists I, J : A_I{\cap}B_J{\neq}\emptyset,$$

i.e. if either $R_1$, or $R_2$, or $R_3$ holds, then there exists at least one pair of intersecting quadrants, hence all of the three relationship types $R_1$, $R_2$, $R_3$ shown in fig. 3.25 are subtypes of the intersection relationship type, i.e.:

$$AR_iB \Rightarrow A{\cap}B, \forall i{\in}\{1,2,3\}.$$

Moreover, since the multimedia equality relationship type is a subtype of multimedia inclusion, hence the relationship type $R_3$ is a subtype of the relationship type $R_2$:

$$AR_3B \Rightarrow AR_2B.$$

### Example #3

In the example 2 in sect. 3.4.2 a sample HR application was described. The multimedia elements of type `Image` were considered, with each image defined as a set

of persons represented on the image. The four basic multimedia relationship types were defined, describing pairwise intersection, inclusion, equality, and disjointness of images. These relationship types are based on annotations, which indicate the presence or absence of the same people on different photos. For example, two department group photos are said to be intersecting iff both photos portray at least one same person.

To better adapt to application requirements, a set of additional multimedia relationship types could be introduced to allow further detailing the semantics of existing multimedia relationships. Consider the following subset separation criteria for an image $P$:

$$P = P^M \cup P^E,$$

where $P^M$=$\{x|x$ is a manager$\}$, and $P^E$=$\{x|x$ is a regular employee$\}$. I.e. every set description of a photo is divided into two subsets: the managerial employees portrayed on the photo, and the regular employees on the photo.

Suppose that $D_1$ and $D_2$ are two group photos of two different company departments. Then, using the newly introduced separation criterion, a maximum of $4^{2^2}$=256 new relationship types can be introduced, like, for example, the relationships $R_1$, $R_2$, $R_3$ defined below:

| | $D_1^M$ vs. $D_2^M$ | $D_1^M$ vs. $D_2^E$ | $D_1^E$ vs. $D_2^M$ | $D_1^E$ vs. $D_2^E$ |
|---|---|---|---|---|
| $R_1$ | $\bar{\cap}$ | $\subset$ | | |
| $R_2$ | $\cap$ | | | |
| $R_3$ | | $\bar{\cap}$ | $\bar{\cap}$ | |

Semantically the relationship types $R_1$, $R_2$, $R_3$ are defined as follows:

$R_1$: The managers of the department $D_1$ are all regular employees of the department $D_2$.

$R_2$: There are common managers in $D_1$ and $D_2$.

$R_3$: Managers of one department cannot be regular employees of the other department.

It is important to note that empty cells in the table above mean any of the four basic multimedia relationship types. From this point of view $R_1$, $R_2$, $R_3$ should be considered not as regular relationship types, but rather as relationship type families. Putting particular relationships instead of the empty cells, or else substituting some already defined relationships with their more specific version (e.g. changing $D_1^M \subset D_2^E$ by $D_1^M = D_2^E$ in the definition of $R_1$) allows obtaining even more specific versions of $R_1$, $R_2$, $R_3$.

**Figure 3.26:** E-FIT pictures.

### Example #4

In the example 2 above a set of additional multimedia relationship types was introduced to provide pairwise comparison of city region images belonging to a global city plan image ($U$). Each image was represented by a set of pixel coordinates within the global image $U$.

Another example of an image-enhanced application that can benefit from the set-separation based approach is an E-FIT system [PBK00]. E-FIT (Electronic Facial Identification Technique) is a computerized method of synthesizing images of the faces of criminals from witness descriptions, which is commonly used by police and other law-enforcement institutions around the world. Unlike the city plan application, the E-FIT images are not part of any global image, and can generally be represented as:

$$\{(x, y, c)\},$$

i.e. a triplet representing pixel coordinates and color.

Since E-FIT images are composed of pre-defined images of human facial parts, i.e. eyes, eyebrows, chins, noses, haircuts, etc., then partitioning E-FIT images into subsets corresponding to various parts of the face allows introducing multimedia relationship types comparing two E-FITs based on facial features.

Suppose two E-FITs $P_1$ (left) and $P_2$ (right) shown in fig. 3.26. $P_1$ and $P_2$ are partitioned into subsets representing the nose, the chin, the eyes, and the rest of the face:

$$P_i = P_i^{nose} \cup P_i^{chin} \cup P_i^{eyes} \cup P_i^{rest}.$$

This partitioning criterion allows introducing multimedia relationship types like

$R_1$, $R_2$, $R_3$ described below:

$$\begin{aligned} R_1 &= (P_1^{nose}{=}P_2^{nose}) \wedge (P_1^{eyes}\bar{\cap}P_2^{eyes}), \\ R_2 &= (P_1^{nose}{=}P_2^{nose}) \wedge (P_1^{eyes}\bar{\cap}P_2^{eyes}) \wedge (P_1^{chin}{=}P_2^{chin}), \\ R_3 &= (P_1^{nose}\bar{\cap}P_2^{nose}) \wedge (P_1^{eyes}{=}P_2^{eyes}) \wedge (P_1^{chin}\bar{\cap}P_2^{chin}). \end{aligned}$$

Semantically the relationship types $R_1$, $R_2$, $R_3$ can be defined as:
$R_1$: both persons have the same nose, but different eyes.
$R_2$: both persons have the same nose and chin, but different eyes.
$R_3$: both persons have the same eyes, but different noses and chins.

Obviously, the relationship type $R_2$ is a subtype of $R_1$, i.e.:

$$AR_2B \Rightarrow AR_1B.$$

In particular, for the two E-FITs shown in fig. 3.26, the relationship $R_3$ takes place.

### Example #5

In the examples 3 and 4 in sect. 3.4.1 two sample approaches to set-based representation of video data were described. The general idea is to represent a video clip as a set of pairs of video frames and their timestamps: $V{=}\{(f_i, t_i)\}$. Also the four basic multimedia relationship types were introduced for the case of video data. In order to better meet the user requirements, a subset-partitioning-based approach can be used to introduce additional video relationship types.

Suppose a digital movie collection, where every film starts with a fifteen second reel of the media company that produced the movie (e.g. Universal Studios, Paramount, Warner Bros., etc.). Since the introduction reels for all the movies produced by a certain media company are the same, we can provide a media-based comparison of movies based on their production house. The following subset partitioning criterion is proposed:

$$V = V^{15-} \cup V^{15+},$$

where $V^{15-}{=}\{(f_i, t_i) : t_i{\in}[t_0; t_0{+}15]\}$ and $V^{15+}{=}\{(f_i, t_i) : t_i{\in}[t_0{+}15; t_N]\}$, i.e. every movie is separated in two parts: the first 15 seconds, and the rest of the movie.

Although such partitioning allows introducing almost $4^{2^2}{=}256$ additional video relationship types, only 2 distinct families thereof make sense:

$$R_1 : V_1^{15-}{=}V_2^{15-} \text{ and } R_2 : V_1^{15-}\bar{\cap}V_2^{15-}.$$

Indeed, the relationship type $R_1$ above implies that the movies $V_1$ and $V_2$ are produced by the same media company, while the relationship type $R_2$ implies

heterogeneous origins of $V_1$ and $V_2$.

In the case when movies begin with several introduction reels of arbitrary length, which is often the case in the modern film-making industry, a shot-based representation of video data could be used instead of the per-frame representation. Consequently each movie can be defined as a sequence of shots, with any introductory reel also corresponding to a particular shot. Then by separating a movie ($V$) in two contiguous parts: the sequence of introductory reels ($V^I$) and the movie itself ($V^M$) we can define two additional families of video relationship types:

$$R_3 : V_1^I \cap V_2^I \text{ and } R_4 : V_1^I \subset V_2^I.$$

The relationship type $R_3$ above implies that some media companies took part in producing both $V_1$ and $V_2$, while the relationship type $R_4$ implies that all of the media companies who produced $V_1$ have also produced $V_2$. The previously defined relationship types $R_1$ and $R_2$ also get a more generalized meaning, since they allow considering several introductory reels per movie instead of just one in the previous definition.

### 3.4.5 Section Summary

In this section we have introduced a special kind of relationships called *multimedia representational relationships*, which allow enriching relationship types in MADS with additional constraints of multimedia essence.

In the sect. 3.4.1 we have introduced four basic representational relationship types and have further provided a mechanism of introducing additional representational relationship types in the sect. 3.4.3.

While multimedia representational relationships rely on the abstract set-based definition of multimedia data introduced in the sect. 3.3.3, we have provided in sect. 3.4.2 and sect. 3.4.4 several examples of implementing abstract set-based representational relationships in different application environments.

In the next section we introduce a formal approach to set-based partitioning of conceptual multimedia elements, on which multimedia representational relationships rely.

## 3.5 Formal Methodology for Multimedia Data Partitioning

As demonstrated in the sect. 3.4.3, the set-based partitioning of abstractly defined multimedia data plays an important role in the context of defining complex custom

multimedia representational relationship types. Nevertheless, until now no formal approach to partitioning of conceptual multimedia elements has been provided. In this section we propose a generic methodology that allows to effectively partition multimedia data at the conceptual level independently of any limited set of partitioning criteria.

### 3.5.1 Classical Multimedia Segmentation Techniques

As we have previously demonstrated, users are free to choose whatever particular representation model they prefer behind the abstract set-based definitions of multimedia elements (see sect. 3.3.3). This in its turn implies a priori unlimited number of possible set partitioning criteria for multimedia elements. Nevertheless, such partitionings would often follow some traditional partitioning semantics that is peculiar to the abstract datatype of the multimedia element.

For instance, in the case of multimedia elements of type `Image`, users would generally want to base their image partitioning criteria on the foundations of various *image segmentation* approaches. Image segmentation is a classical problem in digital image processing and computer vision [SS01]. It can be described as the process of partitioning an image into a set of non-overlapping homogeneous regions:

$$P=\bigcup_{i=1}^{n}, \text{ where } \forall i\neq j : P_i\cap P_j=\emptyset.$$

The homogeneity within image regions is defined by a certain criterion, which could be based, for example, on grayscale levels or color histograms of the image, variations in textures or texture scales, etc. In fact, there exists an ample quantity of segmentation criteria, and their diversity conditions the multitude of possible segmentations. Thus, the solution of the segmentation problem is in general not unique and a variety of possible solutions is the result of a great deal of subjectivity inherent to the segmentation task.

Among the existing image segmentation techniques found in the literature one could distinguish [GW07]:

- Clustering.
  First, different feature vectors are extracted by means of classical image processing techniques (e.g. color histograms, wavelet analysis, etc.). Clustering algorithms are then used for obtaining set representatives.

- Edge based approaches.
  Consist in tracking the border of objects through edge detection techniques either based on filtering or active contours.

- Watersheds.
  Conceptually follows the idea of region growing. Pixel intensities are taken as heights and the level sets are progressively flooded. Then regions are estimated based on accumulation basins.

- Probabilistic approaches.
  A set of methods where one tries to model the distribution of regions within an image.

- Thresholding.
  Segmenting the image histogram into significant regions. A further refinement allows for local thresholding.

- Active Contours.
  A moving curve is attracted by regions of high gradient corresponding to edges in the image. Provided one correctly handles topology changes, multiple objects can then be segmented.

- Watersnakes.
  Represent energy-driven watershed segmentation using the distance-based definition of the watershed line. A priori smoothness information can be imposed to the energy functional leading to better segmentation results than the original watershed technique.

The number of possible image segmentation approaches like the ones described above is a priori unlimited and various segmentation techniques can be combined to form new complex segmentations. Furthermore, it is important to provide conceptual level partitioning of set-based representations of multimedia elements akin to various media segmentation approaches, like, for instance, image segmentation techniques used in digital image processing.

Hence, at the conceptual modeling level we should provide a proper methodology for defining multimedia set separations, which are:

- formal (i.e. based on formal foundations);

- abstract enough (in order to conform to abstract set-theory-based definitions of multimedia elements);

- extensible (i.e. not limited to pre-defined set of possible partitioning criteria);

- flexible (i.e. allowing complex partitionings by combining multiple partitioning criteria).

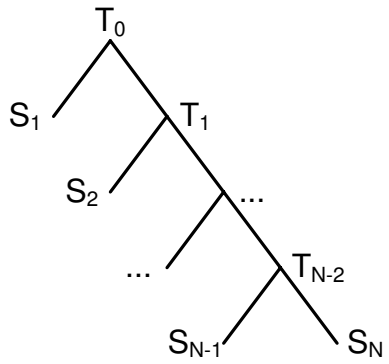



**Figure 3.27:** A binary tree partitioning a text element into sentences.

### 3.5.2 Binary-Tree Based Multimedia Partitioning Approach

Taking into account the requirements specified in the previous subsection, we have decided to provide a multi-criterial binary-tree-based approach to partitioning of multimedia elements. The general idea of our approach consists in progressively partitioning a set-based representation of a multimedia element by splitting it into two parts (subsets), which can then be further split in two parts each, and so on.

Several similar approaches of binary partitioning trees have been developed by the image processing community (see e.g. [SG00]). Compared to these techniques our binary-tree multimedia partitioning approach does not limit itself to only the pixel-based representation of images, but is applicable to any type of multimedia data independently of its underlying representation. Moreover, as it will be shown further on, our partitioning approach allows to combine together different partitioning criteria.

Let's illustrate our multimedia partitioning approach with an example. Consider a multimedia element $T_0$ of type `Text`. Suppose that one of the methods defined with the datatype `Text` allows obtaining the first sentence of the text element (`Text FS(self)`). Thus, in order to partition the text element into sentences, the method `FS()` has to be applied progressively forming a binary tree of sentences (fig. 3.27).

At the first step, the entire text $T_0$=$self$ (tree root) is divided into two parts, namely the first sentence of the text ($S_1$) and a tail $T_1$ (the rest of the text). The procedure is then progressively applied to the text tail:

1) $T_0=S_1 \cup T_1$, where $S_1=FS(T_0)$ and $T_1=T_0 \setminus S_1$;
2) $T_1=S_2 \cup T_2$, where $S_2=FS(T_1)$ and $T_2=T_1 \setminus S_2$;
...
$N-2$) $T_{N-3}=S_{N-2} \cup T_{N-2}$, where $S_{N-2}=FS(T_{N-3})$ and $T_{N-2}=T_{N-3} \setminus S_{N-2}$;
$N-1$) $T_{N-2}=S_{N-1} \cup S_N$, where $S_{N-1}=FS(T_{N-2})$ and $S_N=T_{N-1}=T_{N-2} \setminus S_{N-1}$.

In the example above (fig. 3.27) a complete partitioning tree dividing a textual multimedia element into its composing sentences is represented. However, due to a stepwise nature of our algorithm, constructing the entire partitioning tree right away is not necessarily required. For instance, in the text partitioning example above, in order to obtain the second sentence of a textual element, we only need to execute 2 steps of the algorithm, i.e. only build the first two levels of the partitioning tree:

$$S_2{=}FS(T_0 \backslash FS(T_0)).$$

This simplifies and speeds up a lot the partitioning process, since obtaining the $i$-th sentence of a text block is independent of the total number of sentences in the text, be it just $i$ sentences, or $10^i$ sentences. Putting it another way, the partitioning process becomes deterministic with respect to the power of partitioning (i.e. with respect to the number of partitionings that can possibly be obtained).

This last property becomes particularly important in networked environments when dealing, for instance, with streaming multimedia data, or even with distributed multimedia data in P2P networks. In fact, the stepwise nature of the proposed partitioning approach allows working with a part (a segment) of a streaming multimedia element as soon as it arrives, without necessarily having to wait for the rest of the stream (or for the other parts of the distributed media source). In this sense, the binary-tree-based multimedia partitioning mechanism could even be used as the abstract modeling foundation of streaming multimedia data, where the streamed data is represented as a binary tree, which is constructed ("grows") progressively as the data keeps on arriving (see e.g. fig. 3.27).

Another advantage of the step-wise partitioning approach is the possibility of performing complex multi-criteria partitionings by applying different partitioning functions at different steps (i.e. at different tree nodes). Recurring to the example of a binary-tree partitioning of a textual element into its composing sentences (fig. 3.27), imagine that we would like to further partition text sentences into words. Suppose that the multimedia datatype `Text` provides another method function `Text FW(self)`, which returns the first word of the text. Then, by combining different partitioning criteria (e.g. "by word" and "by sentence") we can obtain complex multi-criteria partitionings. For instance, to get the third word of the second sentence of a text block, it is sufficient to perform 5 partitioning steps (see fig. 3.28).

As one can notice from fig. 3.28, just as it is the case with single-criterion partitionings, also the multi-criteria partitionings do not require to have a complete tree constructed beforehand in order to obtain some particular segments. As it was mentioned above, this progressive nature of our binary-tree-based partitioning approach is particularly suitable to represent e.g. streaming multimedia data in networked environments. Moreover, multi-criteria progressive partitionings provide additional degree of flexibility when working with streaming media applications.

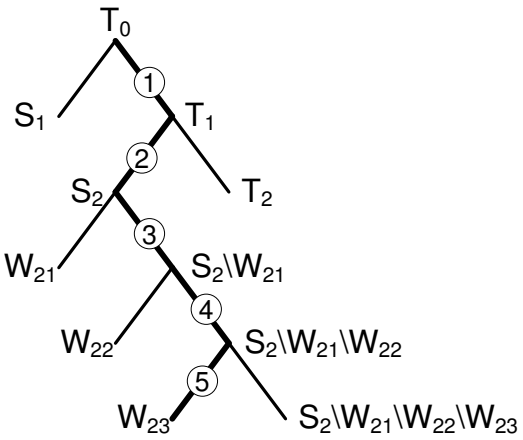



**Figure 3.28:** An example of a multi-criteria partitioning tree.

Indeed, in such kind of environments, multimedia data is rarely partitioned using only one single criterion. Instead, media streams are often first split into sections by a timeline component (either fixed-interval timestamp-based partitionings, or varying-interval semantic-based partitionings relying on organization of certain types of streaming videos into sequences of shots, scenes, cuts, etc.) [YWX+07] [CLT08]. Once this initial time-based pre-segmentation is done, each individual resulting segment can be further analyzed and partitioned by some different segmentation criteria, e.g. image-based analysis using color- and texture-based approaches, or using moving-objects-based MPEG2-like approaches, etc.

Further considering our multi-criteria progressive tree-based partitioning approach applied to streaming and distributed multimedia data, one can state that it can be easily adopted as a formal conceptual foundation for MPEG-7 descriptions. MPEG-7 [Mar04] [MSS02] is a popular emerging multimedia content description standards, which draws a lot of attention nowadays both in research and industry. Issued from the Moving Picture Experts Group, MPEG-7 unlike the other standards from the MPEG group does not deal with actual encoding of moving pictures and audio information. Instead, it uses XML to store metadata descriptions, which can then be encoded in a binary form and timeline-attached to a physical media stream. This, for instance, provides a mechanism to tag particular events, or synchronize lyrics to a song, or deliver subtitles with the video data, etc. In this way, when a subscriber is receiving a MPEG-7 enhanced media stream, he/she can extract the accompanying metadata descriptions on the fly without having to wait until the delivery finishes.

This principle fits perfectly into our multi-criteria progressive tree-based partitioning approach. Indeed, to construct a partitioning tree capable of extracting MPEG-7 descriptions, a streaming media source extended with MPEG-7 descriptions can be first pre-partitioned on timeline basis, and then the obtained partitions

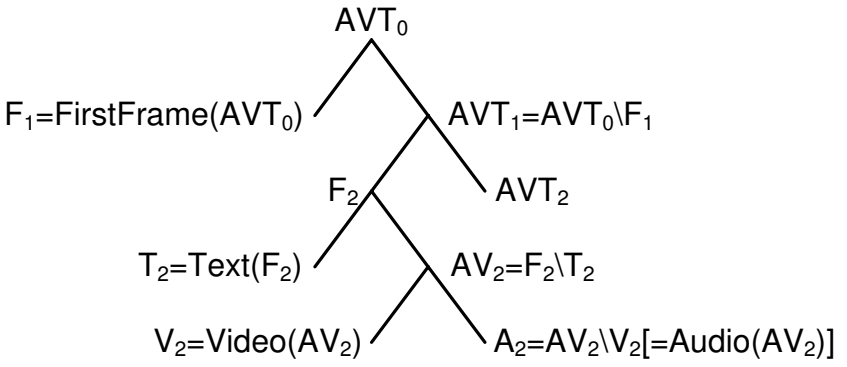



**Figure 3.29:** MPEG-7 like partitioning tree for streaming multimedia data.

can be further processed to extract the MPEG-7 metadata. For example, in case of a streaming media data of type `AudioVideoText`, where the audio-video stream is enhanced with a subtitle track in MPEG-7 format, the initial temporal per-frame partitioning is then further augmented with track-based segmentation allowing the extraction of subtitles (see fig. 3.29).

It should be noted that independently of MPEG-7 descriptions, the partitioning tree in fig. 3.29, generally speaking, demonstrates an important feature of multimedia analysis, namely media-based partitioning of multimedia elements of complex datatypes. As mentioned in sect. 3.3, all complex multimedia datatypes in MADS (e.g. `AudioVideoText`, `PictureText`, etc.) provide method functions for extracting single-media components from complex-media data. Thus, for example, the method `Text()` of the complex multimedia datatype `AudioVideoText` (see fig. 3.29) is used to extract the textual part of the multimedia element. Using these media-splitting method functions in stepwise partitioning trees allows obtaining fine-grained media-type-based partitionings controlled by the preceding (higher-level) nodes of the partitioning tree, whether homo- or heterogeneous (i.e. higher-level nodes could possibly result from completely different partitioning criteria). For instance, in fig. 3.29, splitting a multimedia element of a complex type into its constituent media components (audio, video, and text) is not done at once for the entire multimedia element, but is rather selectively performed on a frame-level basis. In this way, the media-splitting partitioning are conditioned by their corresponding higher-level timeline-based partitionings.

### 3.5.3 Theoretical, Technological and Perceptional Limitations of the Tree-Based Partitioning

It is easy to show that in order to partition a multimedia element into $K$ segments, a binary partitioning tree of at least $[ceil(log_2(K))+1]$ nodes would be required. Let's investigate on the maximum size of such a tree.

Theoretically, a multimedia partitioning tree could be unlimited in size since any part of a multimedia element could be further partitioned into its constituent parts using a multitude of partitioning criteria, and so on recursively. In practice, however, multimedia partitioning trees would mostly be finite (although possibly very big in size). Firstly, this finitude can be explained by the fact, that just like any kind of data nowadays, most multimedia data is represented and treated in its digitized form. The numeric character of multimedia data imposes natural limitations on the maximal degree of their possible partitioning. This in its turn constrains the maximal size of the partitioning tree.

Let's consider, for instance, digital image data, which is represented in its bitmap form. Suppose that the size of a partitioning tree is expressed as the number of its leaves. In fact, the size of whatever partitioning tree applied to a numeric image will always be limited by the number of pixels in the image bitmap, since a bitmap pixel represents the finest element of an image, and therefore it cannot be partitioned any further.

Similarly, any partitioning tree for a multimedia element of type `Text` represented in a character-based digital form (e.g. ASCII, Unicode, etc.) is effectively limited in size by the total number of characters in the textual block, since in this case a single character would constitute the elementary unit of a text, and it would neither conceptually nor technically be possible to split individual characters into e.g. their composing phonetic sounds.

Nevertheless, various technical limitations imposed on digital representations of multimedia information of different types, which can altogether be characterized as *resolution limitations*, are by far not the only factors that determine the finitude of multimedia partitioning trees. As a matter of fact, a much more important limitation factor is the physiological and situational bounds of human perception.

Although not directly related to any technical characteristics of media rendering or capturing devices, human perception, as studied by cognitive sciences, is the process of acquiring, interpreting, selecting, and organizing sensory information by a human [Noe05]. Methods for studying human perception range from physiological (i.e. essentially biological) approaches through psychological through philosophical approaches. In fact, the scope of human abilities to perceive objects and phenomena of the surrounding reality imposes *perceptive resolution limitations* similar to those induced by technical constraints of various digitized media. For example, in case of visually perceived information, physiological limits of a human eye impose effective constraints on the maximal perceivable image resolution even when observing real-world scenery. Thus, in [WPLM01], a virtual reality setting based on a $10\times13$-foot display with a total of 20 million pixels was reported to provide image resolution quality at which people with a perfect visual acuity could not make a difference between a real world scene and a digital picture. Similarly, some other known limitations of human vision abilities, such as color distinction bound of approximately 10 million colors, and maximal perceivable picture refresh frequency of approximately

100∼120 Hz also limit the physiological bounds of human visual perception.

The examples above demonstrate that the concepts normally appropriate to digitized multimedia data can also effectively be applied to any kind of multimedia information whatsoever, including e.g. real world scenery observed directly by the spectator. The rationale behind this argumentation is that from the biological point of view, the perception limits of human sensory organs are close to (and sometimes even lower than) those of the existing digitizing and rendering devices for numerical multimedia data.

Last but not least, it is important to note that the distinctive characteristic of the human perception processes is the substantial role of the brain in the overall perception mechanism. In fact, due to the brain processing of the information obtained with the sensory organs, the human is capable to "see" beyond the biological limits of his/her eyes, ears, nose, etc. Thus, for instance, the stereoscopic vision ability of humans (i.e. basically the ability to not simply see an object, but also to evaluate the distance towards it) is provided not due to some kind of a special sensory organ, but is instead a feature of the brain, which deduces the distance information from the visual stimuli captured by the two eyes. Nevertheless, in our opinion, the role that the brain plays in the human perception process does not introduce substantial differences between the digital multimedia data representation and multimedia data representation model inherent to the biological mechanisms of human perception.

Firstly, although the part of the brain involvement in the overall perception process is indeed very high, the brain's functioning still heavily depends on the information it receives from the sensory organs. Thus, e.g. even the most creative human mind cannot reconstruct the color of infrared light, simply because it has no experimental notion of it, which is due to the fact that human eyes (unlike for example the eyes of some animals) cannot capture the light in the infrared spectrum.

Secondly, the brain involvement in the perception process is almost as high in the case of perceiving digitized multimedia information as it is in the case of perceiving analogous or even real-life information. For example, the stereovision effect, which we have mentioned above, can also be achieved in a digital image environment using a variety of existing hardware like stereoscopic glasses, stereo displays, etc. Moreover, the first stereoscopic devices based on analogous images date back to as early as the $19^{th}$ century [SS99]. In fact, no matter the nature or the origin of the visual information, it is the brain and the brain alone that is responsible for reconstructing the 3D scenery from the pairs of images.

In our opinion, the main difference that the brain does introduce into the multimedia perception process is the variation in interpretations of multimedia information amongst different users, or even for the same user. This peculiarity can be characterized as multi-representation / multi-perception problem and will be discussed in the sect. 3.6.

We can thus conclude that the existing forms of discrete numerical representation of multimedia data are general enough to be used at the conceptual modeling level even for applications adhering to the *multimedia as metadata* view, since such kind of applications are usually not interested in any particular media resolution details, but rather in the overall perception of the multimedia information by humans (annotators, final application users, etc.).

### 3.5.4 String-based Representation of Multimedia Partitioning Trees

Referring to elements of a tree (tree nodes) introduced in sect. 3.5.2 using the exact mathematical formulae leading from the tree root to a tree node is somewhat inconvenient. Moreover, since our binary partitioning trees can get very big in size, and a large span of different partitioning criteria can be used within the same tree, referring to individual tree nodes (i.e. particular multimedia partitions) can become especially difficult.

For this reason we propose an alternative notation technique to identify particular multimedia partitions using string-like identifiers.

It should be noted that in many applications using binary trees and especially applications using binary search (e.g. relational database indexes), the particular path in the tree structure is not of a great importance, and what matters mostly is the data stored in the node (e.g. the reference to a particular data block stored in an index node). Moreover, in such kind of applications tree paths are generally not deterministic and can change if e.g. the index is reconstructed. However, our multimedia partitioning trees do not belong to this class of applications. What interests us is not just the particular node itself, but rather its path in the partitioning tree, since it is this path itself that defines a particular segment that a tree node represents.

A path in a partitioning tree represents in fact a sequence of transformations applied to the initial multimedia element (tree root). At each step some transformation function corresponding to a partitioning method is applied to the result of the previous transformation and so on, i.e.:

$$N_{i+1}{=}P_X(N_i), \text{ or } N_{i+1}{=}N_i \backslash P_X(N_i).$$

#### Homogenous Partitioning Trees

In case of homogenous partitioning trees (i.e. with the selfsame $P_X$ used throughout the entire tree), a sequence of "0" and "1" could be used to identify a path within the partitioning tree, where "0" means following the left sub-branch, and "1" means following the right sub-brunch. For example, the $4^{th}$ sentence of a text block in the partitioning tree in fig. 3.27, can be denoted as "1110".

In this way, any sentence of a text block can uniquely be determined as a triplet:

$$\langle T_i, Text.FS(), path\_string \rangle, \tag{3.4}$$

where $T_i$ - is the text block, $Text.FS()$ - is the segmentation method function returning the first sentence of a text, and *path_string* - is a 0/1-string identifying a path in the binary partitioning tree. Thus, the $4^{th}$ sentence of a text element, whose partitioning tree is represented in fig. 3.27, is defined by the following triplet:

$$S_4 = \langle T_0, Text.FS(), \text{`1110'} \rangle. \tag{3.5}$$

Such string-based representation is not only more intuitive but is also much more compact compared to the explicit algebraic representation of partitions. For example, compared to the string-based representation of $S_4$ in expression 3.5, the underlying algebraic expression of $S_4$ is:

$$\begin{aligned} S_4 = \quad & \{\{[T_0 \backslash T_0.FS()] \backslash [T_0 \backslash T_0.FS()].FS()\} \backslash \\ & \backslash \{[T_0 \backslash T_0.FS()] \backslash [T_0 \backslash T_0.FS()].FS()\}.FS()\}.FS(). \end{aligned} \tag{3.6}$$

Another important property of this approach besides compactness is its scalability with respect to the size of a partitioning tree. For example, adding one level of leaf nodes to the bottom of a binary partitioning tree can double the total number of its nodes. Nevertheless, in order to represent all the new nodes (i.e. the new sub-partitions) we only need to add one character to the binary 0/1-string.

### Heterogeneous Partitioning Trees

We have shown above how the 0/1-string-based notation simplifies identifying the multimedia segments obtained with multimedia partitioning trees. Although this approach is characterized by compactness of notation and a good scalability with respect to the size of a tree, 0/1-strings are only suitable for homogenous partitioning trees, which are constructed using a unique partitioning function. While homogenous partitionings represent an important class of partitioning trees, nevertheless, as we have mentioned above, one of the main advantages of our binary tree partitioning approach is its suitability for heterogeneous partitionings, where different partitioning methods can be used within the same tree.

Just like homogenous partitioning trees, all the more heterogeneous binary partitioning trees can become considerably big in size. Hence, referring to particular nodes of partitioning trees using their underlying algebraic expressions can often become complicated. In order to facilitate referencing the nodes of heterogeneous partitioning trees, we propose using a generalized string-based approach similar to the 0/1-string approach described above. Although based on the same principle as the 0/1-string approach, i.e. using symbols to represent paths in binary partitioning trees, the main difference of the generalized string approach is the bigger size of the

alphabet used to construct path strings. In fact, the number of characters (symbols) of the alphabet sufficient to represent any node element of a heterogeneous binary partitioning tree is equal to:

$$N(A)=2\cdot S, \tag{3.7}$$

where $S$ - is the number of various unique segmentation methods used within a tree. Thus, in the particular case of $S$=1 (i.e. homogenous partitioning trees) only 2 symbols (e.g. "0" and "1" in our case) are sufficient to represent any node of the tree.

As it is easy to notice from the expression 3.7, the size $N(A)$ of the alphabet does not depend on the size of the partitioning tree. This particular property is one of the main advantages of the string-based path notation approach, since it provides for a good scalability of the algorithm with regards to the number of tree nodes. Just like in the case of homogenous partitioning trees (see above), also in the case of heterogeneous partitionings, adding one complete level of nodes to the bottom of a tree, which can double the total number of tree nodes, requires increasing the maximum length of a path string by only one symbol. This last statement in its turn displays another important property of the path-string notation approach, namely its scalability with respect to the number $S$ of partitioning methods used within a tree. Firstly, this scalability implies that the size of the path string does not depend on the partitioning complexity level (i.e. the number $S$ of different partitioning methods used throughout the path). Putting it another way, the size of the path string depends solely on the number of tree branches that make up the path, no matter the semantics of the tree nodes concerned. Secondly, and more generally, the path-strings remain invariant with respect to their tree size evolution, i.e. adding new nodes to the tree, whether using new or already existing partitioning methods.

In order to devise a suitable notation technique for paths in heterogeneous partitioning trees, we are going to base ourselves on the triplet approach used for homogeneous partitions (see expression 3.4 above). The major difference introduced by heterogeneous trees as compared to homogeneous ones is the multitude as well as non-static nature of the number $S$ of various unique partitioning methods used within a single tree. As a result, two different notation mechanisms can be proposed.

1. The first notation approach extends the triplet-based approach of homogenous trees and allows taking into account any arbitrary (however constant) number of partitioning methods. This approach provides a good level of notation compactness, however is only suitable for partitioning trees with a pre-defined constant set of partitioning methods. The n-tuple notations obtained in this manner are presented in the expression 3.8 below:

$$\langle M, PM_1(), PM_2(), \ldots, PM_n(), path\_string\rangle, \tag{3.8}$$

After specifying the multimedia element $M$ corresponding to the tree root (i.e. the multimedia element that is the subject of partitioning), all unique partitioning methods $PM_i()$ used within the tree are listed. Finally, the last element of the n-tuple in the expression 3.8 is the path string expression itself, which consists of a sequence of characters (symbols). To preserve the brevity of notation we propose to use the following alphabet:

$$\{p_1, \overline{p_1}, \ldots, p_N, \overline{p_N}\},$$

where $p_k$ denotes a node obtained by applying the partitioning method $PM_k()$ to the previous node of the tree, i.e.:

$$\text{`}p_k\text{'} \mapsto N_i{=}PM_k(N_{i-1})$$

and $\overline{p_k}$ denotes a node obtained as the remainder of applying the partitioning method $PM_k()$ to the previous node of the tree, i.e.:

$$\text{`}\overline{p_k}\text{'} \mapsto N_i{=}N_{i-1}\backslash PM_k(N_{i-1}).$$

For example, to represent the third word of the second sentence in a text block shown in fig. 3.28 the following string-based notation tuple can be used:

$$W_{23}{=}\langle T_0, Text.FS(), Text.FW(), \text{`}\overline{p_1}p_1\overline{p_2p_2}p_2\text{'}\rangle, \tag{3.9}$$

which is much more readable and compact than the underlying algebraic expression for $W_{23}$.

2. The second notation approach provides a less compact notation compared to the first one. However, it is particularly suitable for highly dynamic environments, where the set of partitioning methods used within a tree is not a priori known and is prone to evolve. Due to this peculiarity and unlike the first notation approach described above, pre-constructing an alphabet of characters used to compose tree path strings becomes hampered. Moreover, constructing a preliminary alphabet and trying to extend it later as needed would require rewriting the already existing path strings. For this reason we propose using a dynamically constructible alphabet with symbols that are simply equal to the names of partitioning methods. Using this approach, any subset of a multimedia element can be described as a path in a heterogeneous partitioning tree, which can be expressed by a following pair:

$$\langle M, path\_string\rangle, \tag{3.10}$$

where $M$ is the multimedia element corresponding to the tree root, and the *path_string* is composed of a sequence of symbols from the following dynamic alphabet:

$$\{PM_1, \overline{PM_1}, \ldots, PM_k, \overline{PM_k}, \ldots\},$$

where $PM_k$ denotes a node obtained by applying the partitioning method $PM_k()$ to the previous node of the tree, i.e.:

$$`PM_k' \mapsto N_i{=}PM_k(N_{i-1}),$$

and $\overline{PM_k}$ denotes a node obtained as the remainder of applying the partitioning method $PM_k()$ to the previous node of the tree, i.e.:

$$`\overline{PM_k}' \mapsto N_i{=}N_{i-1}\backslash PM_k(N_{i-1}).$$

Using this second notation, the expression 3.9 for $W_{23}$ can be rewritten as follows:

$$W_{23}{=}\langle T_0, `\overline{Text.FS}; Text.FS; \overline{Text.FW}; \overline{Text.FW}; Text.FW'\rangle.$$

Last but not least, it is important to note that despite limiting ourselves to only binary trees, our multimedia element partitioning approach still stays generic and allows including not only partitioning method functions, but also partitioning methods with any number of outputs, which is possible due to a well-known one-to-one mapping algorithm between general ordered trees and binary trees (the filial-heir chain algorithm).

### 3.5.5 Object-Level Notation for Multimedia Partitions

As it has been mentioned above, partitions of multimedia data values are themselves multimedia data values. For example, the third word of a second sentence of a text value is also a text value. Indeed, partitions are formally defined as subsets of multimedia data values, and from the point of view of the set theory subsets of sets are also sets in their own right.

In order to provide a mechanism of formally describing multimedia data values obtained by partitioning, we provide here an object-level notation for multimedia data partitions. This notation is, in particular, useful when comparing partitions of different multimedia data objects participating into a complex multimedia representational relationship.

While the full algebraic notation like the one presented in expression 3.6 represents a formal way of describing multimedia partitions, this form of notation is not suitable to use at the object level, while it is too heavy and it references the initial (root) object of the partition multiple times. To cope with this problem, we propose to extend a well-known dot-based notation, which we already use to represent the result of applying a method function to a multimedia data object (see e.g. $T_0.FS()$ in expression 3.6), with an additional operator $\odot$, which signifies the remainder of applying a method function that directly follows the $\odot$ sign, for example:

$$T_0{\odot}FS() = T_0\backslash T_0.FS() \quad .$$

Using the notation based on . and $\odot$ symbols, the third word of the second sentence of a text block in fig. 3.28 is represented as follows:

$$W_{23} = T_0 \odot FS().FS() \odot FW() \odot FW().FW() \quad .$$

The proposed . and $\odot$ based notation is compact and is suitable for the object-level representation, since is does not use algebraic set-oriented operations like \, and the initial object only appears once at the beginning of the expression, which, in particular, allows to easily determine the initial multimedia object that is being partitioned. The proposed notation allows to represent homogeneous as well as heterogeneous partitions. It is important to notice that the object-level . and $\odot$ based notation easily transforms into its underlying algebraic set-based representation by sequentially parsing the object-level expression and iteratively transforming every . and $\odot$ into the corresponding algebraic set-based expression.

### 3.5.6 Section Summary

In this section we have presented a formal methodology for conceptual-level partitioning of abstractly defined multimedia data. The proposed methodology can be qualified as generic, since it is independent of any particular partitioning criteria.

The set-based partitioning of abstractly defined multimedia data provides a basis for defining complex custom multimedia representational relationship types as described in the sect. 3.4.3.

In the next section we introduce another important multimedia conceptual modeling aspect, which is that of multirepresentation in multimedia.

## 3.6 Multirepresentation of Multimedia Data in MADS

In the previous sections we have addressed different modeling aspects pertinent to the new multimedia modeling dimension of MADS. In this section we introduce another important multimedia conceptual modeling aspect, which is that of multi-representation in multimedia.

### 3.6.1 Multimedia and Multirepresentation

One of the most important requirements for a powerful modern conceptual database model is the multi-representation support. Even though a database provides information about the selfsame domain of discourse, its perceptions by various applications using the database can and probably will be different [SVPZ99]. Due to diversity of application requirements, various design goals, and heterogeneous

user preferences, providing a single unified view of the information pertaining to the domain of discourse is barely sufficient and often is simply impossible. For the reasons we have just cited, the selfsame information might need to be represented within a database in several different forms (different representations), with each representation corresponding to some particular viewpoint of a user or designer of the system. Nevertheless, it is important to note that despite possible divergences between perceptions, they still represent the same underlying facts, and therefore one of the primary challenges in a multi-representation database system is the reconciliation of heterogeneous representations of the information within a single database.

Similarly to classic alpha-numerical as well as geographic/spatio-temporal data, the multimedia information can also benefit a lot from multirepresentation features. Indeed, multi-representation allows capturing various multimedia descriptions of the same real-world phenomena. The existence of these multiple representations can basically be caused by:

- Multiple application requirements.

  The same multimedia-enhanced database can be used by a variety of applications possibly belonging to different application domains or branches.

  For example, in a multimedia-enhanced hospital patient database, a patient record may contain a passport photo to be used by the hospital administration department, as well as a variety of medical imagery like X-ray scans, ultrasonic images, etc. to be used by hospital's medical personnel. The database system must thus be able to accommodate both sorts of multimedia information and to present either of them depending on the type of the application that queries the data.

- Technical requirements and limitations.

  Different depictions of the same multimedia information across various representations may in particular be governed by diverse limitations of technical nature.

  A representative example of such scenario is a content database supporting a web-based application that is available in a regular version for broadband access and a simplified lower-end version for mobile access. Designed to be accessible in mobile environments, which are generally distinguished by modest hardware characteristics, the multimedia information provided by the latter version of the application would generally be available in a poorer resolution, be represented by less resource consuming datatypes (e.g. still images instead of video clips), or simply be absent as compared with the regular version of the web application.

- Diverse personal preferences of the users of the system.

Due to various aesthetic, ergonomic, and other individual considerations various application users might prefer the same multimedia information to be represented in different ways.

Although personal user preferences would govern in the first place the presentational aspects of multimedia information, i.e. the way that the multimedia data is rendered by the output devices (e.g. screen alignment, sound settings, etc.) depending on the end user querying the system, they can also define which various multimedia information portraying the same real-world events needs to be present in the database in order to satisfy the "tastes" of various users of the system. Thus, for example, in a faculty meeting database 3 various multimedia representations of each meeting corresponding to 3 different personal user profiles are available: a photo & audio representation, a panorama-wide common video representation, and a compound video representation composed of individual videos of meeting participants.

Comparing this multi-representation rationale with the other two rationales mentioned above, it is important to note that even though personal user preferences are generally independent of the limitations imposed by application requirements and limitations related to technical characteristics of the system (see above), the latter can still serve as an effective upper boundary for the representations corresponding to personal user profiles. For instance, referring to the above example about a web-based application accessible in regular and mobile environments, it is quite obvious that mobile users of the system would most probably prefer to get at least the same full-quality audio-visual presentations as the desktop users, however these are the technical limitations of the system that hamper this.

Multi-representation in databases constitutes a general problem of conceptual database design, and, obviously, it is not uniquely inherent to multimedia applications.

The regular approach to providing multi-representation support in database applications is by using view-like mechanisms. Nonetheless, view-like approaches, e.g. relational views in RDBMS, IS-A hierarchies in object-oriented databases, or XSL-transformations in XML databases, only provide a very simplistic solution of the multi-representation problem [PSZ06]. The general idea of a view-like approach is to derive new representations of information from a predefined set of initial representations. Putting it another way, views provide a centralized approach, where auxiliary representations are derived from a core master representation. For example, in relational databases a table Employee containing information about all company employees independently of their rank would serve as a central master representation of information about the company staff. Based on this master representation additional underlying representations could be introduced to provide e.g. an insight into the hierarchical organisation of the company. These additional representations would be based on a series of relational views regrouping, for

instance, the executive management of the company, the middle-layer management, and non-managing staff. However, the information provided by these views would not introduce any new information that is not yet found in the database. Instead, they would simply represent subsets of data already available in the Employee table based on some restraining criteria provided by the views' `WHERE` clauses. Furthermore, the view mechanism on its own does not yet provide the way to clearly separate representations the one from the other, and to do this resorting to an access rights mechanism would be required.

It should be noted that the centralized nature of the view mechanism makes this approach even more improper in the case of multimedia-enhanced application environments. In sect. 2.4 two different classes of multimedia-enabled application were introduced: multimedia-centric applications adhering to the *multimedia as data* viewpoint, and multimedia-enhanced applications following the *multimedia as metadata* approach. While *multimedia as data* approach suits well for the types of applications where multimedia recordings themselves constitute the domain of discourse (e.g. digital museums or photo galleries, on-line digital music stores, etc.), the *multimedia as metadata* approach represents the class of multimedia-enhanced applications, i.e. application where, independently of the domain of discourse, multimedia data is used along with the other kinds of data (classical alpha-numerical data, XML, geographical data, etc.) to provide information pertaining to the application domain. In the latter types of multimedia applications (i.e. multimedia-enhanced applications) two distinct pieces of multimedia information describing the same object/event of the universe of discourse however belonging to two different representations are generally mutually independent. This means that a priori there does not exist any automatic mapping capable of deriving multimedia data in one representation from the corresponding multimedia data in another representation. For the reason given above, the view-based mechanism cannot be used in multimedia-enhanced applications that adhere to the *multimedia as metadata* viewpoint.

### 3.6.2 Multirepresentation in a Multimedia-Enhanced MADS Schema

In order to fully and without compromising the other modeling aspects satisfy the multimedia multirepresentation modeling requirements described in the previous subsection, we resort to multirepresentation capabilities provided in MADS (see sect. 3.2.5).

Following the orthogonality principle of MADS, the multirepresentation in multimedia data can be manifested at the entity type level, at the relationship type level, and at the attribute level. Both the stamping mechanism and inter-representation relationship types can be used to introduce multirepresentation in a multimedia-enhanced MADS conceptual schema.

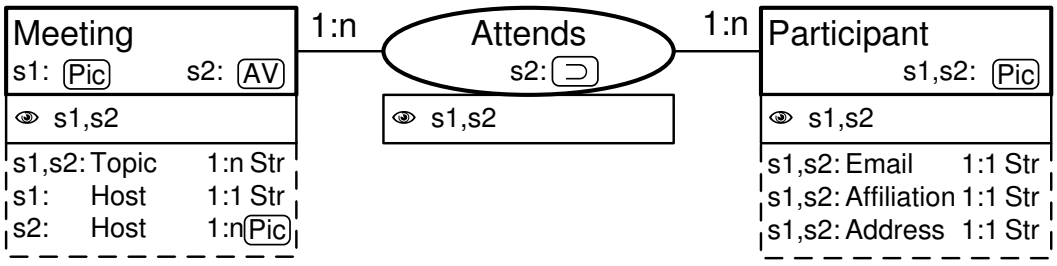



**Figure 3.30:** A fragment of a multimedia MADS schema with stamping.

In fig. 3.30 a sample fragment of a multimedia meeting application schema is shown. The schema is enhanced with multirepresentation semantics by allowing to differentiate between two different visions of the application domain by means of the stamping mechanism. The first vision of the application domain, which is that of meeting organizers, is characterized by the meeting still being planned and its multimedia recordings not yet being available. We associate with this vision the stamp $s_1$. The second vision of the application domain is that of the meeting archive maintenance team, which is above all interested in multimedia recordings available after the meeting. We associate with this second vision the stamp $s_2$. Taking advantage of the multirepresentation support, we decide to describe the multimedia semantics of a `Meeting` with a picture of its meeting room (already achievable in $s_1$), and to provide instead a full-length audio-visual report of the meeting as soon as it becomes available (obviously, only accessible in $s_2$). In order to exercise some control on the audio-visual representation of the object type `Meeting` in the perception $s_2$, we prescribe that all the meeting participants must be seen in the audio-visual footage in question. This is achieved by constraining the relationship type `Attends` with a representational relationship of type `MultimediaInclusion`. Finally, instead of storing the information about `Meeting.Host` as a simple character string value, we rather opt for a set of up-to-date photos taken right on the spot. Certainly, these latter are only available in $s_2$.

Obviously, instead of building a single object type that contains all of its representations, as in fig. 3.30, we could alternatively use inter-representation relationship types to provide multirepresentation multimedia modeling capabilities, just as it is the case for spatio-temporal or other kinds of data in MADS. For example, the fig. 3.31 illustrates an alternative notation for multiple representations of the entity type `Meeting` from the fig. 3.30 using an inter-representation relationship type `Corresponds`.

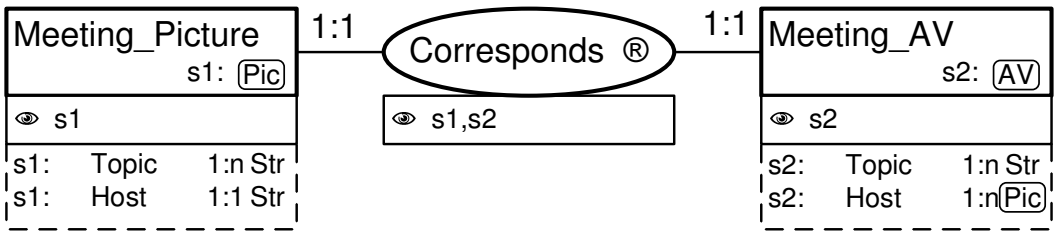



**Figure 3.31:** Inter-representation relationships in a multimedia MADS schema.

### 3.6.3 The Role of Multirepresentation in Semantic Multimedia Zooming

In the subsections above we have discussed how the multimedia data can benefit from the multirepresentation support in a similar way as the other types of data. We have also demonstrated some of the peculiarities of dealing with multirepresentation in multimedia-enhanced databases. In this subsection we describe another interesting application of the multirepresentation as applied to multimedia data, namely the semantic zooming.

As we have mentioned in the sect. 3.2.5, in the area of geographical and spatio-temporal applications the multi-representation in MADS provides a natural way of representing cartographic maps at different scales [PSZ06]. In this connection, representations corresponding to different map scales can be seen as zoomed views on the domain of discourse. We believe that adopting a similar approach when dealing with multimedia information in MADS can be beneficial for automatically deriving and controlling zoomed views of multimedia data.

Introducing multi-representation support into multimedia-enhanced applications allows providing multiple multimedia depictions of the same underlying objects and events belonging to the universe of discourse. As described in sect. 3.6.1, the multitude of possible multimedia representations can be due to a number of factors, which can basically be grouped in three categories: multiple application requirements, technical requirements and limitations, and personal user preferences. While in the general case the multimedia depictions of the selfsame objects/events in various representations are mutually independent, rather often there is a need to provide a series of multimedia representations of the object/event, which altogether represent a zooming sequence, i.e. one multimedia representation in a sequence is a zoomed version of another one.

This kind of requirement is especially common in visual media applications (i.e. still images, videos, graphical presentations, etc.), where the notion of zoom has initially emerged in the context of image-capturing equipment and was related to the optical resolution of camera lenses. Later on, zooming also became peculiar to digital imaging, where it related in particular to the resolution of digital image cap-

**Figure 3.32:** Semantic-pervasive zooming.

turing and rendering devices (e.g. image sensors and computer displays). Although nowadays zooming pertains primarily to the imaging media domain, this is mostly due to a sufficient technical advancement in this field of multimedia, and it does not imply zooming being exclusively a technical requirement and solely an imaging feature. First and foremost, zooming represents a user requirement and can be expressed by user preferences. The idea of zooming can have reference to any type of multimedia information whatsoever. For instance, for textual data, de-zooming could allow obtaining semantically more condensed abstract version of the text.

Although zooming could be sometimes perceived as a semantically neutral operation (e.g. replacing a digital image by a lower-resolution counterpart while still preserving the general visual quality of the picture), zoom operations in general can be characterized as semantically pervasive. While with the conventional zoom objects only change their size, with the semantic zoom they can additionally change shape, details, or even their very presence in the display [Bou03].

Let's consider an example of image zooming shown in fig. 3.32. As one can notice, a bird seen on the picture on the left cannot be seen anymore on the picture on the right (see fig. 3.32) because its size on the picture becomes too small after a de-zooming operation. Generally speaking, zooming-out (de-zooming) can lead to losses in the level of detail due to a decreased resolution, while zooming-in (zooming) can lead to losses induced by image border cropping (even though the level of detail of the rest of the image increases). However, besides purely visual (perceptional) changes, zooming can also change the semantic contents of multimedia data, which can, in its turn, imply changes in the corresponding database schema for multimedia-oriented applications, or simply make the data incompliant with the existing database schema for multimedia-enhanced applications. In this connection, if a picture on the left in fig. 3.32 was used as a multimedia representation for an entity of type `FlyingBird`, then the image on the right in fig. 3.32 would not be anymore eligible as an entity of this type.

Another conceptual inconsistency with the database schema, which could arise from zooming is the multimedia-related referential integrity. In the sect. 3.4 we

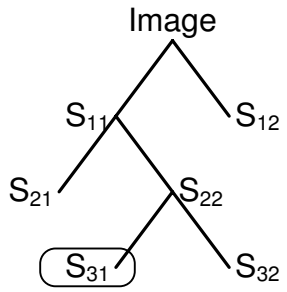



**Figure 3.33:** A partitioning tree-based integrity constraint.

have introduced multimedia representational relationships, which allow enriching a conceptual database schema with additional constraints of multimedia nature. We have further introduced four basic multimedia relationship types (intersection, inclusion, equality, and disjointness), as well as described the methodology for introducing custom representational relationship types based on separating the multimedia elements into sub-partitions. Due to semantic losses caused by zooming operations, a multimedia representational relationship may also become invalidated, as the multimedia sub-partitioning condition defining the relationship may become invalid.

The semantic-pervasive zooming problematic described above brings up several interesting issues.

Firstly, in order to hamper situations like the one presented in fig. 3.32, an integrity constraint mechanism that would check the compliance of the multimedia information to be stored in the database with the conceptual schema should be provided.

Secondly, it could reveal beneficial to provide an automated mechanism to help derive zoomed versions of a multimedia database by providing an individual stamped representation for each zoomed view. In this connection, each representation could provide a modified view of the database schema, which would take into account the changes in representational relationships due to zoom-related semantic losses.

For both of the issues mentioned above we propose to use an integrity-constraint-based approach relying on tree partitionings of multimedia elements. For example, to provide a multimedia integrity constraint checking for the case of fig. 3.32, a partitioning tree like the one presented in fig. 3.33 could be used to check the conformity of a newly introduced image of a bird with the database conceptual schema. Traversing the partitioning tree represented in fig. 3.33, an image to be checked is first split into the part representing the sky ($S_{11}$) as well as the lower part of the image ($S_{12}$). In the case of a pixel bitmap representation of the image, the segmentation criteria used here could be a mix of color-based and spatial segmentation.

The subset $S_{11}$ is further split into the sky part itself ($S_{21}$) and all the smaller-scale objects in the sky ($S_{22}$). At this step a color-noise filtering-based segmentation could be used. Finally, by running a contour-based shape analyzer on the objects from $S_{22}$ we obtain the subset $S_{31}$ corresponding to the part of the image that represents a bird. Note that in the case of an annotation-based representation of the image, a similar tree-partitioning approach could be adopted, gradually partitioning the set of image annotations according to some criteria like e.g. image-bound spatial anchors of the annotation tags, etc.

The separation tree shown in fig. 3.33 can be used as an integrity constraint condition that provides a sort of a filter, which determines if an image corresponds to the database schema or not. Quite obviously, if the parsing sequence $S_{11}$-$S_{22}$-$S_{31}$ were applied to the de-zoomed picture shown at the right in fig. 3.32, the result of this segmentation would be NULL, which would indicate that the picture in question is not a valid candidate value for an entity type `FlyingBird`. Since multimedia representational relationships are also based on segmentation of multimedia elements, an approach similar to the partitioning tree approach presented above can also be used to enforce the multimedia representational relationships integrity.

### 3.6.4 Section Summary

In this section we have discussed the multirepresentation support in the new multimedia modeling dimension of MADS. Similarly to other kinds of data, multimedia data can also benefit from the multirepresentation support in order to deal with multiple application requirements, multiple technical requirements and limitations, as well as diverse personal user preferences.

Another interesting application of the multimedia multirepresentation support is in the field of semantic multimedia zooming, where multirepresentation provides a powerful solution to representing semantically pervasive multimedia zooming operations.

## 3.7 Chapter Summary

In this chapter we have introduced a novel multimedia-enhanced conceptual model, which allows to efficiently consider not only the classical *multimedia as data* representation, but also the *multimedia as metadata* representation, making the model suitable for various kinds of multimedia applications.

The model that we have proposed is based on MADS (Modeling Application Data with Spatio-temporal features), which is a powerful conceptual model in particular characterized by structural completeness, spatio-temporal modeling capabilities, and multi-representation support (see sect. 3.2). Applying the or-

thogonality principle of MADS, we have developed a new multimedia extension of MADS conceptual model, which builds upon its existing modeling features for non-multimedia data.

The basis of the proposed multimedia modeling extension is the multimedia datatype hierarchy, which allows to efficiently classify and manage various sensorial and perceptional varieties of multimedia data (see sect. 3.3.1). Multimedia datatypes can be used to characterize attributes, entity types, and relationship types in MADS. It is important to note that the conceptual-level definition of the essence of multimedia datatypes in MADS has been provided using abstract concepts of the set theory. The abstract definition allows us to deal with potentially very different representations of the same datatypes, like, for example, pixel-based or annotation-based representations of pictures.

Another modeling aspect that has been proposed is multimedia representational relationships, which provide a mechanism of introducing multimedia-related constraints into a MADS multimedia schema. In sect. 3.4.2 and sect. 3.4.4 several examples of implementing abstract set-based representational relationships in different application environments have been provided.

The multimedia representational relationships rely on a principle of conceptual set-based partitioning of multimedia data (see sect. 3.5), which we extensively use throughout this chapter.

Last but not least, in the sect. 3.6 we discuss the benefits of the multirepresentation support in the new multimedia modeling dimension of MADS, as well as peculiarities of semantic multimedia zooming.

# Chapter 4

# Logical Multimedia Modeling

In chapter 3 we have introduced a novel multimedia-enhanced conceptual model based on MADS (Modeling Application Data with Spatio-temporal features). Despite the rich modeling capabilities offered by the multimedia-enhanced MADS, a conceptual model alone cannot cover all the aspects of multimedia data. Indeed, according to the conceptualization principle of conceptual modeling [vG82], the multimedia-enhanced conceptual model should only take into account high-level semantic facts relating to the audio-visual depiction of the universe of discourse, and should not be affected by any multimedia implementation features. Those latter, in their turn, should be taken care of at logical and physical design stages.

Although the main focus of our work is the conceptual modeling of multimedia data, in order to complete the picture we discuss in this chapter the peculiarities of logical multimedia modeling and of conceptual-to-logical transformations.

## 4.1 Multimedia Document Models

A logical multimedia model determines the logical organization of multimedia data. From the sensorial point of view, defining a logical multimedia model corresponds to defining *complex multimedia documents*, which represent the composition of media elements of various types into logically coherent multimedia units [JYZ04]. A powerful multimedia document model must generally consider two main aspects of logical multimedia data organization: the compositional aspect, and the presentational aspect [BK01].

The *compositional aspect* of a multimedia document model generally addresses the composition of different media elements of various types into a logical multimedia unit of the model. From the compositional point of view, at least three central subaspects have to be emphasized:

- A *temporal model*, which describes temporal organization of the elements

within a multimedia document. For instance, from the temporal point of view, a multimedia document can be seen as a sound clip being played back in parallel with a group of two video clips, which are, in their turn, being played sequentially one after another. Various temporal models like point-based, interval-based [All83], event-based, etc. can be used.

- A *spatial model*, which describes spatial organization of media elements within a multimedia document. It addresses issues like position of visual media elements on the screen or window, as well as their mutual layout (e.g. overlapping, adjacent, etc.). In general, three different approaches are used: absolute positioning, directional relations, and topological relations [EF91].

- A *user interaction* framework, which allows users to influence the way a presentation of the multimedia document proceeds by choosing from different presentation paths. User interaction is tightly related to the concept of *hypermedia*, which was first introduced by Ted Nelson as early as in 1965 [Nel65]. Although not necessarily available in all multimedia document models, user interaction is believed to be the preeminent aspect differentiating multimedia documents from regular linear multimedia recordings (e.g. a picture or a video clip). In particular, interaction models allow users to control the presentation playback (e.g. rewind, search, repeat, etc.), or change the presentation environment characteristics like zoom, playback speed, etc. One of most widely-supported types of interaction in multimedia document models is navigational interaction, which uses nonlinear hyperlinks to select from a variety of possible presentation paths.

The *presentational aspect* is the second main aspect of multimedia document modeling. The presentational aspect should generally be independent of the compositional aspect, which is to a great extent conditioned by reusability and user adaptation of a document model [BK01]. Indeed, separating the structure (composition) of a multimedia document from its layout keeps the structure of the document independent of such presentation-related issues as screen size, volume level, resolution, etc., which in its turn greatly simplifies reusing the document as a building block for composing other multimedia documents. Separating the structure from the presentation also allows to better account for a variety of user preferences, since a document structure that is not influenced by any particular end-device layout details is more generic and hence easier to adapt to a broader range of end-visualization user requirements. Moreover, the independence of the compositional aspect from the presentational aspect permits to use different models in different situations, which, in its turn, allows combining strengths of various models to better accommodate to either compositional or presentational environments.

It would be important to note that although spatial characteristics are present in both compositional and presentational aspects of multimedia documents, they do not represent the same characteristics of multimedia documents. The compositional aspect of multimedia document models deals with semantic spatial-related issues of multimedia information no matter the presentation of the multimedia document on a particular end-user rendering device. For example, in a composite landscape picture the sky should be positioned above the land. This spatial constraint is purely semantic and does not depend on a particular way the landscape picture will be displayed on the screen. On the other hand, specifying the pixel coordinates where the landscape picture should appear on the screen of a particular user deals with a presentational aspect spatial characteristic.

We present hereby a brief state-of-the-art overview of existing logical multimedia models.

### 4.1.1 HyTime Hypermedia Model

Hypermedia/Time-based Structuring Language (HyTime) from ISO's WG8 has been developed as a standard for structured hypermedia interchange [Gol91]. The second edition of the standard was published in 1997.

HyTime is SGML[1]-based, and it defines a set of hypertext-oriented element types that, in effect, supplement SGML and allow SGML document authors to build multimedia documents in a standardized way.

In HyTime, a multimedia document is interpreted as "a collection of information that is identified as a unit and that is intended for human perception".

HyTime uses point-based temporal model and absolute-positioning-based spatial model. The standard allows for structural reuse of a whole document or document fragments, as well as of particular media elements and their parts. This powerful support for reuse is one of the major strengths of HyTime. Also, in HyTime, the documents are described at a high conceptual level, which allows for presentation-neutral modeling of multimedia document contents.

While the rigidity of the standard is one of its strong points, it is, at the same time, one of its primary limitations with regards to the HyTime's lack of extensibility and adaptation. Another important drawback is the lack of user-interaction capabilities.

[1]The Standard Generalized Markup Language (ISO 8879:1986 SGML) is an ISO Standard metalanguage, in which one can define markup languages for documents. The most famous derivatives of SGML are XML and HTML.

```
<smil xmlns="http://www.w3.org/2000/SMIL20/CR/Language">
    <head>
        <layout>
            <region id="imageRegion" left="10" .../>
        </layout>
    </head>
    <body>
        <par dur="120s">
            <img id="photo" src="myPhoto.jpg" region="imageRegion"/>
            <audio id="song" src="mySong.mp3" />
        </par>
    </body>
</smil>
```

**Figure 4.1:** A sample SMIL document.

### 4.1.2 SMIL Multimedia Document Model

SMIL (Synchronized Multimedia Integration Language) [W3Cb] [BR09] is a W3C Recommended XML markup language for describing multimedia presentations. It defines markup for timing, layout, animations, visual transitions, and media embedding, among other things. Similarly to a HTML document, a SMIL document is typically divided between an optional `<head>` section and a required `<body>` section. The `<head>` section contains layout and metadata information. The `<body>` section contains the timing information, and is generally composed of combinations of two main tags: parallel (`<par>`) and sequential (`<seq>`), which define respectively parallel (synchronous) and sequential (asynchronous) flow of media objects (streams) composing a SMIL presentation. The media objects within a SMIL document are referred to by URLs, allowing them to be shared between documents and stored externally. In contrast to HyTime, SMIL offers a rather comprehensive modeling of static adaptation to the technical infrastructure, as well as the navigational interaction.

Fig. 4.1 shows a simple SMIL document. The presentation contains an image and an audio stream, which are played in parallel. The media content of a SMIL document is structured by using so-called *time containers*. A time container (or operator) carries a particular temporal semantics that allows defining the temporal placement of media objects. These operators are `<seq>`, `<par>` and `<excl>` elements. A `<seq>` container, short for "sequence", defines a sequence of elements in which elements are played one after another. A `<par>` container, short for "parallel", defines a simple time grouping, in which multiple elements can playback at the same time. Finally, `<excl>` is a time container with semantics based upon `<par>`, but with the additional constraint that only one child element may play at any given

time.

### SMIL Modules and Profiles

In SMIL, a *module* is a collection of semantically-related XML elements, attributes, and attribute values that represents a unit of functionality. Modules are defined in coherent sets, meaning that the elements of a module are associated with the same XML namespace. A *language profile* is a combination of modules. Modules are atomic in that they cannot be subset when included in a language profile. Furthermore, a module specification may include a set of integration requirements, with which the language profiles that include the module must comply.

SMIL 2.0 provides a scalability framework, where a family of scalable SMIL profiles can be defined using subsets of the SMIL 2.0 language profile. A SMIL document can be authored conforming to a scalable SMIL profile such that it provides limited functionality on a resource-constrained device while allowing richer capabilities on more capable devices. In particular, SMIL 2.0 Basic is a profile that meets the needs of resource-constrained devices such as mobile phones and PDAs. The SMIL Basic profile provides the basis for defining scalable and interoperable SMIL profiles. A SMIL profile allows a SMIL user agent to implement only the needed subset of the SMIL 2.0 standard while maintaining document interoperability between device profiles built for different needs. A scalable profile enables user agents of a wide range of complexity to render, from a single and scalable SMIL document, customizable presentations adapted to the capabilities of the target devices. The advantages of scalable profiles are:

- Authors can re-purpose SMIL content targeting a wide range of devices that implement SMIL semantics.
- The rendering of the same content can be improved automatically as devices get more powerful.
- All SMIL 2.0 documents can share a document type, a schema, and a set of defined namespaces, and the required default *xmlns* declaration.
- Any future SMIL 2.0 extensions can easily be incorporated into SMIL documents and user clients.

A scalable profile is defined by extending the SMIL Basic profile, using the content control facilities to support application/device specific features via a namespace mechanism.

The SMIL 3.0 Language profile [Mul08] extends the SMIL 2.x Language profile with new functionalities introduced in SMIL 3.0 Modules.

```
...
<seq>
    <video src="video1.mpeg"/>
    <par>
        <a href=http://www.site.com/new_presentation.smil show="new">
            <video src="video2.mpeg" ...  />
        </a>
        <text src="text1.txt" ...  />
    </par>
</seq>
...
```

**Figure 4.2:** Hyper-linking example in SMIL.

### Adaptation Support in SMIL

The structure of SMIL documents is defined in a way that helps to adapt them to different contexts. The SMIL specification defines several mechanisms that enable the adaptation process. These mechanisms are: spatial document adaptation, hypermedia extensions, and content control, which we briefly describe below.

In SMIL 2.0, the spatial organization of the media objects is specified using `layout`, `root-layout` and `region` elements. The presentation layout and the content (media objects) of the multimedia document are separated. Therefore, layout modifications (adaptations) can be done without changing the content. Likewise, the content could be modified or adapted, without changing the spatial organization of a document. In SMIL, in order to link the content to a spatial organization, the `region` identifiers are used (see fig. 4.1).

SMIL specification also defines several mechanisms of interaction. One of the most important among them is hypermedia links, which help provide interaction for scene decomposition. Fig. 4.2 shows a sample SMIL document containing a hyperlink. When the user clicks on the video object "video2.mpeg", a new presentation window is started. This is done by providing links to media objects that, due to limited display size, cannot be displayed on the display together with other media objects.

Another SMIL 2.0 specification defines content control modules, which contain elements and attributes that provide for runtime content choices and optimized content delivery. SMIL provides a "test-attribute" mechanism to process an element only when certain conditions are met, for example when the language preference specified by the user matches that of a media object. One or more test attributes

```
...
<par>
    <video src="video.mpeg" ...  />
    <switch>
        <audio src="soundHQ.aiff" systemBitrate="56000" ...  />
        <audio src="soundMQ.aiff" systemBitrate="28800" ...  />
        <audio src="soundLQ.aiff" ...  />
    </switch>
</par>
...
```

**Figure 4.3:** Using the SMIL switch element.

may appear on media object elements or timing structure elements; if the attribute evaluates to *true*, the containing element is played, or is ignored otherwise.

SMIL also provides a *switch* element for expressing that some document parts are alternatives, and that the first one fulfilling certain conditions should be chosen. This feature is often used, for example, to let the client choose from different language versions of a SMIL document. Another common use is to let select the quality of an audio-visual presentation in order to comply with available network bandwidth. Fig. 4.3 illustrates the adaptation by alternative substitution in SMIL. In the multimedia document in fig. 4.3 one of audio objects is selected to accompany the video object. If the system bitrate is 56000 or higher, the object "soundHQ.aiff" is selected. If the system bitrate is at least 28800 but less than 56000, the object "soundMQ.aiff" is selected. If no other objects are selected, the alternative "soundLQ.aiff" is selected, since it has no test attribute (thus is always acceptable) and no other test attributes evaluate to true.

It is important to note that despite being a powerful multimedia document format, which has provided for its merited popularity, SMIL still represents first and foremost a multimedia presentation format. In spite of the above-discussed adaptation support in SMIL, the content of a multimedia document and its visualization aspects tightly interflow, which seriously hampers structural and semantic reusability of SMIL documents.

### 4.1.3 Madeus

Madeus [VRL00] [JRT01] is an authoring environment for multimedia documents, which incorporates an XML-based multimedia document model integrated with a GUI user interface (see fig. 4.4).

In Madeus the temporal relationships between document elements are based on

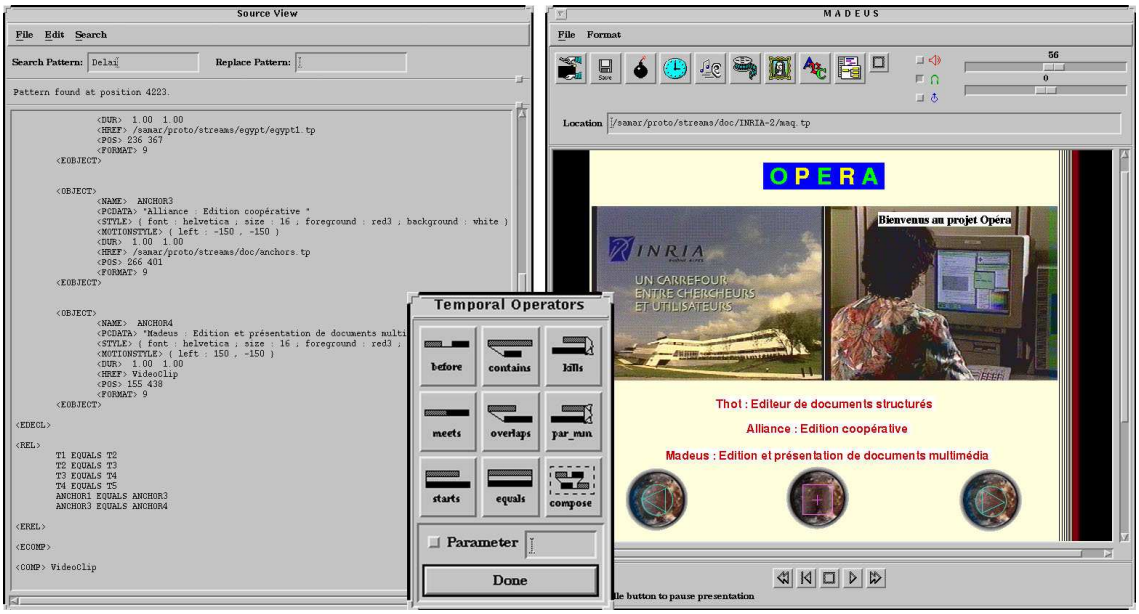



**Figure 4.4:** Madeus tool interface.

Allen relationships [All83] (interval-based temporal model). While the spatial model of Madeus is absolute-positioning-based, the model also provides some support for spatial relationships like *TopAlign*, *HorizontalCenter*, etc. Another distinctive feature of Madeus is a tight coupling of presentation services and editing services, which is achieved by using a constraint-based reasoning mechanism [JRT01]. Each time a user-defined relation is inserted into the document, it is translated into constraints upon the objects participating in the relationship. These constraints are then given to a reasoner, which checks the consistency of the overall resulting set of constraints.

It is important to note that although the Madeus model provides means to define a spatial layout within the document, there is no clear separation of layout and structure. In fact, transforming an existing Madeus document by means of XSLT into different presentations from the same source document is due to the nature of XML but not due to the Madeus model itself. Hence, the documents are only partially independent from their actual presentation. However, the high structuredness of the document specification provides a higher level compositional support than SMIL (see sect. 4.1.2). As compared to the HyTime model (see sect. 4.1.1), Madeus does support user-interaction, while also providing about the same good level of reuse as HyTime. Unfortunately, just like HyTime, Madeus does not offer document adaptation capabilities (user-adaptability, definition of alternatives, etc.).

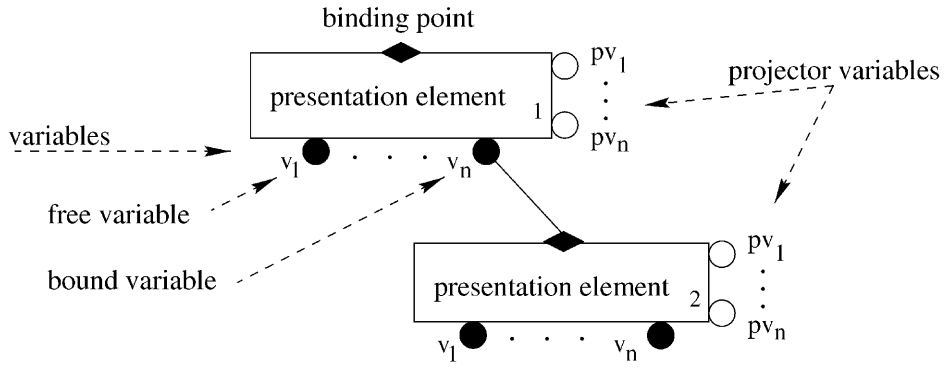



**Figure 4.5:** Graphical representation of the basic document elements.

### 4.1.4 $Z_YX$ Model

Taking into account the requirements towards a rich logical multimedia (document) model cited earlier in this chapter, as well as taking into consideration the shortcomings of some of the existing multimedia document models described above, S. Boll, W. Klas et al. have proposed an adaptable and reusable multimedia document model $Z_YX$ [BKW99], [BK01]. The $Z_YX$ model provides support for reuse of structure and layout of document fragments and for contextual adaptation of the content and its presentation. The model design aims to fulfill the three principal requirements: *reusability*, *adaptability* and *presentation-neutrality*.

In $Z_YX$ a multimedia document is described by means of a tree. The nodes of the tree are the *presentation elements* and the edges of the tree bind the presentation elements together in a hierarchy. Each presentation element has one binding point with which it can be bound to another presentation element. It also has one or more *variables* with which it can bind other presentation elements. Additionally, each presentation element can bind *projector variables* to specify the element's layout. Fig. 4.5 depicts the graphical representation of these basic elements.

Presentation elements can be media objects or elements that represent the temporal, spatial, layout, and interactive relationships between the media objects. Fig. 4.6 represents a document tree model.

The fragment starts with the presentation of the root element, whose binding points can bind fragments of other presentation elements into a more complex multimedia document tree. The root element is a sequential element *seq*, which binds the media objects *Image* and *Text* to its variables `v2` and `v4`, and a parallel element *par* to its variable `v3`, respectively. The *par* element synchronizes a *Video* and an *Audio*. Unbound variables `v1` and `v5` of the root element can be used later, e.g. to add a title at the very beginning of the multimedia document and a summary at the end of the document sequence.

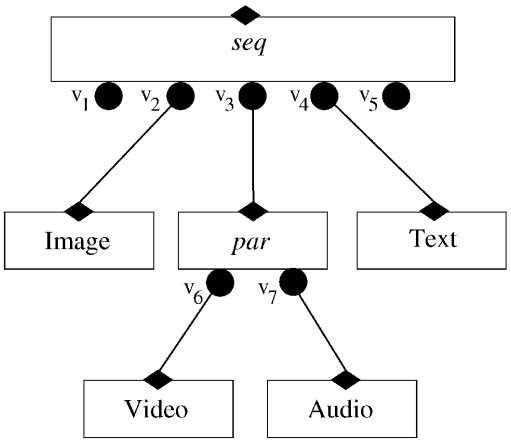



**Figure 4.6:** Simple document tree - a $Z_YX$ fragment.

$Z_YX$ model provides features that allow for reusing parts of media or document fragments. With regards to the granularity of reusability, the model supports two levels of reusability: reusability at the media level, and reusability at the fragment level. Reusability at the media level is assured by means of selector elements: *temporal-s* and *spatial-s* elements. The former helps the selection of a temporal part of a continuous media, while the latter supports the selection of a spatial area of visual media objects. Reusability at the fragment level is provided by the presence of free (unbound) variables, the encapsulation of a fragment into a complex media element, and the use of external media elements (fragments composed in other formats). With regards to the reusability type, the model supports identical and structural reuse.

The identical reuse can be realized by usage of selector elements (see above), while for implementation of the structural reusability the model provides the projector elements that influence the visual and audible layout. They define how a media object or fragment is presented, e.g. the presentation speed of a video, the spatial position of an image, etc. The use of projectors allows separating compositional and presentational aspects of multimedia documents in $Z_YX$. By means of projector elements, one can change the layout of a document, which allows for reusability of the same document structure with different presentation layouts.

Another inherent property of the $Z_YX$ model is adaptability. It is defined as the ability of the document to best match the context of the user that requests the document. To support this, both descriptions of the context and multimedia content that can be adapted to this context are needed. The model captures the context in a user profile, i.e. the metadata that describes the user's topics of interest, presentation system environment, and network connection characteristics.

The model provides two presentation elements for an adaptation of the document

to a user profile: the switch element and the query element. The switch element allows specifying different alternatives for a specific part of the document. Associated with each of the alternatives under a switch element, is metadata that describes the context in which this specific alternative is the best choice for presentation. This metadata is specified as a set of discriminating attribute-value pairs for each alternative. During the presentation, the user profile is evaluated against the metadata of the switch, and an alternative, whose discriminating attributes best match the current user profile, is selected for presentation. A switch element can be used only if all alternatives can be modeled at authoring time, i.e. before the presentation. Hence, the switch element implements the requirement for static adaptability of the model.

Non-static adaptability can be specified with a query element. By means of metadata, the query represents the fragment that is expected at this point in the presentation. When the document is selected for presentation, the query element is evaluated and the element is replaced by the fragment best matching the metadata given by the query element. The query element provides for dynamic adaptability of the model, since the evaluation of the query and the selection of the fragment take place just before the presentation.

Last but not least, $Z_YX$ complies with the requirement of presentation-neutrality. In [BK01], S. Boll et al. define that a multimedia content is presentation-neutral when the multimedia material is independent of the actual realization of a presentation for a particular client and under a certain context. The requirement of presentation-neutrality is strongly interrelated with the structural reusability.

The explicit separation of structure and layout allows for presentation-neutral representations. As outlined before, the variables of a presentation element need not to be bound in the first place, which also applies to the projector variables. It is possible to specify the presentation-neutral course of the presentation and, later on, bind the presentation-dependent layout once the document has been selected for presentation. From this point on, the presentation-neutral structure of the document is bound via projector variables to the presentation-dependent layout defined by a set of projectors.

### 4.1.5 Section Summary

In this section we have provided a state-of-the-art overview of logical multimedia document modeling. Several existing systems have been presented, and a set of basic universal requirements for multimedia document models has been listed.

Despite a wide adoption of the SMIL format (see sect. 4.1.2), there does not exist nowadays a predominant multimedia document model. In particular, substantial shortcomings of SMIL make place to other models, which take a more generalized approach to multimedia document modeling.

Considering the requirements towards a multimedia document model, we have stressed the importance of differentiating between the compositional and the presentational aspects of multimedia documents. The compositional aspect deals with the structural side of multimedia documents, namely formation of complex documents out of basic multimedia components and/or other multimedia documents, as well as temporal and spatial synchronization of the document components. On the other hand, the presentational aspect rather deals with rendering of multimedia documents, taking into account screen and sound settings of the document presentation to the end user, as well as, to a large extent, the notions of personalization and user-adaptability.

In the next section we discuss the peculiarities of conceptual-to-logical transformations.

## 4.2 Interconnecting the Multimedia Modeling Layers

In the previous section we have presented an overview of logical multimedia document modeling. As it has been shown, this area on its own represents a vast research domain, which has drawn a significant attention of the scientific community. Our state-of-the-art overview has revealed that despite a number of common characteristics peculiar to many of the existing approaches, there does not exist nowadays a predominant generic multimedia document model. Taking into account the above-mentioned, and willing to keep our approach as generic as possible, we have decided to stay independent of any particular logical multimedia document model and to possibly be able to deal with any model that satisfies the set of requirements discussed in the sect. 4.1.

### 4.2.1 Peculiarities of the Conceptual-to-Logical Mapping

In order to represent all various modeling facets of multimedia data, from its semantic representation to the underlying raw data storage and encoding, where each facet could be fully explored without the need to compromise one modeling aspect in favor of another, we need a compound integrated modeling framework, where each multimedia modeling aspect can be independently treated within its own layer (i.e. conceptual, logical, and physical). Because of the multitude of possible conceptual and logical representations of multimedia data, and taking into account the structural and semantic heterogeneity of the models used at various layers, interconnecting the multimedia models at different layers is a challenging task.

In order to better understand the peculiarities of multimedia inter-layer map-

pings, let's first consider a classical conceptual-to-logical mapping problem. In the classical database modeling, the designer generally chooses among a rather restricted set of models suitable for each of the design levels. For example, one could use an entity-relationship diagram at the conceptual level, a relational schema at the logical level, and a RDBMS-specific storage-explicit schema at the physical level. Although the various modeling techniques used at different stages of the modeling process are generally inter-independent across the modeling layers (e.g. one could use UML instead of the ER model, and still implement it within a relational database), there nevertheless exists a certain dependency between them. As a matter of fact, once the designer chooses a particular conceptual modeling technique he intends to use, he will most probably logically implement his conceptual schema using a logical model, which is the most common one for the conceptual model in question. Moreover, the methodologies of inter-layer mappings are also well-known, and the logical implementation of the conceptual schema will most probably be done automatically by one of the many CASE tools. Similarly, once the logical model is devised, its most probable physical implementation technique is known right after. Thus, for example, once a schema designer has devises a conceptual database model using ER diagrams, it will then most probably be logically implemented using a relational model, which will be afterwards implemented within a vendor-specific RDBMS (e.g. Oracle, MS SQL Server, etc.). Throughout the entire modeling process the designer will be assisted by a plethora of available CASE tools (e.g. CA Erwin, Oracle Designer, etc.), both during constructing the conceptual schema, and during the automatic inter-schema (inter-layer) transformation phases.

As compared to classical database modeling scenarios, our multimedia inter-layer mapping problem is characterized by a higher level of mutual cross-layer independence of the models used at different layers. As it has been stated above, using multimedia-enhanced MADS model at the conceptual layer does not influence the further choice of the logical model (e.g. SMIL, $Z_YX$, or other), leaving the designer the option of using whatever suitable model he prefers. It is important to note that this inter-layer independence property is far more than just a "nice-to-have" requirement, but is in fact brought about by the semantic essence of conceptual and logical multimedia models.

First of all, the logical multimedia document models have been conceived as stand-alone models in their own right, without aiming at providing a lower-level implementation of some initial higher-level conceptual model. As a matter of fact, it may be often required to store in a multimedia database a collection of multimedia documents, which have been constructed before the existence of the multimedia database and of the conceptual multimedia database schema. In that case, the existence and the structure of multimedia documents is governed not by the concepts expressed in a database schema, but rather by the availability of the underlying multimedia data sources (files), which existed during the creation of multimedia documents.

Furthermore, a conceptual multimedia-enhanced model like MADS and various logical multimedia document models are also semantically very different. Whereas a conceptual model aims at representing various concepts inherent to the universe of discourse of the application, and where some of the concepts are possibly enhanced with additional descriptions of multimedia nature, the modeling domain of logical multimedia models is the multimedia documents and, in particular, the structural composition of their constituent elements, which eventually influences the way a multimedia document is represented on the end-user visualization devices. Hence, the two types of models are potentially very different and possibly there would exist no direct correspondence between the two.

To illustrate the ideas presented above, let's consider the following examples.

In the case of annotation-based representation of the semantic essence of multimedia data, the multimedia-related constraints expressed in the conceptual model only deal with the semantic part of the multimedia data and do not concern its sensorial part. Thus, in example 2 in sect. 3.4.2 the conceptual model describes only the semantic aspect of employee photos (i.e. information provided via photo annotations tags), however no constraints on the sensorial (imagerial) aspect of the photos are imposed. In this particular case we could, for instance, compare multimedia objects of sensorially very different multimedia datatypes like `Image` and `Audio`, which would be hardly achievable in the case of pixel-bitmap-based representation of images and sound-wave-based representation of audio. While in the case of annotation-based conceptual representation of multimedia data we would be able to implement the conceptual multimedia constraints within a logical model for multimedia annotations (e.g. using Annotea or RDF format), we would certainly not be able to derive from such a conceptual model any meaningful guidelines for devising a logical multimedia document model (e.g. SMIL or $Z_YX$), which is due to a high level of discrepancy in interpreting the semantic essence of multimedia data at conceptual and logical levels.

Furthermore, even in the case of sensorial-inspired conceptual representation of multimedia data (e.g. pixel-bitmap-based representation of images or image-frame-based representation of videos), coming up with clear rules for conceptual-to-logical mapping is not a straightforward task. The example in fig. 4.7 shows a representation of the universe of discourse of a multimedia meetings application at conceptual and logical levels. A conceptual MADS schema in fig. 4.7 represents, in particular, the semantic multimedia knowledge pertaining to the universe of discourse: the entity types `Participant` and `Meeting` are characterized by multimedia extents of type `AudioVideo`, and they are linked by a relationship type `TakesPart`, which bears an additional multimedia representational constraint of type `MultimediaInclusion`. The logical multimedia representation of the universe of discourse in fig. 4.7 is given in the form of a sample $Z_YX$ document, which describes the compositional aspect of a logical multimedia document representating a particular meeting `Meeting`

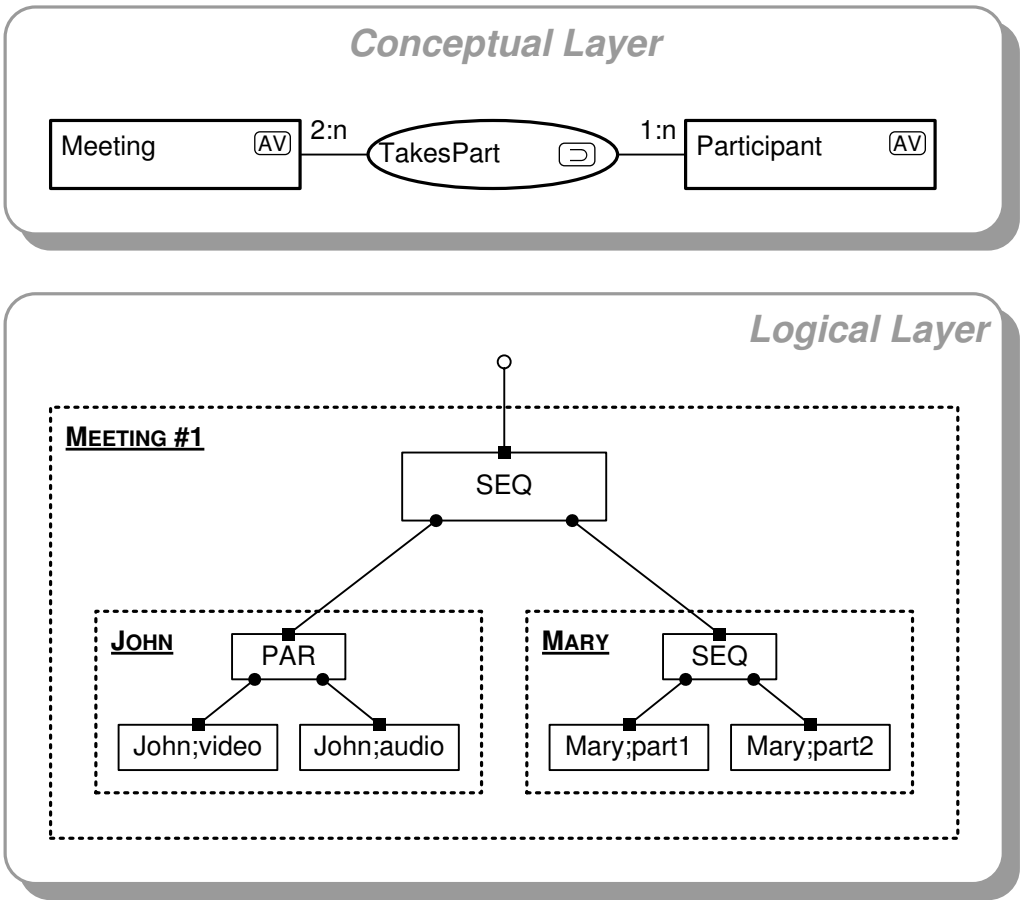



**Figure 4.7:** A sample multi-layer multimedia application schema.

`#1` and its participants `Mary` and `John` from the multimedia meeting application. Although the $Z_YX$ document in fig. 4.7 provides a perfectly valid logical multimedia representation of the application knowledge, the conceptual multimedia model alone would not have been sufficient to derive this logical multimedia representation. Thus, the fact that the $Z_YX$ representation of `Mary` is obtained as a sequential composition of two successively recorded video camera footages is influenced not by the conceptual multimedia model, but rather by the availability of underlying physical multimedia data sources. Furthermore, the exactly sequential composition of the $Z_YX$ representation of `Meeting #1` out of the $Z_YX$ representations of its participants is not the consequence of John and Mary participating in the Meeting #1 one after the other, but is simply due to the application user requirements for multimedia document visualization and playback. Finally, if dedicated audio-visual recordings of the Meeting #1 were made, then the $Z_YX$ representation of `Meeting #1` could be composed of these dedicated meeting recordings instead of being composed of the $Z_YX$ representations of its participants as it is the case in fig. 4.7. Obviously, such alternative logical multimedia document for `Meeting #1` would also constitute a perfectly legal logical multimedia representation of the application data.

In this subsection we have exposed the peculiarities of the multimedia inter-layer mapping problem. As it has been illustrated in the two examples above, contrary to classical conceptual-to-logical mapping approaches in traditional databases, the high level of semantic independence of conceptual and logical multimedia models in multimedia-enhanced systems may condition a significant semantic gap between the conceptual and logical models, which makes elaborating a precise set of inter-layer mapping rules a very challenging task.

### 4.2.2 General Conceptual-to-Logical Transformation Guidelines

As discussed above, the conceptual-to-logical mapping and transformation approach, which we adopt, builds upon the assumption of mutual independence of the models used at different modeling layers, which is first of all due to a high level of semantic heterogeneity of conceptual and logical multimedia models. While on one hand this allows, in particular, to integrate into a multi-layer system any suitable multimedia document model, it also conditions an important semantic gap between the conceptual model and the logical model, which substantially complicates the inter-layer mapping.

For the reasons cited above, instead of seeking to provide some restricted rigid set of inter-layer mapping rules, our alternative idea consists in providing a set of mapping recommendations or guidelines for the multimedia schema designer. These guidelines are meant to help the schema designer to come up with such a logical representation of multimedia data that follows the conceptual representation thereof no matter the particular logical multimedia model that the designer chooses.

Nevertheless, due to the facultative nature of the recommendations, the schema designer has the right to reject the proposed mapping guidelines and to come up with different conceptual-to-logical mappings. In the latter case it is then up to the schema designer to check the validity of the employed inter-layer mappings and to make sure that the logical representation of multimedia information adheres to its conceptual representation form.

As presented in sect. 3, the main modeling elements of the MADS conceptual modeling dimension are: multimedia datatypes (both simple and complex), simple multimedia representational relationships, multimedia partitioning mechanisms and complex multimedia relationships, and multirepresentation of multimedia. A possible logical presentations of these modeling concepts greatly depends first of all on the particular conceptual interpretation of the multimedia semantics, which, as we have shown in the examples in sect. 3.4.2 and sect. 3.4.4, can be very different.

Let's set out the general guidelines of how these basic multimedia modeling concepts can be implemented in a logical multimedia document model.

### Multimedia Datatypes

The logical representation of conceptual multimedia elements should take into account the multimedia extents of the latter. The multimedia datatypes of multimedia elements in MADS (see the hierarchy in fig. 3.11) should generally be represented in the logical model by semantically similar datatypes with regards to their sensorial nature.

For example, we could represent a MADS element of type `Image` by a SMIL document consisting of a JPEG or a GIF file. Similarly, we could use an AVI file-based $Z_YX$ object to represent a conceptual element of type `AudioVideo`.

Note that for complex multimedia datatypes from the hierarchy in fig. 3.11, e.g. the datatype `AudioVideo`, it could alternatively be possible to have them logically represented as a structural combination of elements of simple (constituent) datatypes. For example, in fig. 4.7 the audio-visual logical multimedia element `John` is composed of a parallel synchronization of a video element `John;video` and an audio element `John;audio`.

### Simple Multimedia Representational Relationships

The four simple multimedia representational relationships specified in a MADS conceptual schema (see sect. 3.4.1) can also find their logical representation in a multimedia document. In fact, they can describe the compositional constraints between the logical document representations of the conceptual entities participating into a representational relationship.

In particular, the `MultimediaInclusion` relationship between entity types `A` and `B` (i.e. $A \subset B$) can possibly imply that the complex logical multimedia document

element representing an instance of `B` is composed of, among other things, logical multimedia document elements representing instances of `A`. Note that at this stage is not possible to specify the timeline-synchronization aspect of this composition, i.e. parallel ($PAR$), or sequential ($SEQ$), etc.

The `MultimediaEquality` relationship between entity types `A` and `B` (i.e. $A{=}B$) can imply that at the logical multimedia modeling layer, we may use the same multimedia document elements to represent the related entities of `A` and `B`.

The `MultimediaIntersection` relationship between entity types `A` and `B` (i.e. $A{\cap}B$) can imply that the complex logical multimedia document elements representing related instances of `A` and `B` are composed of some common logical multimedia document elements. Note that at this stage is not possible to specify the timeline-synchronization aspect of this composition, i.e. parallel ($PAR$), or sequential ($SEQ$), etc.

Finally, the `MultimediaDisjointness` relationship between entity types `A` and `B` (i.e. $A\bar{\cap}B$) can imply that the complex logical multimedia document elements representing related instances of `A` and `B` are composed of no common logical multimedia document elements.

### Complex Multimedia Representational Relationships

Complex multimedia representational relationships are based on the 4 simple multimedia representational relationships applied to partitions of multimedia elements.

A possible logical implementation of conceptual set-based multimedia partitionings in a multimedia document model could consist in using operators like spatial or temporal selectors in $Z_YX$ (see sect. 4.1.4), which allow to extract a portion (spatial or temporal) of a logical multimedia document element. In that way, complex multimedia representational relationships can be logically implemented in a multimedia document model by applying the four simple multimedia representational relationships (see above) to the results of selector operators.

### Multirepresentation in Multimedia

Finally, a possible general logical implementation of multirepresentation in multimedia could be by using a $SWITCH$-like operator, which is, in particular, available in SMIL (see sect. 4.1.2) and $Z_YX$ (see sect. 4.1.4). A $SWITCH$ element specifies different alternatives for a specific part of a multimedia document based on a set of discriminating attribute-value pairs for each alternative.

In MADS a perception $s$, for which $n$ parameters have been chosen as relevant, is denoted by a vector: $s{=}\langle p_1, p_2, \ldots, p_n\rangle$, where each $p_i$ is the value for the perception $s$ of its $i^{\text{th}}$ parameter. Hence, by translating the parameters $p_i$ from the conceptual to the logical level, and by verifying their values against the entry test conditions of a $SWITCH$ element, we can provide for each conceptual perception different logical multimedia document implementations of MADS multimedia elements.

### 4.2.3 Extended Conceptual-to-Logical Transformation Guidelines

In the previous subsection we have provided the general basic guidelines for concep-tual-to-logical multimedia model transformations. In this subsection we extend the basic guidelines by taking into account the knowledge provided at the other modeling dimensions of the MADS schema besides the multimedia modeling dimension.

We have shown in the previous subsection that due to the semantic gap between the conceptual and the logical multimedia models, by taking into consideration only the multimedia modeling dimension of MADS, it could be possible to suggest the existence of a compositional relationship of some kind between the elements of a logical multimedia document model. However, it is in general not possible to specify the particular kind of this composition (e.g. sequential, parallel, etc.).

Indeed, the compositional operators like $PAR$ or $SEQ$, which are inherent to the majority of the modern logical multimedia document models, convey the semantics of time-based (timeline-based) composition of multimedia elements. It goes without saying, that many conceptual multimedia datatypes in MADS like e.g. `Image` or `Text` (see the hierarchy in fig. 3.11) do not even contain a temporal component, and hence do not manifest any time-related semantics. On the other hand, for the datatypes like `Video` or `Audio`, which do incorporate the notion of temporal duration, their temporal component simply defines the playback length of the multimedia element and does not pertain to the notion of a common timeline that could be used to synchronize the composition of different multimedia elements.

In our opinion, such additional premise information could optionally be derived from the conceptual schema elements that belong to other modeling dimensions of MADS, like compositional, temporal, or spatial.

It is important to stress, that the above idea does not contradict the principle of inter-layer independence that we rely on in our work. As it has been stated above, due to the optional nature of the proposed inter-layer mapping guidelines, the conceptual-to-logical inference rules can simply be considered as (optional) recommendations. Thus, we do not oblige the structural, temporal, or spatial composition of the logical multimedia documents to necessarily follow respectively the structural, temporal, or spatial composition of the conceptual multimedia elements. However, since both the conceptual MADS model and the logical multimedia document model work with the same application domain (universe of discourse), there is a fair probability that compositional aspects of conceptual and logical models at least partially match.

Let's describe how conceptual modeling features from the structural, temporal, and spatial dimensions of MADS can help us extend our conceptual-to-logical


**Figure 4.8:** A sample MADS schema with multimedia and temporal extents.

multimedia mapping guidelines.

### Temporal Conceptual Modeling Dimension

In the previous subsection we have mentioned, in particular, that the multimedia representational relationships (e.g. the `MultimediaInclusion`) can imply the existence of compositional bonds between the logical multimedia documents representing the instances of the objects linked by the representational relationship. However, we were not able to specify the particular timeline-synchronization semantics of these compositions.

We believe that this kind of information can, in its turn, be suggested by the temporal dimension of the MADS conceptual model (see sect. 3.2).

Consider a sample conceptual schema in fig. 4.8. According to the general inter-layer multimedia mapping guidelines introduced in sect. 4.2.2, the multimedia extents of the entity types `Conference` and `Speaker`, and the `MultimediaInclusion` representational type of the relationship type `PresentsAt`, let us suggest that the logical multimedia document representation of a conference can be composed of multimedia document representations of its speakers.

Taking into account the temporal extents of type `Interval` of the entity types in fig. 4.8, and the temporal extent of type `IntervalBag` of the relationship type `PresentsAt`, we could suggest that the composition of multimedia document representations of conference speakers in the multimedia document representation of the conference should be of synchronization type $SEQ$, i.e. sequentially composed. Moreover, the values of the time extents of particular instances of `Conference`, `Speaker`, and `PresentsAt` would indicate us the order of appearance of the speakers in the sequential composition of the multimedia document representation of their conference.

Besides the temporal extents of relationship types in a MADS schema, also the synchronization relationships (see table 3.2) can be used as premises for extended conceptual-to-logical multimedia mapping guidelines.

Supposing that two entity types `Entity_A` and `Entity_B` in a MADS conceptual schema both have multimedia and temporal extents, and that the multimedia representational type of the relationship type `Relationship_A_B`, which links `Entity_A` and `Entity_B`, suggests the presence of compositional bonds between logical multimedia documents representing instances of `Entity_A` and `Entity_B`, then we could alternatively precise the timeline-synchronization semantics of these


**Figure 4.9:** A sample MADS schema with multimedia and spatial extents.

logical multimedia document compositions based on the synchronization type of the relationship type `Relationship_A_B` (if any).

We describe below possible implications that various synchronization relationship types (see table 3.2) could have on the compositional aspect of logical multimedia documents:

- `SyncPrecede`: sequential composition ($SEQ$), with each successive element starting delayed with regard to the end of the element that precedes it.
- `SyncWithin`: parallel composition ($PAR$), with some of the elements starting delayed with regard to other elements, and where at least one of the elements that does not start delayed finishes the last.
- `SyncStart`: parallel composition ($PAR$) of elements with different durations.
- `SyncEqual`: regular parallel composition ($PAR$).
- `SyncMeet`: regular sequential composition ($SEQ$).
- `SyncOverlap`: parallel composition ($PAR$), with some of the elements starting delayed with regard to other elements.
- `SyncFinish`: parallel composition ($PAR$), with some of the elements starting delayed with regard to other elements, and where all elements finish simultaneously.
- `SyncDisjoint`: unordered sequential composition ($SEQ$), with obligatory delays (pauses) between elements.

### Spatial Conceptual Modeling Dimension

Similarly to the temporal modeling dimension, the spatial modeling dimension of MADS can also be used to extend our conceptual-to-logical multimedia mapping guidelines.

In fig. 4.9 a sample MADS conceptual schema with spatial and multimedia semantics is presented. According to the general inter-layer multimedia mapping guidelines introduced in sect. 4.2.2, the multimedia extents of the entity types `Meeting` and `Participant`, and the `MultimediaInclusion` representational type

of the relationship type `TakesPart`, let us suggest that the logical multimedia document representation of a meeting can be composed of multimedia document representations of its participants.

Based on the topological relationship of type `TopoWithin` of the relationship type `TakesPart`, as well as the coordinates of the meeting participants and of the meeting venue, we could propose that the composition of multimedia document representations of meeting participants in the multimedia document representation of their meeting should be spatially aligned in a similar way as the positions of participants in the meeting room, as expressed by the spatial elements of the MADS schema in fig. 4.9.

We describe below possible implications that various topological relationship types (see table 3.1), which refer to the real-world geometry of spatial elements described a MADS conceptual schema, could have on the compositional aspect of logical multimedia documents with regards to their spatial on-screen composition:

- `TopoDisjoint`: a composition, where screen visualization regions of participating elements are spatially disjoint.
- `TopoOverlap`: a composition, where screen visualization regions of participating elements spatially overlap.
- `TopoWithin`: a composition, where screen visualization regions of participating elements are spatially placed one within the other.
- `TopoTouch`: a composition, where screen visualization regions of participating elements spatially touch.
- `TopoCross`: a composition of image and video elements, where screen visualization regions of images spatially overlap with screen visualization regions of videos for some of the video frames.
- `TopoEqual`: a composition, where screen visualization regions of participating elements spatially coincide.

### Structural Conceptual Modeling Dimension

Besides the temporal and the spatial modeling dimensions, we could also use the structural semantics of a MADS schema to extend our conceptual-to-logical multimedia mapping guidelines.

The structural dimension of MADS provides a rich set of modeling elements and constructs (see sect. 3.2). While these modeling components are used to provide the conceptual description of the application domain (universe of discourse) independently of any possible multimedia extents thereof, we believe that some of the

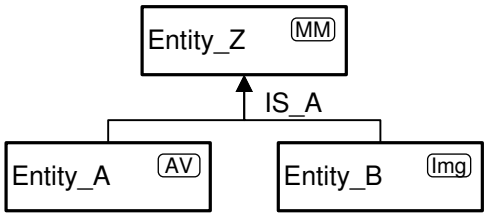



**Figure 4.10:** A sample MADS multimedia schema with an `IS_A` link.

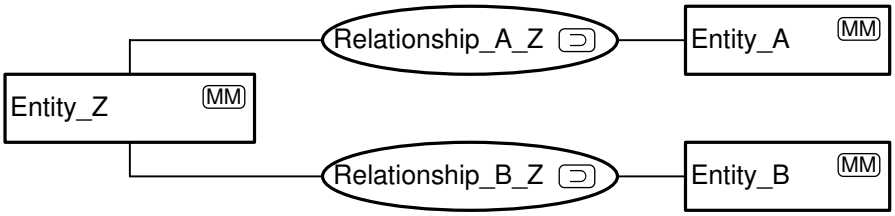



**Figure 4.11:** A sample MADS multimedia schema with multiple relationships.

structural modeling components could be used to enrich the inter-layer multimedia mapping guidelines.

In fig. 4.10 a sample multimedia MADS schema employing an `IS_A` link (i.e. a generalization/specialization relationship) is shown. Both the generalized and the specialized entity types in the schema are characterized by multimedia extents. Since the semantics of an `IS_A` link is that of a classification refinement, which binds two instances that are different representations of the same real-world entity, we could translate an `IS_A` link into the logical multimedia document level by such constructs, which allow to choose the multimedia document representations of particular instances of the same kind (e.g. the multimedia documents representing the instances of the object type `Entity_Z` from fig. 4.10) out of a set of different representation alternatives. This can be achieved by using constructs like `<excl>` in SMIL (see sect. 4.1.2), or `switch` and `query` operators in $Z_YX$ (see sect. 4.1.4).

In another example in fig. 4.11 a multimedia entity type `Entity_Z` is related at the same time with the multimedia entity type `Entity_A` and the multimedia entity type `Entity_B`. According to the general inter-layer multimedia mapping guidelines introduced in sect. 4.2.2, the multimedia extents of the entity types `Entity_A` and `Entity_B`, and the `MultimediaInclusion` representational type of the relationship types `Relationship_A_Z` and `Relationship_B_Z`, let us suggest that on the one hand the logical multimedia document representation of instances of `Entity_Z` can be composed of multimedia document representations of instances of `Entity_A`, and on the other hand the logical multimedia document representation of instances of `Entity_Z` can also be composed of multimedia document representations of instances of `Entity_B`.

In a situation like the one presented in the fig. 4.11, the logical multimedia documents representing instances of both entity types `Entity_A` and `Entity_B` could all participate in the structural composition of the logical multimedia documents representing the instances of `Entity_Z` on equal terms. We could additionally use the information provided by the temporal and the spatial modeling dimensions to deduce temporal and spatial compositional synchronization features among the multimedia document elements representing instances of `Entity_A` and `Entity_B`.

## 4.3 Chapter Summary

In this chapter we have provided an overview of the peculiarities of logical multimedia modeling and of conceptual-to-logical inter-layer transformations.

We have started by providing a state-of-the-art overview of logical multimedia document models. Several existing systems have been presented, and a set of basic universal requirements for multimedia document models has been listed. In particular, we have stressed the importance of differentiating between the compositional and the presentational aspects of multimedia documents.

Then we have discussed the problems of conceptual-to-logical multimedia model transformations. We have argued about the peculiarities of our multimedia inter-layer transformation problem as compared with classical conceptual-to-logical transformation approaches in traditional databases. It has been shown that the high level of semantic independence of conceptual and logical multimedia models substantially complicates the inter-layer transformation task.

The solution that we have proposed consists in providing a set of non-mandatory mapping guidelines, which are intended to help the schema designer in coming up with rich logical multimedia document representations of the application domain, which conform with the conceptual schema.

# Chapter 5

# Experimental Implementation Results

In this chapter we describe a mock-up application designed to support some of the theoretical ideas presented in previous chapters. The application is written in PL/SQL and builds upon the Oracle 10g interMedia cartridge, which implements a number of multimedia-oriented features in the Oracle object-relational DBMS.

## 5.1 Sample Application Framework Overview

In sect. 3.3.3 we have presented the idea of abstract set-based conceptual representation of multimedia data. Later on, in sect. 3.4 the idea of set-based partitioning of multimedia data elements has been described. In particular, the set-based partitioning model has allowed us to provide a foundation for extending the four basic multimedia representational relationships introduced in sect. 3.4.1 with additional multimedia representational relationships (see sect. 3.4.3).

In order to illustrate the above-mentioned theoretical ideas, we have developed a mock-up application, which gives example of implementing the four basic multimedia representational relationships (intersection, inclusion, equality, and disjointness), as well as the partitioning-based additional multimedia representational relationships. The implementation domains that have been chosen for our sample application are a photo archive of company employees, and an EFIT image collection (see example 2 in sect. 3.4.2 and example 4 in sect. 3.4.4 respectively).

The sample application has been implemented on top of the Oracle 10g interMedia cartridge. Oracle interMedia [Cor07] provides multimedia utilities inside the Oracle object-relational DBMS. It enables the management and retrieval of image, audio, and video data represented in a number of popular multimedia formats and allows automating metadata extraction and basic image processing. A simplified interMedia class diagram is presented in fig. 5.1[1]. Of a particular interest to us

[1]The flash lines in fig. 5.1 represent pseudo generalization-specialization links. Although for-

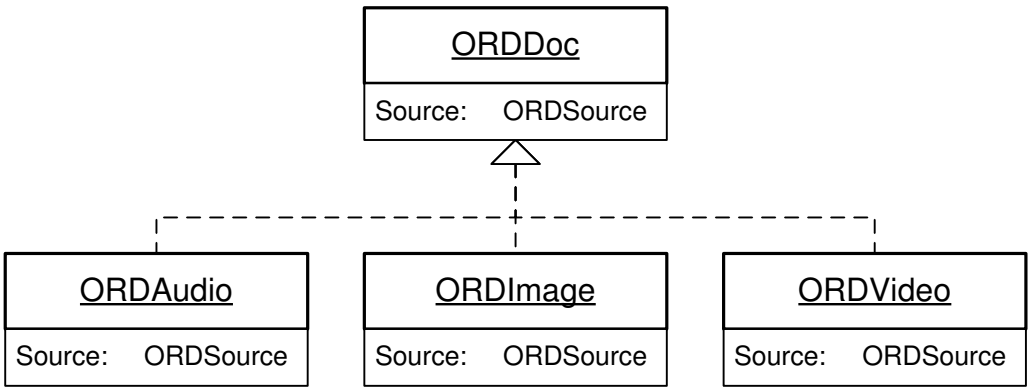



**Figure 5.1:** Oracle interMedia simplified object type diagram.

is the `ORDImage` object datatype, which supports the storage, management, and manipulation of image data.

For our sample application we elaborate the interMedia object type hierarchy from fig. 5.1 by introducing 3 new object types that extend the `ORDImage` object type (see fig. 5.2).

The object type `Image` is an abstract (inheritable and non-instantiable) object type, which incorporates an attribute `Pic` of type `ORDImage`, as well as a number of attributes and methods (not shown here for the sake of simplicity) common to both examples that our sample application illustrates. Inheriting from the object type `Image` are the object subtypes `EFITImage` and `AnnotatedPhoto`, which provide methods and attributes necessary to store and manage the image data for both examples illustrated by our sample application, i.e. respectively an EFIT image collection, and a photo archive of company employees.

Our mock-up application has been developed according to a 3-tier architecture (client - application server - database), which is schematically represented in fig. 5.3.

As already mentioned above, the data tier of the application has been implemented inside an object-relational Oracle database with interMedia cartridge. In particular, the object methods were written in PL/SQL and the queries where written in SQL.

The logic tier of the application has been implemented using Oracle Forms running under Oracle Application Server. A Forms application implements the GUI logic and acts as a database client sending the SQL queries and retrieving the query results via a SQL*Net connection to the database.

---

mally at the object class level the represented inheritance relationships do not exist, the class ORDDoc does allow to store and manage any media data including image, audio, and video.

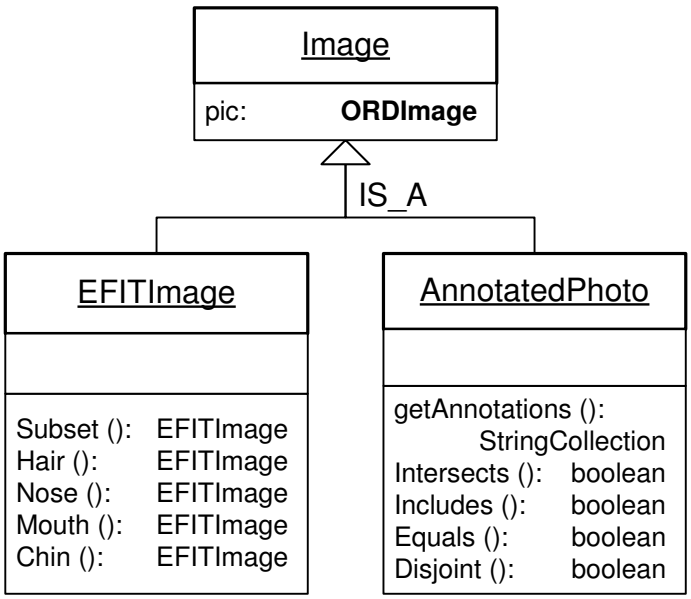



**Figure 5.2:** Object type diagram of the mock-up application.

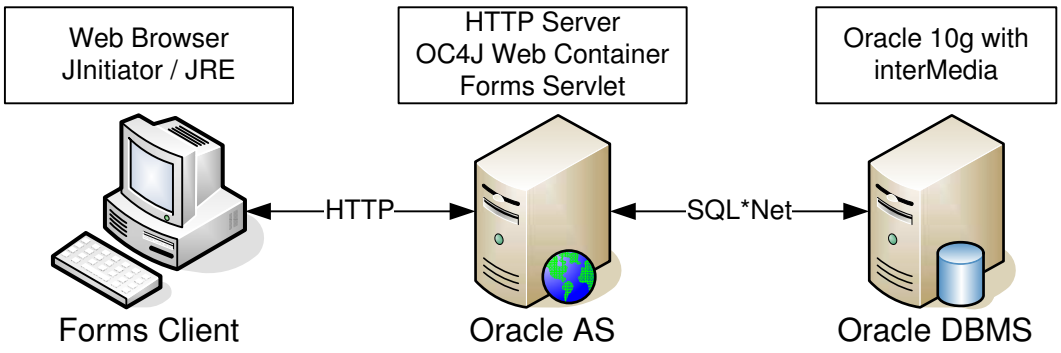



**Figure 5.3:** Mock-up application architecture.

Finally, the presentation tier is made up of a Java applet running in a Web browser on a client machine. The communication between the client and the application server is done via Web (HTTP).

It should be noted that the above-described technologies used at different application tiers are all platform-independent, which allows our application to be run in various heterogeneous environments.

In the next sections we present an overview of the two examples implemented in our sample application.

## 5.2 Sample Application: Employee Database

The object type `AnnotatedPhoto` (see fig. 5.2) allows storing and managing photos augmented with annotation tags about the people they portray (see example 2 in sect. 3.4.2). The annotations are represented in the form of Dublin Core descriptions (DC:Subject) [PNN+07] [NPJN08], which are encapsulated into XMP metadata [Inc05] embedded into image files. The object type `AnnotatedPhoto` inherits from the abstract object type `Image`, and provides the following additional methods:

- `getAnnotations ():StringCollection`.
  A method function returning the set of annotations tags of a photo.
- `Intersects (photo2 AnnotatedPhoto): boolean`.
  A method function to check if a photo intersects with another photo with regards to their annotations.
- `Includes (photo2 AnnotatedPhoto): boolean`.
  A method function to check if a photo includes another photo with regards to their annotations.
- `Equals (photo2 AnnotatedPhoto): boolean`.
  A method function to check if a photo is equal to another photo with regards to their annotations.
- `Disjoint (photo2 AnnotatedPhoto): boolean`.
  A method function to check if a photo and another photo are disjoint (do not intersect) with regards to their annotations.

The methods described above implement the four basic multimedia representational relationships in the context of the annotation-based representation of the content of employee photos. In that way, the method functions of the object type `Image` allow us, for example, to verify that an employee is seen on his (her) department group photo, or that the members of the database laboratory are all

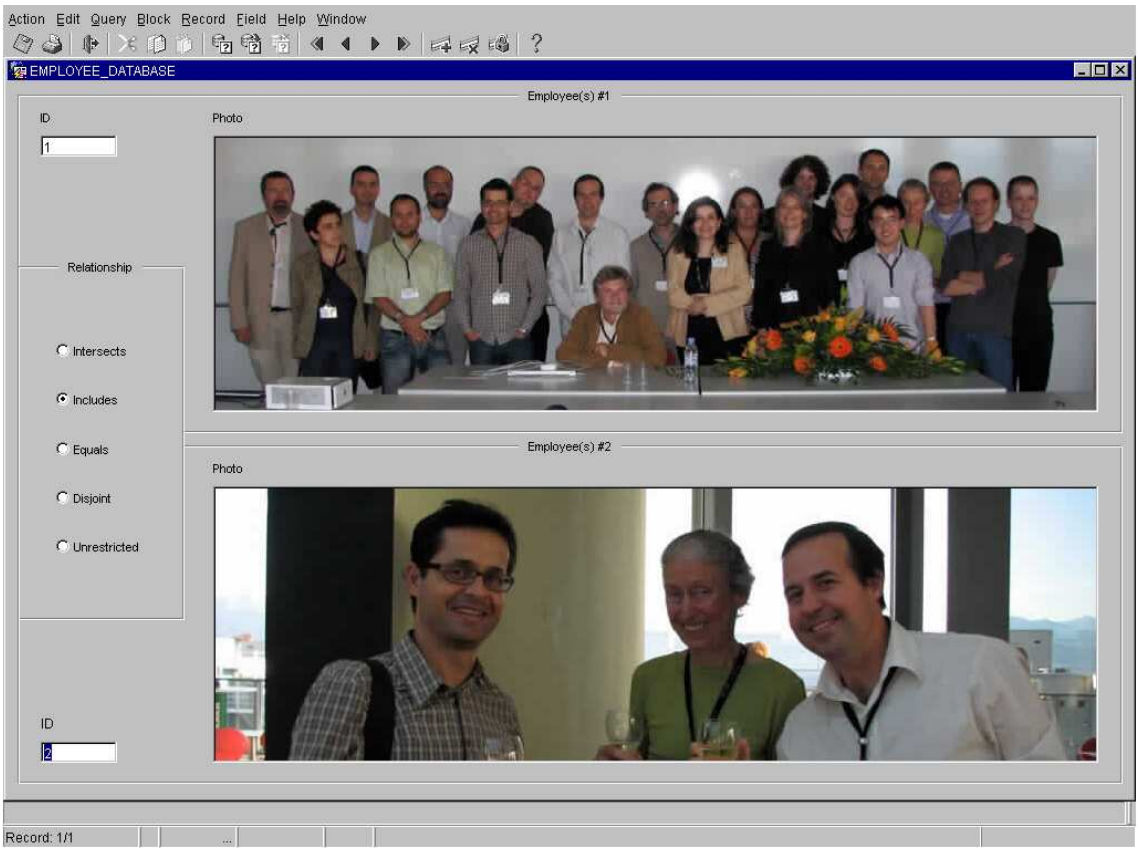



**Figure 5.4:** Mock-up application screenshot (employee database).

present on the computer science department annual photo.

The sample database has been populated with several employee photos of type `AnnotatedPhoto`. Each photo portrays an employee or a group of employees, whose identities are stored with the photo in the form of annotations. The fig. 5.4 shows a screenshot of our mock-up application.

The application window is visually divided into 3 panes: the top pane represents an employee photo from the sample database; the bottom pane represents a different employee photo from the sample database; and the left pane allows choosing the relationship between the photos in the top and the bottom panes. Once the reference employee photo is selected in the top pane, the user can choose one of the four basic multimedia relationships in the left pane. Now, by switching to the bottom pane and executing the query, the system will return into the bottom pane the employee photos that are in the selected relationship with the photo shown in the top pane.

## 5.3 Sample Application: Suspect Database

The object type `EFITImage` (see fig. 5.2) allows storing and managing suspect EFIT images like those described in the example 4 in sect. 3.4.4. Inheriting from the abstract type `Image`, the object type `EFITImage` provides the following specific methods:

- `Subset (x, y, w, h: integer): EFITImage`.
  A method function to crop a region of an EFIT image.

- `Hair (): EFITImage`.
  A method function allowing to extract the hair part of an EFIT image.

- `Nose (): EFITImage`.
  A method function allowing to extract the nose part of an EFIT image.

- `Mouth (): EFITImage`.
  A method function allowing to extract the mouth part of an EFIT image.

- `Chin (): EFITImage`.
  A method function allowing to extract the chin part of an EFIT image.

The methods described above provide a mechanism of image bitmap-based partitioning of EFIT images into their constituent facial parts. Using this partitioning mechanism, we can define complex multimedia representational relationships between two EFIT images, which would allow us, for example, searching for suspects with same noses and chins, or same haircuts but different mouth shapes.

The sample database has been populated with several EFIT objects of type `EFITImage`, each one representing a different suspect photo. The fig. 5.5 shows a screenshot of our mock-up application.

Similarly to the case of annotated employee photos, the application window is visually divided into 3 vertical parts (panes): the left pane representing an EFIT image from the sample database; the right pane representing a different EFIT image from the sample database; and the middle pane for controlling the complex relationship between EFIT images from the left and the right panes. Once the reference EFIT image is selected in the left pane, the user specifies a combination of radio buttons in the middle pane, thus composing the required multimedia relationship (up to 81 combinations are possible). Now, by switching to the right pane and executing the query, the system will return into the right pane the EFIT images that are in the selected relationship with the image shown in the left pane.


**Figure 5.5:** Mock-up application screenshot (suspect database).

## 5.4 Chapter Summary

In this chapter we have presented a mock-up application that illustrates some of the theoretical ideas of conceptual modeling of multimedia data presented in the previous chapters.

In particular, we have shown how the abstract set-based representation of multimedia data elements, as well as simple and complex multimedia representational relationships can be implemented in practice.

Two different possible implementations of image data, namely an annotation-based implementation and a bitmap-pixel-based implementation, have been demonstrated.

# Chapter 6

# Conclusion and Future Directions

## 6.1 Thesis Contributions

In this thesis we have addressed a complex problem of conceptual modeling of multimedia data. We have defined a conceptual modeling approach, which helps to bridge the gap between the semantic content of multimedia data and its underlying physical representation by conceptually representing multimedia data not in a way that is driven by its low-level vision, but rather in a way, which considers multimedia information alongside of other types of information about the universe of discourse of the application that the multimedia information pertains to.

The main contributions of the thesis can be summarized as follows:

- **Multimedia duality principle.**

Conceptual models aim at enabling a representation of the world in accordance with the user's perception and goals. Regarding multimedia data, two alternative ways of apprehending data of this kind are easily identifiable and can be expressed by stating that multimedia information is perceived as either data or metadata. We named this principle the duality principle of multimedia data and made it the fundamental assumption of our modeling approach. The *multimedia as data* perception, which views multimedia data as the subject of modeling in its own right, is inherent to multimedia-centric applications, where multimedia information itself represents the main part of the universe of discourse. On the other hand, the *multimedia as metadata* perception, which is inherent to multimedia-enhanced applications, views multimedia data as an additional source of information about whatever universe of discourse that the application pertains to. While many existing multimedia information systems can be characterized as multimedia-centric, we have demonstrated the increasing importance of the multimedia-enhanced paradigm by the example of applications of the multimedia meeting framework, which we have presented in this thesis.

The multimedia duality principle provides a basis of understanding the essence of multimedia data. In particular, it has provided the theoretical ground of the need of an alternative multimedia-extended conceptual modeling paradigm.

- **Multimedia-extended conceptual modeling technique.**

To achieve the objective of providing a multimedia model allowing to represent any aspect relative to the universe of discourse of an application, whether multimedia or not, we have focused on a modeling technique that extends an existing conceptual model with multimedia modeling features. Indeed, covering multimedia as metadata aspects implies that multimedia features chosen as relevant for the description of an object of interest should be considered in a way most similar to any other feature of the object, be it alphanumeric, spatial, or temporal. Consequently, developing a multimedia conceptual model basically reduces to integrating a multimedia modeling dimension into a conceptual model. This approach provides a non-intrusive way of modeling multimedia-related information, since it builds upon an existing application model, thus allowing for full backward compatibility for multimedia-disabled environments. Although our multimedia-extended modeling approach is generally independent of any particular primary conceptual model, in this thesis we have provided an implementation of our modeling approach based on MADS (Modeling Application Data with Spatio-temporal features), which is a powerful conceptual model defined in our laboratory, and which is, in particular, characterized by structural completeness, spatio-temporal modeling capabilities, and multirepresentation support. The proposed multimedia model is provided in the form of a new orthogonal modeling dimension of MADS, which has been integrated with the other existing modeling dimensions.

Due to its validity for various kinds of applications independently of their universe of discourse, the proposed conceptual multimedia model is suitable not only for multimedia-enhanced but also for multimedia-centric applications like digital image collections (i.e. applications, whose universe of discourse is primarily concerned with multimedia data). In the case of multimedia-centric applications, and compared with other existing modeling approaches like MPEG-7 DS, our MADS-based conceptual multimedia model is distinguished by the expressive power of an extended ER-like approach, as well as, in particular, spatio-temporal and multirepresentation modeling features.

- **A set of conceptual multimedia modeling constructs.**

The elaborated multimedia modeling extension of MADS provides the following main modeling constructs: multimedia datatypes, simple and complex representational relationships, a multimedia partitioning mechanism, and multimedia multirepresentation features.

The multimedia datatype hierarchy is the basis of the proposed multimedia modeling extension. It allows to classify and manage various sensorial and perceptional

varieties of multimedia data. Multimedia datatypes allow to characterize attributes, entity types, and relationship types in a MADS conceptual schema. The conceptual definition of the essence of multimedia datatypes in MADS has been provided using abstract concepts of the set theory, which allows to deal with potentially very different representations of the same datatypes, like, for example, pixel-based or annotation-based representations of pictures.

Multimedia representational relationships, in their turn, provide a mechanism of introducing multimedia-related constraints into a multimedia conceptual schema. Four basic multimedia representational relationships have been defined, and the mechanism for defining additional complex representational relationships based on conceptual set-based partitioning of multimedia data has been described.

Furthermore, we have shown the benefits of MADS multirepresentation support for conceptual modeling of multimedia data, and, in particular, for semantic multimedia zooming.

- **Conceptual-to-logical mapping guidelines.**

Although the main focus of this thesis is the conceptual multimedia database modeling, we have examined the peculiarities of logical multimedia document models and of conceptual-to-logical transformations.

Having provided an overview of logical multimedia document modeling, we have demonstrated that the high level of semantic independence of conceptual and logical multimedia models substantially complicates the inter-layer transformation task. The conceptual-to-logical mapping solution that we have proposed consists in providing a set of non-mandatory mapping guidelines, which are intended to help the schema designer in coming up with rich logical multimedia document representations of the application domain, which conform with the multimedia-enhanced conceptual schema.

- **A sample implementation.**

The feasibility of the research ideas presented in this thesis has been illustrated by a mock-up application that implements some of the theoretical ideas of conceptual modeling of multimedia data described in this work. In particular, we have shown how the abstract set-based representation of multimedia data elements, as well as simple and complex multimedia representational relationships can be implemented in practice. Two different possible implementations of image data, namely an annotation-based implementation and a bitmap-pixel-based implementation, have been provided. The application has been developed on top of Oracle 10g interMedia cartridge, which provides a number of multimedia-oriented features in the Oracle object-relational DBMS.

## 6.2 Future Research Directions

Conceptual modeling of multimedia databases is a complex problem, which becomes increasingly important in the modern age of information technology. Although in this thesis we have considered several aspects of multimedia modeling, we could not cover the entire spectrum of problems inherent to this research domain. This section presents some of the possible future research directions.

**Fuzzy-Set Extensions.** In chapter 3 we have conceptually defined multimedia data using principles of the set theory. The set-based definition of multimedia data has, in particular, been used in sect. 3.4 to define simple and complex multimedia representational relationships. We believe that an interesting evolution of our conceptual multimedia model could consist in extending the set-based representation of multimedia data to fuzzy sets by introducing the degrees of membership of set elements. Such an extension would, for example, allow to introduce fuzzy multimedia representational relationships like "almost equal" or "almost disjoint" for situations where the exact representational relationships do not exist or are difficult to verify.

**Conceptual-to-Logical Transformation.** In chapter 4 we have presented the principles of logical multimedia document modeling. We have argued that the high level of semantic independence of conceptual and logical multimedia models substantially complicates the inter-layer transformation task. The mapping solution that we have proposed consists in providing a set of non-mandatory mapping guidelines intended to help the schema designer to construct logical multimedia document representations of the application domain, which conform with the conceptual multimedia schema. Needless to say, the conceptual-to-logical mapping can turn out to be a laborious and challenging task. For this reason, we believe that it would be beneficial to develop a software tool, which would help the schema designer by proposing possible logical implementations based on the mapping guidelines described in chapter 4. Moreover, in order to verify the compliance of a logical multimedia document model provided by the schema designer with the upper-level conceptual multimedia model, a verification tool based on a logical reasoner could also be developed.

**User Case Studies.** Throughout this thesis we have referenced the multimedia meeting framework, which we have initially presented in sect. 2.4.2. Furthermore, in chapter 5 we have demonstrated the practical feasibility of our research ideas with a sample mock-up application. Nevertheless, we think that it would be beneficial to conduct a series of various application domain real-life user case studies using the multimedia modeling approach described in this work. Such full-scale testings would allow, on the one hand, to better verify implementation peculiarities of our multimedia model in different environments, and, on the other hand, to better an-

alyze the requirements of various potential users of the model. The latter could, in particular, help us refine the modeling elements provided in our approach to better suit the user expectations. For example, a possible refinement could consist in extending our hierarchy of multimedia datatypes, or else enriching our model with additional representational relationship constraints.

# Bibliography


[AFH+95] Jonathan Ashley, Myron Flickner, James Hafner, Denis Lee, Wayne Niblack, and Dragutin Petkovic. The query by image content (QBIC) system. In ACM, editor, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data: May 23–25, 1995, San Jose, California*, volume 24(2) of *SIGMOD Record (ACM Special Interest Group on Management of Data)*, pages 475–475, pub-ACM:adr, 1995. ACM Press.

[All83] J. F. Allen. Maintaining knowledge about time temporal intervals. *Communications of the ACM*, 26:832–842, 1983.

[AN97] Donald A. Adjeroh and Kingsley C. Nwosu. Multimedia database management & requirements and issues. *IEEE MultiMedia*, 4(3):24–33, 1997.

[ATP+05] Thanos Athanasiadis, Vassilis Tzouvaras, Kosmas Petridis, Frederic Precioso, Yannis Avrithis, and Yiannis Kompatsiaris. Using a multimedia ontology infrastructure for semantic annotation of multimedia content. In *Proc. of 5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot '05)*. Springer, 2005.

[BDJS04] Hassina Bounif, Oleksandr Drutskyy, Fabrice Jouanot, and Stefano Spaccapietra. A multimodal database framework for multimedia meeting annotations. In Yi-Ping Phoebe Chen, editor, *MMM*, pages 17–25. IEEE Computer Society, 2004.

[BFG+96] Jeffrey R. Bach, Charles Fuller, Amarnath Gupta, Arun Hampapur, Bradley Horowitz, Rich Humphrey, Ramesh Jain, and Chiao fe Shu. The Virage image search engine: An open framework for image management. In *Proceedings of SPIE-96, 4th Conference on Storage and Retrieval for Still Image and Video Databases*, pages 76–87, San Jose, US, 1996.

[BK97] Vasudev Bhaskaran and Konstantinos Konstantinides. *Image and Video Compression Standards: Algorithms and Architectures*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[BK01] Susanne Boll and Wolfgang Klas. ZYX-A multimedia document model for reuse and adaptation of multimedia content. *IEEE Trans. Knowl. Data Eng*, 13(3):361–382, 2001.

[BKW99] Susanne Boll, Wolfgang Klas, and Utz Westermann. Exploiting ordbms technology to implement the zyx data model for multimedia documents and presentations. In *BTW*, pages 232–250, 1999.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[Bou03] Maged Boulos. The use of interactive graphical maps for browsing medical/health Internet information resources. *International Journal of Health Geographics*, 2003.

[BPS+05] Stephan Bloehdorn, Kosmas Petridis, Carsten Saathoff, Nikos Simou, Vassilis Tzouvaras, Yannis S. Avrithis, Siegfried Handschuh, Ioannis Kompatsiaris, Steffen Staab, and Michael G. Strintzis. Semantic annotation of images and videos for multimedia analysis. In Asunción Gómez-Pérez and Jérôme Euzenat, editors, *ESWC*, volume 3532 of *Lecture Notes in Computer Science*, pages 592–607. Springer, 2005.

[BR09] Dick C.A. Bulterman and Lloyd W. Rutledge. *SMIL 3.0: Flexible Multimedia for Web, Mobile Devices and Daisy Talking Books, 2nd ed.* Springer, 2009.

[Car07] Jorge Cardoso. *Semantic Web Services: Theory, Tools and Applications.* IGI Global, 2007.

[CLT08] C. Cai, K.M. Lam, and Z. Tan. An efficient scene-break detection method based on linear prediction with bayesian cost functions. *CirSysVideo*, 18(9):1318–1323, September 2008.

[Cor07] Oracle Corporation. Oracle Multimedia: Feature overview, 2007. http://www.oracle.com/technology/products/intermedia/.

[CS06] Jorge Cardoso and Amit P. Sheth, editors. *Semantic Web Services, Processes and Applications*, volume 3 of *Semantic Web And Beyond Computing for Human Experience.* Springer, 2006.

[Dav95] Marc Davis. Media streams: an iconic visual language for video representation. pages 854–866, 1995.

[DC98] John David N. Dionisio and Alfonso F. Cárdenas. A unified data model for representing multimedia, timeline, and simulation data. *IEEE Trans. on Knowl. and Data Eng.*, 10(5):746–767, 1998.

[DOS03] Michael C. Daconta, Leo Joseph Obrst, and Kevin T. Smith. The semantic web : a guide to the future of XML, web services, and knowledge management, 2003.

[DS06] Oleksandr Drutskyy and Stefano Spaccapietra. Modeling multimedia data semantics with MADS. In Mong-Li Lee, Kian-Lee Tan, and Vilas Wuwongse, editors, *DASFAA*, volume 3882 of *Lecture Notes in Computer Science*, pages 838–848. Springer, 2006.

[EF91] M. J. Egenhofer and R. D. Franzosa. Point-set topological spatial relations. *International Journal on Geographical Information systems*, 5(2):161–174, 1991.

[FSA96] Charles Frankel, Michael J Swain, and Vassilis Athitsos. Webseer: An image search engine for the world wide web. Technical Report TR-96-14, Department of Computer Science, University of Chicago, July 31 1996. Mon, 05 Aug 1996 18:09:43 GMT.

[GC05] Roberto García and Òscar Celma. Semantic integration and retrieval of multimedia metadata. In Siegfried Handschuh, Thierry Declerck, and Marja-Riitta Koivunen, editors, *5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2005)*, Galway, Ireland, November 2005. CEUR Workshop Proceedings.

[GGM+02] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Ro Oltramari, and Luc Schneider. Sweetening ontologies with dolce. pages 166–181. Springer, 2002.

[Gol91] C.F. Goldfarb. Standards-HyTime: a standard for structured hypermedia interchange. *Computer*, 24(8):81–84, Aug 1991.

[GS00] Anne J. Gilliland-Swetland. Setting the stage. In Murtha Baca, editor, *Introduction to Metadata: Pathways to Digital Information*. Getty Research Institute, 2000.

[GW07] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice Hall, August 2007.

[HJ99] Karel Hrbacek and Thomas Jech. *Introduction to Set Theory, Third Edition, Revised and Expanded (Pure and Applied Mathematics)*. CRC, 3 edition, June 1999.

[Hun01] Jane Hunter. Adding Multimedia to the Semantic Web: Building an MPEG-7 Ontology. January 01 2001.

[ID3] ID3.org. ID3v2. http://www.id3.org/.

[Inc05] Adobe Systems Incorporated. XMP Specification, 2005. http://www.adobe.com/products/xmp/.

[IT05] Antoine Isaac and Raphaël Troncy. Using several ontologies for describing audiovisual documents : A case study in the medical domain. In *Workshop on Multimedia and the Semantic Web, Second European Semantic Web Conference (ESWC 2005), Heraklion, Crete*, 2005.

[JEI02] JEITIA. Exchangeable image file format for digital still cameras: Exif Version 2.2, 2002. http://www.digicamsoft.com/exif22/exif22/html/exif22_1.htm.

[JPA06] Stéphane Jean, Guy Pierra, and Yamine Aït Ameur. Domain ontologies: A database-oriented analysis. In José A. Moinhos Cordeiro, Vitor Pedrosa, Bruno Encarnação, and Joaquim Filipe, editors, *WEBIST 2006, Proceedings of the Second International Conference on Web Information Systems and Technologies: Internet Technology / Web Interface and Applications, Setúbal, Portugal, April 11-13, 2006*, pages 341–351. INSTICC Press, 2006.

[JRT01] Muriel Jourdan, Cécile Roisin, and Laurent Tardif. Constraint techniques for authoring multimedia documents. *Constraints*, 6(1):115–132, 2001.

[JYZ04] Yan Jianfeng, Zhang Yang, and Li Zhanhua. A multimedia document database model based on multi-layered description supporting complex multimedia structural and semantic contents. In *MMM*, pages 33–, 2004.

[KKS02] José Kahan, Marja-Riitta Koivunen, and Ralph R. Swick. Annotea: an open RDF infrastructure for shared web annotations. *Computer Networks*, 39(5):589–608, 2002.

[KPP+06] Dimitrios I. Kosmopoulos, Sergios Petridis, Ioannis Pratikakis, V. Gatos, Stavros J. Perantonis, Vangelis Karkaletsis, and Georgios Paliouras. Knowledge acquisition from multimedia content using an evolution framework. In Ilias Maglogiannis, Kostas Karpouzis, and Max Bramer, editors, *AIAI*, volume 204 of *IFIP*, pages 557–565. Springer, 2006.

[KTCP07] Frederic Kleinermann, Olga De Troyer, Christophe Creelle, and Bram Pellens. Using semantic annotations for customizing navigation paths and virtual tour guides for virtual environments. In *Proceedings of the 4th INTUITION International Conference on Virtual Reality and Virtual Environments*, pages 189–197, 2007.

[Lie01] R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics*, 1(3):469–486, July 2001.

[LSDJ06] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2(1):1–19, February 2006.

[Mar82] J. Martin. *Strategic Data Planning Methodologies*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[Mar04] José M. Martínez. MPEG-7 overview, 2004. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

[MF07] George A. Miller and Christiane Fellbaum. Wordnet then and now. *Language Resources and Evaluation, Springer Netherlands*, pages 209–214, 2007.

[MSS02] B. S. Manjunath, Philippe Salembier, and Thomas Sikora. *Introduction to MPEG-7, Multimedia Content Description Interface*. John Wiley and Sons, Ltd., Jun 2002.

[Mul08] Sjoerd Mullender. SMIL 3.0 Language Profile, 2008. http://www.w3.org/TR/2008/REC-SMIL3-20081201/smil-profile.html.

[Nel65] T. H. Nelson. Complex information processing: a file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th national conference*, pages 84–100, New York, NY, USA, 1965. ACM.

[Noe05] Alva Noe. *Action in Perception*. The MIT Press, 2005.

[NPJN08] Mikael Nilsson, Andy Powell, Pete Johnston, and Ambjörn Naeve. Expressing Dublin Core metadata using the Resource Description Framework (RDF), 2008. http://dublincore.org/documents/dc-rdf/.

[PBK00] G. Pike, N. Brace, and R. Kemp. Investigating E-FIT using famous faces. In *"Forensic Psychology and Law", A. Czerederecka, T. Jaskiewicz-Obydzinska and J. Wojcikiewicz (Eds) pp. 272 - 276*, 2000.

[PFMT03] G. Papagiannakis, A. Foni, and N. Magnenat-Thalmann. Real-time recreated ceremonies in vr restituted cultural heritage sites. In *CIPA XIXth International Symposium*, pages 235–240, July 2003.

[PNN+07] Andy Powell, Mikael Nilsson, Ambjörn Naeve, Pete Johnston, and Thomas Baker. DCMI abstract model, 2007. http://dublincore.org/documents/abstract-model/.

[PS00] Silvia Pfeiffer and Uma Srinivasan. TV anytime as an application scenario for MPEG-7. In *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*, New York, NY, USA, 2000. ACM.

[PSZ+98] Christine Parent, Stefano Spaccapietra, Esteban Zimanyi, P. Donini, Corinne Plazanet, and Christelle Vangenot. Modeling Spatial Data in the MADS Conceptual Model. In *Proc. of the 8th Int. Symp. on Spatial Data Handling, SDH'98*, 1998.

[PSZ06] Christine Parent, Stefano Spaccapietra, and Esteban Zimanyi. *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach*. Springer-Verlag, 2006.

[RBK96] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Human Face Detection in Visual Scenes. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 875–881. The MIT Press, 1996.

[RBM02] Kamisetty Ramamohan Rao, Z. S. Bojkovic, and D. A. Milovanovic. *Multimedia Communication Systems: Techniques, Standards, and Networks*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2002.

[RKN96] Thomas C. Rakow, Wolfgang Klas, and Erich J. Neuhold. Neuhold: Abstractions for multimedia database systems. In *In: Proc. 2 nd Int. Workshop on Multimedia Information Systems (West Point*, pages 26–28. West Point, 1996.

[SC97] J. R. Smith and S. F. Chang. Visually searching the web for content. *IEEE Transactions on Multimedia*, 4(3):12–20, 1997.

[SG00] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing*, 9(4):561–576, 2000.

[Shi06] Clay Shirky. Ontology is overrated: Categories, links, and tags. *Clay Shirky's Writings About the Internet*, 2006.

[SHN07] Ronald Schroeter, Jane Hunter, and Andrew Newman. Annotating relationships between multiple mixed-media digital objects by extending annotea. In Enrico Franconi, Michael Kifer, and Wolfgang May, editors, *ESWC*, volume 4519 of *Lecture Notes in Computer Science*, pages 533–548. Springer, 2007.

[SS99] I. Sexton and P. Surman. Stereoscopic and autostereoscopic display systems. *IEEE Signal Processing Magazine*, 16(3):85–99, May 1999.

[SS01] Linda G. Shapiro and George C. Stockman. *Computer Vision*. Prentice Hall, January 2001.

[SVPZ99] Stefano Spaccapietra, Christelle Vangenot, Christine Parent, and Esteban Zimanyi. Murmur: A research agenda on multiple representations. In *DANTE '99: Proceedings of the 1999 International Symposium on Database Applications in Non-Traditional Environments*, page 373, Washington, DC, USA, 1999. IEEE Computer Society.

[TCU+04] D Thalmann, R Cetre, B Ulicny, P De Heras Ciechomski, and M Clavien. Creating a Virtual Audience for the Heritage of Ancient Theaters and Odea. In *Tenth International Conference on Virtual Systems and Multimedia*, 2004.

[TPC04] Chrisa Tsinaraki, Panagiotis Polydoros, and Stavros Christodoulakis. Integration of OWL ontologies in MPEG-7 and TV-anytime compliant semantic indexing. In Anne Persson and Janis Stirna, editors, *CAiSE*, volume 3084 of *Lecture Notes in Computer Science*, pages 398–413. Springer, 2004.

[Tro03] Raphaël Troncy. Integrating structure and semantics into audio-visual documents. In Dieter Fensel, Katia P. Sycara, and John Mylopoulos, editors, *International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 566–581. Springer, 2003.

[Van04] Christelle Vangenot. Multi-representation in spatial databases using the MADS conceptual model. In *ICA Workshop on Generalisation and Multiple representation, Leicester, UK*, 2004.

[vG82] J. J. van Griethuysen, editor. *Concepts and Terminology for the Conceptual Schema and the Information Base*. Publ. nr. ISO/TC97/SC5-N695, 1982.

[Voo98] Ellen M. Voorhees. Using WordNet for text retrieval. In Christaine Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 285–303. The MIT Press, Cambridge, Massachusetts, 1998.

[Vos07] Jakob Voss. Tagging, Folksonomy & Co - Renaissance of Manual Indexing? In *Proceedings of ISI*, 2007.

[VRL00] Lionel Villard, Cécile Roisin, and Nabil Layaïda. An XML-Based Multimedia Document Processing Model for Content Adaptation. In *DDEP/PODDP*, pages 104–119, 2000.

[VW03] Jeroen Vendrig and Marcel Worring. Interactive adaptive movie annotation. *IEEE MultiMedia*, 10(3):30–37, 2003.

[W3Ca] W3C. Multimedia Semantics Incubator Group. http://www.w3.org/2005/Incubator/mmsem/.

[W3Cb] W3C. SMIL: Synchronized Multimedia. http://www.w3.org/AudioVideo/.

[W3Cc] W3C. W3C Semantic Web Activity. http://www.w3.org/2001/sw/.

[WPLM01] Brian Wylie, Constantine Pavlakos, Vasily Lewis, and Ken Moreland. Scalable rendering on PC clusters. *IEEE Computer Graphics and Applications*, 21(4):62–70, July/August 2001.

[YWX+07] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang. A formal study of shot boundary detection. *CirSysVideo*, 17(2):168–186, February 2007.

# Oleksandr Drutskyy

Rue Couchirard 10
1004 Lausanne
Switzerland
e-mail: oleksandr.drutskyy@epfl.ch

Age: 30
Ukrainian
Married

## Professional Experience

- Database Administrator, Authorized Officer — since 2006
  Department of Information Technology and Security, Swissquote Bank
  - all-round Oracle DBA (installation and patching, performance tuning, query optimization, backups, migrations, etc.)
  - Oracle DBMS 8i, 9i, 10g under SUN Solaris 8, 9, 10
  - Authorized Officer since 2009

- Research Assistant — 2002-2006
  Database Laboratory, Department of Computer & Communication Sciences, EPFL
  - taking part in research projects in the field of multimedia databases
  - developing a multimedia-enhanced database for meeting minutes browsing in the context of IM2 Swiss National Science Foundation project (im2.ch)
  - preparing and leading master curriculum project and exercise sessions on theoretical foundations of databases, design and implementation of relational, object-relational, and XML database applications (French/English)
  - part-time Oracle DBA of the I&C Department and the Database Laboratory

## Professional Certifications

- Oracle Certified Professional: 10g Database Administrator; PL/SQL and Forms Developer

## Education

- PhD in Computer Science (Databases) — 2003-2009
  EPFL, Lausanne
- Pre-doctoral school in computer science — 2001-2002
  EPFL, Lausanne
- Master of Computer Science (with honors) — 1995-2001
  National Technical University of Ukraine, Kiev

## Languages

- Russian, Ukrainian : mother tongues
- English : fluent (TOEFL 277 of 300)
- French : fluent
- German : very good (Mittelstufe Goethe Institut)

## Personal Interests

- Dogs, jogging, cycling, cinema

## Publications

- H. Bounif, O. Drutskyy, F. Jouanot, S. Spaccapietra. A Multimodal Database Framework for Multimedia Meeting Annotations. In proceedings of the International Conference on Multi-Media Modeling (MMM'04), 2004
- O. Drutskyy, S. Spaccapietra. Modeling Multimedia Data Semantics with MADS. In proceedings of 11th International Conference on Database Systems for Advanced Applications (DASFAA 2006), Springer LNCS