12

Knowledge-Based Multimedia Content Indexing and Retrieval

Manolis Wallace, Yannis Avrithis, Giorgos Stamou and Stefanos Kollias

12.1 Introduction

By the end of the last century the question was not whether digital archives are technically and economically viable, but rather how digital archives would be *efficient* and *informative*. In this framework, different scientific fields such as, on the one hand, development of database management systems, and, on the other hand, processing and analysis of multimedia data, as well as artificial and computational intelligence methods, have observed a close cooperation with each other during the past few years. The attempt has been to develop intelligent and efficient human–computer interaction systems, enabling the user to access vast amounts of heterogeneous information, stored in different sites and archives.

It became clear among the research community dealing with content-based audiovisual data retrieval and new emerging related standards such as MPEG-21 that the results to be obtained from this process would be ineffective, unless major focus were given to the semantic information level, defining what most users desire to retrieve. It now seems that the extraction of semantic information from audiovisual-related data is tractable, taking into account the nature of useful queries that users may issue and the context determined by user profiles [1].

Additionally, projects and related activities supported under the R&D programmes of the European Commission have made significant contributions to developing:

- new models, methods, technologies and systems for creating, processing, managing, networking, accessing and exploiting digital content, including audiovisual content;
- new technological and business models for representing information, knowledge and know-how;
- applications-oriented research, focusing on publishing, audiovisual, culture, education and training, as well as generic research in language and content technologies for all applications areas.

Multimedia Content and the Semantic Web Edited by Giorgos Stamou and Stefanos Kollias
 ISBN: 0-470-85753-6.

In this chapter a novel platform is proposed that intends to exploit the aforementioned ideas in order to offer user friendly, highly informative access to distributed audiovisual archives. This platform is an approach towards realizing the full potential of globally distributed systems that achieve information access and use. Of primary importance is the approach's contribution to the Semantic Web [2]. The fundamental prerequisite of the Semantic Web is 'making content machine-understandable'; this happens when content is bound to some formal description of itself, usually referred to as 'metadata'. Adding 'semantics to content' in the framework of this system is achieved through algorithmic, intelligent content analysis and learning processes.

The system closely follows the developments of MPEG-7 [3–5] and MPEG-21 [6] standardization activities, and successfully convolves technologies in the fields of computational intelligence, statistics, database technology, image/video processing, audiovisual descriptions and user interfaces, to build, validate and demonstrate a novel intermediate agent between users and audiovisual archives. The overall objective of the system is to be a stand-alone, distributed information system that offers enhanced search and retrieval capabilities to users interacting with digital audiovisual archives [7]. The outcome contributes towards making access to multimedia information, which is met in all aspects of everyday life, more effective and more efficient by providing a user-friendly environment.

The chapter is organized as follows. In Section12.2 we provide the general architecture of the proposed system. We continue in Section 12.3 by presenting the proprietary and standard data models and structures utilized for the representation and storage of knowledge, multimedia document information and profiles. Section 12.4 presents the multimedia indexing algorithms and tools used in offline mode, while Section 12.5 focuses on the operation of the system during the query. Section 12.6 is devoted to the personalization actions of the system. Finally, Section 12.7 provides experimental results from the actual application of the proposed system and Section 12.8 discusses the directions towards which this system will be extended through its successor R&D projects.

12.2 General Architecture

The general architecture is provided in Figure 12.1, where all modules and subsystems are depicted, but the flow of information between modules is not shown for clarity. More detailed information on the utilized data models and on the operation of the subsystems for the two main modes of system operation, i.e. *update mode* and *query mode*, are provided in the following sections. The system has the following features:

- Adopts the general features and descriptions for access to multimedia information proposed by MPEG-7 and other standards such as emerging MPEG-21.
- Performs dynamic extraction of high-level semantic description of multimedia documents on the basis of the annotation that is contained in the audiovisual archives.
- Enables the issuing of queries at a high semantic level. This feature is essential for unifying user access to multiple heterogeneous audiovisual archives with different structure and description detail.
- Generates, updates and manages users' profile metadata that specify their preferences against the audiovisual content.
- Employs the above users' metadata structures for filtering the information returned in response to their queries so that it better fits user preferences and priorities.

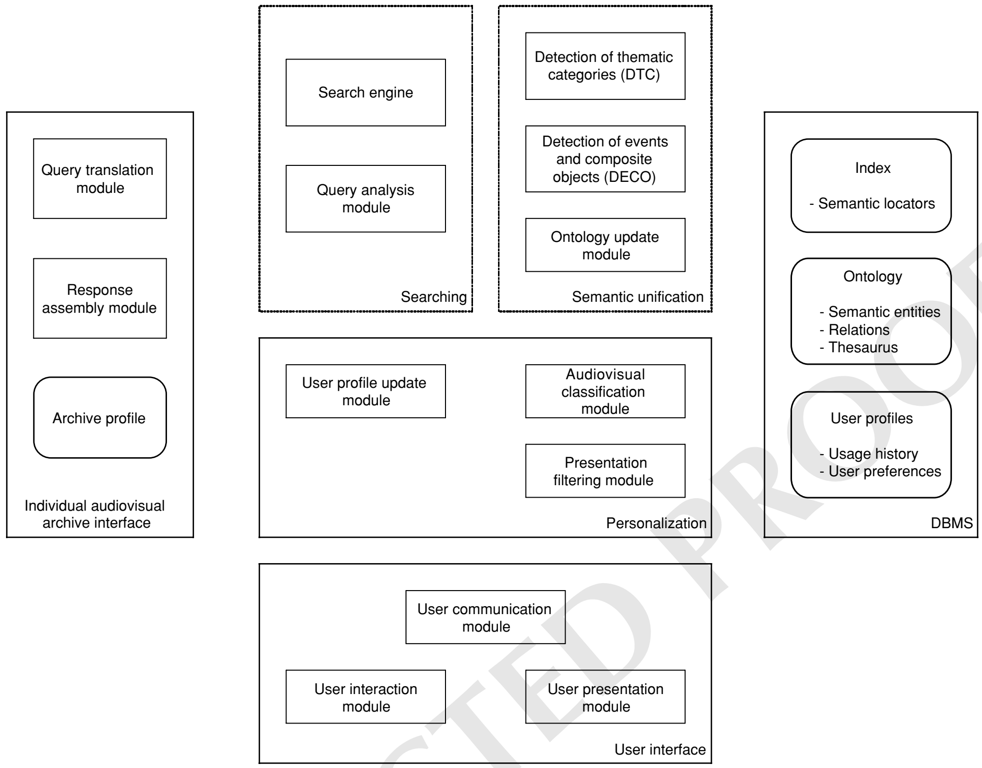

Figure 12.1 General architecture of the system

- Gives users the ability to define and redefine their initial profile.
- Is capable of communicating with existing audiovisual archives with already developed systems with proprietary (user and system) interfaces.
- User interfaces employ platform-independent tools targeting both the Internet and WWW and broadcast type of access routes.

Additionally, it is important that the system has the following features related to user query processing:

- *Response time*: internal intelligent modules may use semantic information available in the DBMS (calculated by *Detection of Thematic Categories* (DTC), *Detection of Events and Composite Objects* (DECO) and the *UserProfile Update Module*) to locate and rank multimedia documents very quickly, without querying individual audiovisual archives. In cases where audiovisual unit descriptions are required, query processing may be slower due to the large volume of information. In all cases it is important that the overall response time of the system is not too long as perceived by the end user.
- *Filtering*: when a user specifies a composite query, it is desirable that a semantic query interpretation is constructed and multimedia documents are filtered as much as possible

according to the semantic interpretation and the user profile, in order to avoid the overwhelming responses of most search engines.
- *Exact matching*: in the special cases where the user query is simple, e.g. a single keyword, the system must return all documents whose description contains the keyword; no information is lost this way.
- *Ranking*: in all cases retrieved documents must be ranked according to the user's preferences and their semantic relevance to the query, so that the most relevant documents are presented first.
- *Up-to-date information*: since the system is designed for handling a large number of individual audiovisual archives whose content may change frequently, the DBMS must be updated (either in batch updates or in updates on demand) to reflect the most recent archive content.

The description of the subsystems' functionality follows the distinction between the two main modes of operation. In *query mode*, the system is used to process user requests, and possibly translate and dispatch them to the archives, and assemble and present the respective responses. The main internal modules participating in this mode are the *query analysis*, *search engine*, *audiovisual classification* and *presentation filtering* modules.

An additional *update mode* of operation is also necessary for updating the content description data. The general scope of the update mode of operation is to adapt and enrich the DBMS used for the unified searching and filtering of audiovisual content. Its operation is based on the *semantic unification* and the *personalization* subsystems. The semantic unification subsystem is responsible for the construction and update of the *index* and the *ontology,* while the personalization subsystem updates the *user profiles*. In particular, a batch update procedure can be employed at regular intervals to perform DTC and DECO on available audiovisual units and update the database. Alternatively, an *update on demand* procedure can be employed whenever new audiovisual units are added to individual archives to keep the system synchronized at all times. Similar choices can be made for the operation of the user profile update module. The decision depends on speed, storage and network traffic performance considerations. The main internal modules participating in the update mode are *DTC*, *DECO*, *ontology update* and *user profile update*.

In the following we start by providing details on the utilized data structures and models, continue by describing the functionality of the objective subsystems operating in offline and online mode, where additional diagrams depict detailed flow of information between modules, and conclude with the presentation of the personalization methodologies.

12.3 The Data Models of the System

The system is aimed to operate as a mediator, providing to the end user unified access to diverse audiovisual archives. Therefore, the mapping of the archive content on a uniform data model is of crucial importance. The specification of the model itself is a challenging issue, as the model needs to be descriptive enough to adequately and meaningfully serve user queries, while at the same time being abstract and general enough to accommodate the mapping of the content of any audiovisual archive. In the following we provide an overview of such a data model, focused on the support for semantic information services.

12.3.1 The Ontology

The ontology of the system comprises a set of description schemes (DSs) for the definition of all semantic entities and their relations. It actually contains all knowledge of semantic information used in the system. The ontology, among other actions, allows:

- storing in a structured manner the description of semantic entities and their relations that experts have defined to be useful for indexing and retrieval purposes;
- forming complex concepts and events by the combination of simple ones through a set of previously specified relations;
- expanding the user query by looking for synonyms or related concepts to those contained in the semantic part of the query.

To make the previous actions possible, three types of information are included in the ontology:

- *Semantic entities*: entities such as thematic categories, objects, events, concepts, agents and semantic places and times are contemplated in the encyclopedia. All normative MPEG-7 semantic DSs are supported for semantic entities whereas the treatment of thematic categories as semantic entities is unique to the system, so additional description schemes are specified.
- *Semantic relations*: the relations linking related concepts as well as the relations between simple entities to allow forming more complex ones are specified. All normative MPEG-7 semantic DSs are supported for semantic relations.
- *A thesaurus*: it contains simple views of the complete ontology. Among other uses, it provides a simple way to associate the words present in the semantic part of a query to other concepts in the encyclopedia. For every pair of semantic entities (SEs) in the ontology, a small number of semantic relations are considered in the generation of the thesaurus views; these relations assess the type and level of relationship between these entities. This notion of a thesaurus is unique to this system and, therefore, additional DSs are specified.

An initial ontology is manually constructed possibly for a limited application domain or specific multimedia document categories. That is, an initial set of *semantic entities* is created and structured using the experts' assessment and the supported *semantic relations*. The *thesaurus* is then automatically created.

A similar process is followed in the ontology update mode, in which the knowledge experts specify new semantic entities and semantic entity relations to be included in the encyclopedia. This is especially relevant when the content of the audiovisual archives is dramatically altered or extended.

Semantic entities

The semantic entities in the ontology are mostly media abstract notions in the MPEG-7 sense. Media abstraction refers to having a single semantic description of an entity (e.g. a soccer player) and generalizing it to multiple instances of multimedia content (e.g. a soccer player from any picture or video). As previously mentioned, entities such as thematic categories, objects, events, concepts, agents and semantic places and times are contemplated in the ontology, and

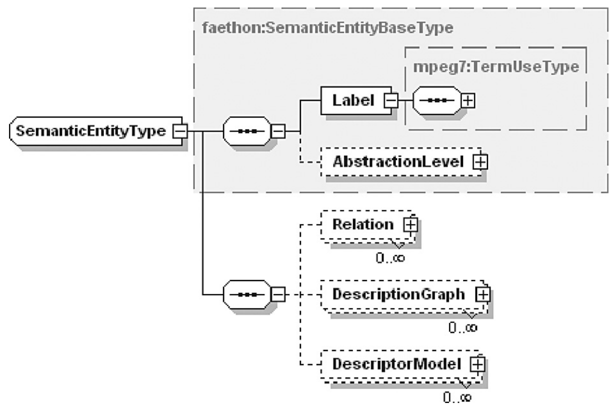

Figure 12.2 The SemanticEntityType. Textual descriptions are supported through Labels and composite objects are described through the DescriptionGraphs

all normative MPEG-7 semantic DSs are supported for SEs. Semantic entities are structured in the SemanticEntities DS (Figure 12.2).

An SE is composed of:

- a textual annotation including synonyms and different language representations;
- zero to several Description Graphs (DGs) relating the various SEs that are associated to the SE and linked by their valued SRs. DGs provide a means for 'semantic definition' of the entity.

Very simple SEs do not require a DG but are only described by their corresponding terms (e.g. ball).

Semantic relations

As previously mentioned, all normative MPEG-7 semantic DSs are supported for semantic relations. Additionally, the definition of custom, system proprietary semantic relations is supported via the utilization of the generic SemanticRelationType (Figure 12.3). In order to make the storage of the relations more compact and to allow for some elementary ontological consistency checks, the relations' mathematical properties, such as symmetry, transitivity and type of transitivity, reflexivity etc., are also stored in the ontology. Using them the ontology update tools can automatically expand the contained knowledge by adding implied and inferred semantic relations between SEs, and validate new information proposed by the knowledge experts against that already existing in the ontology.

An important novelty of the ontology utilized in this system, when compared to the current trend in the field of ontological representations, is the inherent support of degrees in all semantic relations (Figure 12.4). Fuzziness in the association between concepts provides greater descriptive power, which in turn allows for more 'semantically meaningful' analysis of documents, user queries and user profiles. As a simple example of the contribution of this fuzziness in the descriptive of the resulting ontology, consider the concepts of car, wheel

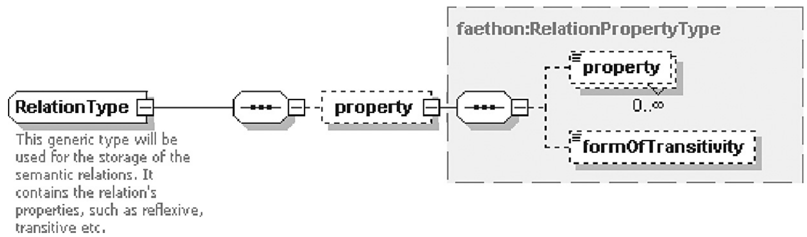

Figure 12.3 The RelationType. In the generic type only the relation properties are required

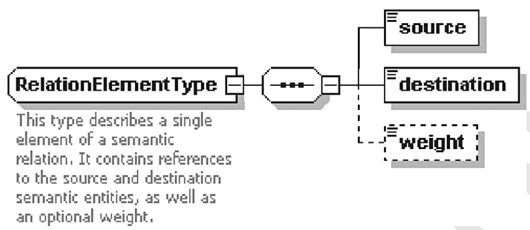

Figure 12.4 The RelationElementType. All semantic relations between pairs of semantic entities are described using this type. The weight, although optional, is of major importance for this system

and rubber. Although the inclusion between them is obvious, it is clear that the inclusion of wheel in car semantically holds more than that of rubber in car. Using degrees of membership other than one, and applying a sub-idempotent transitive closure, i.e. an operation that will allow the relation of car and rubber to be smaller than both the relation between car and wheel and the relation between wheel and rubber, we acquire a much more meaningful representation.

Thesaurus

The description of the relationships among the various SEs in the ontology using a single semantic relation forms a graph structure. The graph nodes correspond to all SEs in the encyclopedia, whereas graph links represent the type and degree of relationship between the connected nodes. Combining all the relations in one graph, in order to acquire a complete view of the available knowledge, results in a very complex graph that cannot really provide an easy to use view of an application domain.

Simplified views of this complex graph structure are represented in the ontology by means of the thesaurus. Since the concept of thesaurus is unique to this system, additional DSs are specified; in order to make the representation more flexible, the same structure as the one used for the distinct semantic relations of the ontology is also utilized for the representation of the ontological views in the thesaurus (Figure 12.5).

All the information in the thesaurus can be obtained by tracking the links among different SEs through the SemanticEntities and SemanticRelations DSs contained in the ontology, based on the thesaurus generation rules, specifying which relations to utilize for each view, and in which way to combine them, as well as the relation properties. Actually, this is the way in

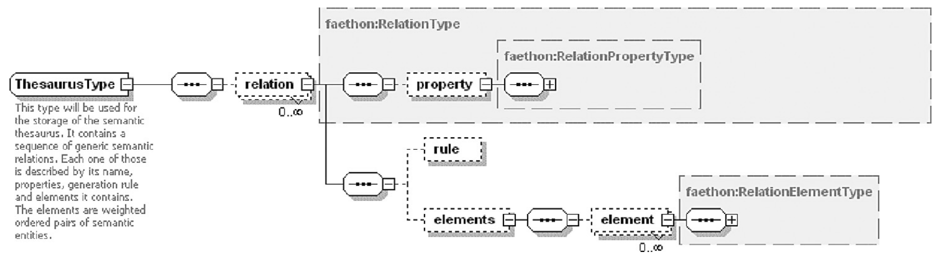

Figure 12.5 The ThesaurusType. Initially only the rule and property fields are filled. In ontology update mode, the ontology update module uses them as input, together with the distinct semantic relations, in order to automatically generate the semantic views stored in the relation element fields

which the thesaurus is initially created and periodically updated in the encyclopedia update module.

The usefulness of the thesaurus is that it codes the information in a simpler, task-oriented manner, allowing faster access.

12.3.2 Index

The index is the heart of the unified access to various archives, as it collects the results of the document analysis taking place in the framework of the semantic unification process. Specifically, the index contains sets of document locators (links) for each SE (thematic category, object, event, concept, agent, semantic place or semantic time) in the ontology (Figure 12.6). Links from thematic categories to multimedia documents are obtained by the DTC procedure (mapping the abstract notions to which each multimedia document is estimated to be related to the thematic categories in the ontology) while links to the remaining SEs are provided by the DECO procedure (mapping the simple and composite objects and events detected in each multimedia document to their corresponding semantic entities in the ontology).

The index is used by the search engine for fast and uniform retrieval of documents related to the semantic entities specified in, or implied by, the query and the user profile. Document locators associated to index entities may link to complete audiovisual documents, objects, still images or other video decomposition units that may be contained in the audiovisual databases (Figure 12.7).

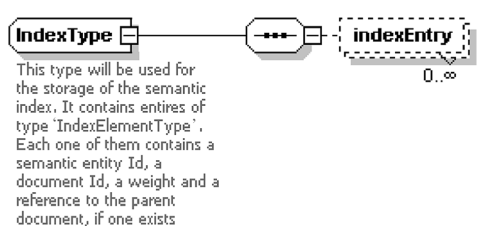

Figure 12.6 The IndexType comprises a sequence of entries, each one referring to a distinct semantic entity–document pair

```
</complexType><complexType name="IndexEntryType">
    <attribute name="semanticEntity" type="IDREF" use="required"/>
    <attribute name="document" type="string" use="required"/>
    <attribute name="weight" type="mpeg7:zeroToOneType" use="optional"/>
    <attribute name="parentDocument" type="string" use="optional"/>
</complexType>
```

Figure 12.7 The Index EntryType; it cannot be displayed graphically, as all of its components are included as attributes rather than child elements. Entities are represented using their unique id in the ontology and documents using a URL, the detailed format of which may be custom to the specific archive. Attribute weight provides for degrees of association, while attribute parentDocument provides for decomposition of multimedia documents into their semantic spatio-temporal components

12.3.3 User Profiles

User profiles contain all user information required for personalization. The contents of the user profiles are decomposed into the *usage history* and the *user preferences*. Profiles are stored using UserProfile Ds, which contain a UserPreferences DS and possibly a UsageHistory DS (Figures 12.8 and 12.9). The UsageHistory DS is only used in dynamic (i.e. not static) profiles.

Usage history

All of the actions users perform while interacting with the system are important for their profile and are therefore included in their usage history (Figure 12.10). When the user logs on to the

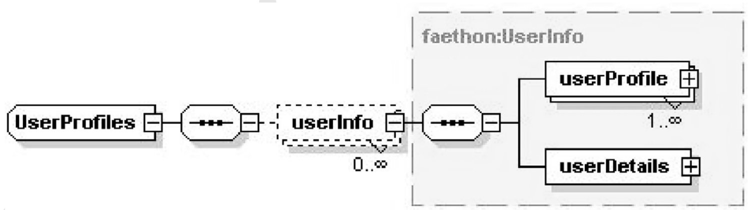

Figure 12.8 As already mentioned, a user may have more than one profile. Distinct profiles of the same user are grouped together via the UserInfo DS

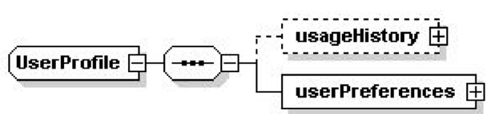

Figure 12.9 The UserProfile DS. The usageHistory part is only utilized for dynamic profiles, i.e. when the user has allowed the system to monitor user actions and based on them to automatically update user preferences

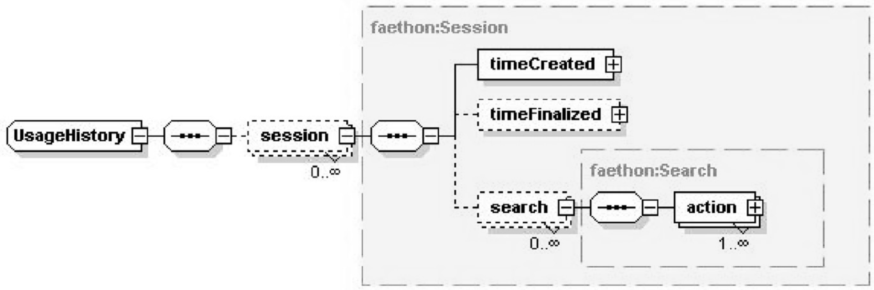

Figure 12.10 The UsageHistory DS. Each action is formed as a selection among new query, request for structural information about the document, request for metadata of the selected document or document segment, or request for the actual media

system a new *session* starts. The session ends when the user logs out, terminates the client program or changes his/her *active profile* (i.e. the profile he/she is currently using).

Within a session a user may try to satisfy a single or more of his/her needs/requests. Each one of those attempts is called a *search*. The search is a complex multi-step procedure; each one of the possible steps is an *action*. Different types of actions are supported by the system; these include formulation of a *query*, request for *structural* or *meta* information and request for the *media* itself.

Usage history contains records of sessions that belong to the same profile, stored using the Session DS. This DS may contain information concerning the time it was created (i.e. the time the session started) as well as the time it was finalized (i.e. the time the session was terminated). It also contains an ordered set of Search DSs. Their order is equivalent to the order in which the corresponding searches were performed by the user. Search DS, as implied by its name, is the structure used to describe a single search. It contains an ordered ser of Action DSs. Since different searches are not separated by a predefined event (as logging on) it is up to the system to separate the user's actions into different searches. This is accomplished by using query actions as separators but could also be tackled using a more complex algorithm, which might for example estimate the relevance between consequent queries. Action DSs may be accompanied by records of the set of documents presented to the user at each time. Such records need not contain anything more than document identifiers for the documents that were available to the user at the time of his/her action, as well as their accompanying ranks (if they were also presented to the user). Their purpose is to indicate what the user was reacting to.

User preferences

User preferences are partitioned into two major categories. The first one includes *metadata*-related and *structural* preferences while the second contains *semantic* preferences (Figures 12.11 and 12.12). The first category of preferences contains records indicating user preference for creation, media, classification, usage, access and navigation (e.g. favourite actors/directors

JWBK013-12 JWBK013-Stamou January 13, 2005 17:10 Char Count= 0

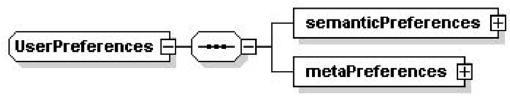

Figure 12.11 The UserPreferences DS. Although the metadata part is supported, the main emphasis of the system is on the semantic portion of the user preferences

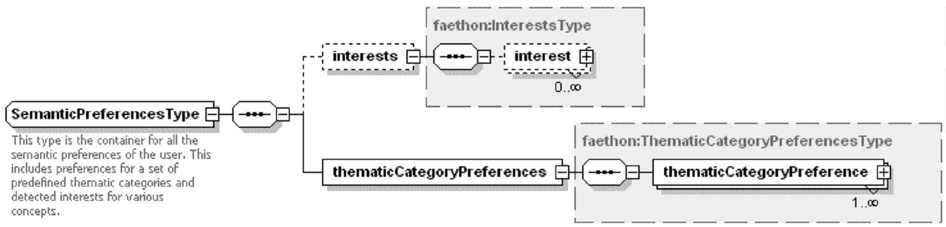

Figure 12.12 Preferences are grouped in preference degrees for the predefined categories and custom, automatically mined, interests

or preference for short summaries). Semantic preferences may again be divided into two (possibly overlapping) categories. The first contains records of *thematic categories*, thus indicating user preference for documents related to them. The second, which we may refer to as *interests*, contains records of *simple* or *composite semantic entities* or *weighted sets of semantic entities*, thus indicating user preference for documents related to them. Both metadata-related and semantic preferences are mined through the analysis of usage history records and will be accompanied by weights indicating the *intensity* of the preference. The range of valid values for these weights may be such as to allow the description of 'negative' intensity. This may be used to describe the user's dislike(s).

Metadata-related and structural preferences are stored using the UserPreferences DS, which has been defined by MPEG-7 for this purpose. Still, it is the semantic preferences that require the greater attention, since it is at the semantic level that the system primarily targets. Semantic preferences are stored using the system proprietary SemanticPreferences DS.

This contains the semantic interests, i.e. degrees of preference for semantic entities and degrees of preference for the various predefined thematic categories. Out of those, the thematic categories, being more general in nature, (i) are related to more documents than most semantic entities and (ii) are correctly identified in documents by the module of DTC, which takes the context into consideration. Thus, degrees of preference for thematic categories are mined with a greater degree of certainty than the corresponding degrees for simple semantic entities and shall be treated with greater confidence in the process of personalization of retrieval than simple interests. For this reason it is imperative that thematic categories are stored separately from interests. The SemanticPreferences DS contains a ThematicCategoryPreferences DS (Figure 12.13), which corresponds to the user's preferences concerning each of the predefined thematic categories, as well as an Interests DS (Figure 12.14), which contains mined interests for more specific entities in the ontology. Static profiles, either predefined

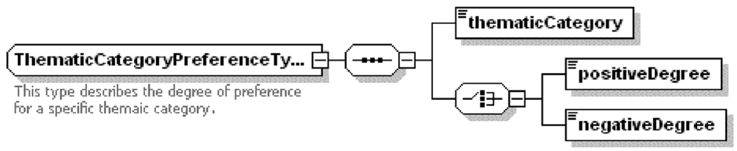

Figure 12.13 The ThematicCategoryPreferenceType allows both for preference and dislike degrees for a given topic/thematic category

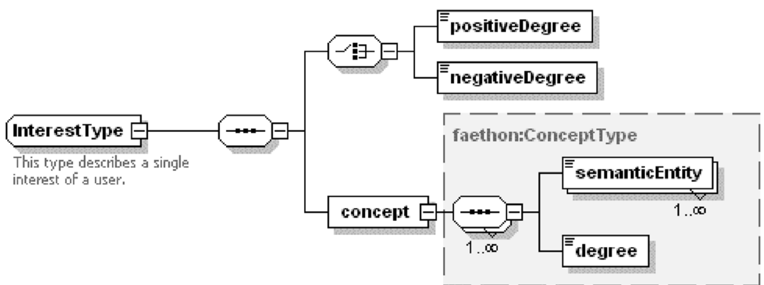

Figure 12.14 The InterestType provides for the representation of complex notions and composite objects in the form of fuzzy sets of semantic entities

by experts or defined by the end users themselves, only contain preferences for thematic categories.

The ThematicCategoryPreferences DS contains a record for each thematic category in the ontology. This entry contains a thematic category identifier and a weight indicating the intensity of the user's preference for the specific thematic category. When, on the other hand, it comes to the representation of more specific, automatically estimated user interests, such a simple representation model is not sufficient [8].

For example, let us examine how an error in estimation of interests affects the profiling system and the process of retrieval, in the cases of positive and negative interests. Let us suppose that a user profile is altered by the insertion of a positive interest that does not actually correspond to a real user interest. This will result in consistent selection of irrelevant documents; the user reaction to these documents will gradually alter the user profile by removing this preference, thus returning the system to equilibrium. In other words, miscalculated positive interests are gradually removed, having upset the retrieval process only temporarily.

Let us now suppose that a user profile is altered by the insertion of a negative interest that does not correspond to a real user dislike. Obviously, documents that correspond to it will be down-ranked, which will result in their consistent absence from the set of selected documents; therefore the user will not be able to express an interest in them, and the profile will not be re-adjusted.

This implies that the personalization process is more sensitive to errors that are related to negative interests, and therefore such interests need to be handled and used with greater caution. Therefore, negative interests need to be stored separately than positive ones, so that they may be handled with more caution in the process of personalized retrieval.

Let us also consider the not rare case in which a user has various distinct interests. When the user poses a query that is related to one of them, then that interest may be used to facilitate the ranking of the selected documents. Usage of interests that are unrelated to the query may only be viewed as addition of noise, as any proximity between selected documents and these interests is clearly coincidental, in the given context. In order to limit this inter-preference noise, we need to be able to identify which interests are related to the user query, and to what extent. Thus, distinct positive interests need to be stored separately from each other as well.

Following the above principles, the Interests DS contains records of the interests that were mined from this profile's usage history; each of these records is composed of an interest intensity value as well as a description of the interest (i.e. the semantic entities that compose it and the degree to which they participate to the interest). Simple and composite semantic entities can be described using a single semantic entity identifier. Weighted sets can easily be described as a sequence of semantic entity identifiers accompanied by a value indicating the degree of membership.

12.3.4 Archive Profiles

The main purpose of an audiovisual archive profile is to provide a mapping of an archive's custom multimedia document DS to the system's unified DSs. Each archive profile contains all necessary information for the construction of individual queries related to metadata, and particularly mapping of creation, media, usage, syntactic, access and navigation description schemes. Therefore, the structure of archive profiles is based on the multimedia document description schemes. Semantic description schemes are included as they are handled separately by the semantic unification subsystem.

In contrast to the ontology, the index and the user profiles, the archive profiles are stored at the distinct *audiovisual archive interfaces* and not in the central DBMS (Figure 12.15).

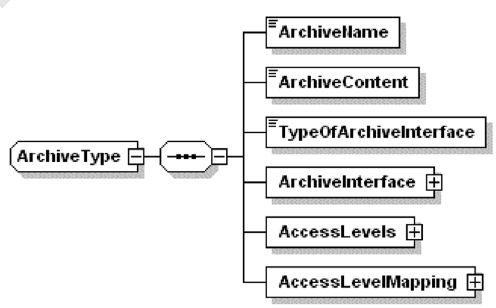

Figure 12.15 The information stored locally at the archive profile allows for the automatic translation of system queries to a format that the custom content management application of the archive can parse, as well as for the translation of the response in the standardized data structures of the system

12.4 Indexing of Multimedia Documents

As we have already mentioned, the main goal of the system is to provide to the end users uniform access to different audiovisual archives. This is accomplished by mapping all audiovisual content to a semantically unified index, which is then used to serve user queries. The update mode of operation, in addition to the analysis of usage history for the update of user preferences, is charged with the effort to constantly adapt to archive content changes and enrich the index used for the unified searching and filtering of audiovisual content.

The index is stored in the DBMS as an XML file containing pairs of semantic entities and documents or document segments, and possibly degrees of association. This structure, although sufficient as far as its descriptive power is concerned, does not allow for system operation in a timely manner. Therefore, a more flexible format is used to represent the index information in main memory; the chosen format employs binary trees to represent the index as a fuzzy binary relation between semantic entities and documents (in this approach each document segment is treated as a distinct document). This model allows for $O(logn)$ access time for the documents that are related to a given semantic entity, compared to a complexity of $O(n)$ for the sequential access to the stored XML index [9]. It is worth mentioning that although thematic categories have a separate and important role in the searching process, they are a special case of other concepts, and thus they are stored in the index together with other semantic entities.

The modules that update the semantic entities in the index and their links to the audiovisual units are DECO and DTC (Figure 12.16). The former takes the multimedia document descriptions as provided by the individual archive interfaces and maps them to semantic entities' definitions in the ontology, together with a weight representing the certainty with which the system has detected the semantic entities in question. Furthermore, it scans the audiovisual units and searches for composite semantic structures; these are also linked together in the index. The latter accepts as input the semantic indexing of each document, as provided in the DBMS, and analyses it in order to estimate the degree to which the given document is related to each one of the predefined thematic categories.

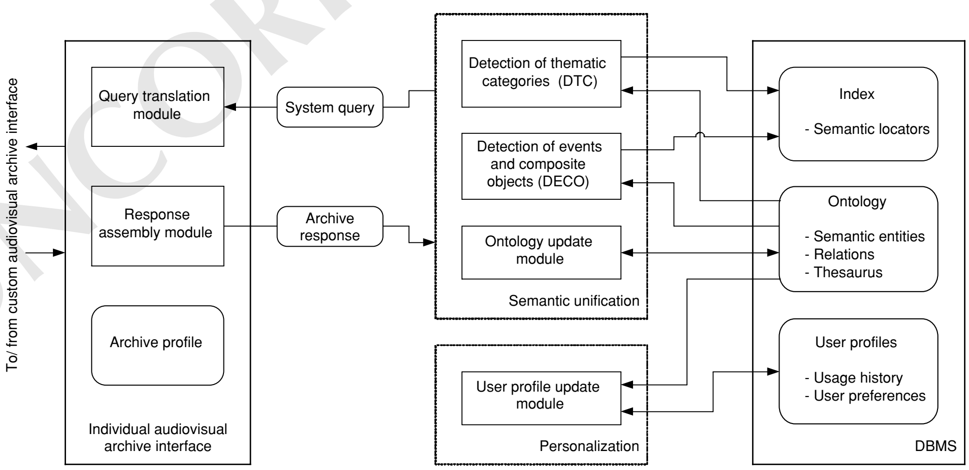

Figure 12.16 The system at update mode of operation

As already mentioned, all update procedures may be performed globally for the entire content of the audiovisual archives at regular intervals or whenever the audiovisual content of an archive is updated. In the latter case, which is preferable due to low bandwidth and computational cost, the update process is *incremental*, i.e. only the newly inserted audiovisual unit descriptions need to be retrieved and processed.

Prior to any indexing process, the content of the ontology may be updated with the aid of the *ontology update* module. The main goal of this module is to update the thesaurus according to any changes in the detailed semantic relations of the ontology, as DECO and DTC rely on correct input from the thesaurus views of knowledge in order to operate. Moreover, the definitions of semantic entities of the encyclopedia need to be updated, especially when the content of the audiovisual archives is dramatically changed.

12.4.1 Detection of Events and Composite Objects

The DECO is executed in a two-pass process for each multimedia document that has to be indexed. In the first pass the audiovisual archive is queried and the full description of a document that has not yet been indexed, or whose description has been altered, is retrieved. The individual archive interface assures that the structure that arrives at the central system is compliant with the MPEG-7 multimedia content description standard, thus allowing a unified design and operation of the indexing process to follow. The DECO module scans the MPEG-7 description and identifies mentioned semantic entities by their definitions in the ontology. Links between these semantic elements and the document in question are added to the index; weights are added depending on the location of the entity in the description scheme and the degree of matching between the description and the actual entity definition in the ontology.

In the second pass, the DECO module works directly on the semantic indexing of documents, attempting to detect events and composite objects that were not directly encountered in the document descriptions, but the presence of which can be inferred from the available indexing information. The second pass of the DECO process further enriches the semantic indexing of the documents.

Although the importance of the DECO as a stand-alone module is crucial for the operation of the overall system, one may also view it as a pre-processing tool for the following DTC procedure, since the latter uses the detected composite objects and events for thematic categorization purposes.

12.4.2 Detection of Thematic Categories

The DTC performs a matching between the archived material and the predefined thematic categories. It takes as input the indexing of each multimedia document, or document segment, as provided in the index by the DECO module, and analyses it in order to estimate the degree to which the document in question is related to each one of the thematic categories. Although the output of DTC is also stored in the index, as is the output of DECO, an important difference exists between the two: the weights in the output of DECO correspond to degrees of confidence, while the degrees in the output of DTC correspond to estimated degrees of association. Another important difference between the DECO and DTC modules is that whereas DECO searches for any semantic link between multimedia documents and semantic entities, DTC limits its operation to the case of thematic categories.

What makes the predefined categories, and accordingly the DTC process, so important, is the fact that through them a unified representation of multimedia documents originating from different audiovisual archives is possible. Thus, they have a major contribution to the semantic unification and unified access of diverse audiovisual sources, which is the main goal of the system.

12.4.3 Indexing Algorithms

The DTC and DECO run in offline time. They first run when the encyclopedia and audiovisual archive documents are constructed to create the index. Every time the audiovisual archives are enriched with new documents, or the annotation of existing documents is altered, the DTC and DECO run in order to update the index accordingly, processing only the updated segments of the audiovisual archives. Every time the ontology is updated the DTC and DECO run for all the audiovisual archives, and all the documents in each archive, in order to create a new index; an incremental update is not appropriate, as the new entities and new thesaurus knowledge views will result in different analysis of the document descriptions. In the following we provide more details on the methodologies utilized by these modules in the process of document analysis, after the first pass of DECO has completed, having provided an elementary semantic indexing of multimedia content.

The utilized view of the knowledge

The semantic encyclopedia contains *110 000* semantic entities and definitions of numerous MPEG-7 semantic relations. As one might expect, the existence of many relations leads to the dividing of the available knowledge among them, which in turn results in the need for the utilization of more relations than one for the meaningful analysis of multimedia descriptions. On the other hand, the simultaneous consideration of multiple semantic relations would pose an important computational drawback for any processing algorithm, which is not acceptable for a system that hopes to be able to accommodate large numbers of audiovisual archives and multimedia documents. Thus, the generation of a suitable view T in the thesaurus is required. For the purpose of analysing multimedia document descriptions we use a view that has been generated with the use of the following semantic relations:

- Part P, inverted.
- Specialization Sp.
- Example Ex. $Ex(a,b) > 0$ indicates that b is an example of a. For example, a may be 'player' and b may be 'Jordan'.
- Instrument Ins. $Ins(a,b) > 0$ indicates that b is an instrument of a. For example, a may be 'music' and b may be 'drums'.
- Location Loc, inverted. $L(a,b) > 0$ indicates that b is the location of a. For example, a may be 'concert' and b may be 'stage'.
- Patient Pat. $Pat(a,b) > 0$ indicates that b is a patient of a. For example, a may be 'course' and b may be 'student'.
- Property Pr, inverted. $Pr(a,b) > 0$ indicates that b is a property of a. For example, a may be 'Jordan' and b may be 'star'.

Thus, the view T is calculated as:

$$T = (Sp \cup P^{-1} \cup Ins \cup Pr^{-1} \cup Pat \cup Loc^{-1} \cup Ex)^{(n-1)}$$

The $(n-1)$ exponent indicates $n-1$ compositions, which are guaranteed to establish the property of transitivity for the view [10]; it is necessary to have the view in a closed transitive form, in order to be able to answer questions such as 'which entities are related to entity x?' in $O(logn)$ instead of $O(n^2)$ times, where $n = 110\,000$ is the count of known semantic entities. Alternatively, a more efficient methodology, targeted especially to sparse relations, can be utilized to ensure transitivity [9]. Based on the semantics of the participating relations, it is easy to see that T is ideal for the determination of the topics that an entity may be related to, and consequently for the analysis of multimedia content based on its mapping to semantic entities through the index.

The notion of context

When using an ontological description, it is the context of a term that provides its truly intended meaning. In other words, the true source of information is the co-occurrence of certain entities and not each one independently. Thus, in the process of content analysis we will have to use the common meaning of semantic entities in order to best determine the topics related to each examined multimedia document. We will refer to this as their *context*; in general, the term *context* refers to whatever is common among a set of elements. Relation T will be used for the detection of the context of a set of semantic entities, as explained in the remaining of this subsection.

As far as the second phase of the DECO and the DTC are concerned, a document d is represented only by its mapping to semantic entities via the semantic index. Therefore, the context of a document is again defined via the semantic entities that are related to it. The fact that relation T is (almost) an ordering relation allows us to use it in order to define, extract and use the context of a document, or a set of semantic entities in general.

Relying on the semantics of relation T, we define the context $K(s)$ of a single semantic entity $s \in S$ as the set of its antecedents in relation T, where S is the set of all semantic entities contained in the ontology. More formally, $K(s) = T(s)$, following the standard superset–subset notation from fuzzy relational algebra [9]. Assuming that a set of entities $A \subseteq S$ is crisp, i.e. all considered entities belong to the set with degree one, the context of the group, which is again a set of semantic entities, can be defined simply as the set of their common antecedents:

$$K(A) = \bigcap K(s_i),\, s_i \in A$$

Obviously, as more entities are considered, the context becomes narrower, i.e. it contains fewer entities and to smaller degrees:

$$A \supset B \rightarrow K(A) \subseteq K(B)$$

When the definition of context is extended to the case of fuzzy sets of semantic entities, this property must still hold. Taking this into consideration, we demand that, when A is a normal fuzzy set, the 'considered' context $\mathcal{K}(s)$ of s, i.e. the entity's context when taking its degree of participation in the set into account, is low when the degree of participation $A(s)$ is high, or when the context of the crisp entity $K(s)$ is low. Therefore:

$$cp(\mathcal{K}(s))=cp(K(s))\cap(S\cdot A(s))$$

where *cp* is an involutive fuzzy complement. By applying de Morgan's law, we obtain:

$$\mathcal{K}(s)=K(s)\cup cp(S\cdot A(s))$$

Then the overall context of the set is again easily calculated as:

$$K(A)=\bigcap \mathcal{K}(s_i),\ s_i\in A$$

Considering the semantics of the T relation and the process of context determination, it is easy to realize that when the entities in a set are highly related to a common meaning, the context will have high degrees of membership for the entities that represent this common meaning. Therefore, the height of the context $h(K(A))$, i.e. the greatest membership degree that appears in it, may be used as a measure of the semantic correlation of entities in set A. We will refer to this measure as *intensity* of the context.

Fuzzy hierarchical clustering and topic extraction

Before detecting the topics that are related to a document d, the set of semantic entities that are related to it needs to be clustered, according to their common meaning. More specifically, the set to be clustered is the support of the document:

$$^{0+}d=\{s\in S: I(s,d)>0\}$$

where $I{:}S\rightarrow D$ is the index and D is the set of indexed documents.

Most clustering methods belong to either of two general categories, partitioning and hierarchical. Partitioning methods create a crisp or fuzzy clustering of a given data set, but require the number of clusters as input. Since the number of topics that exist in a document is not known beforehand, partitioning methods are inapplicable for the task at hand; a hierarchical clustering algorithm needs to be applied. Hierarchical methods are divided into agglomerative and divisive. Of those, the first are more widely studied and applied, as well as more robust. Their general structure, adjusted for the needs of the problem at hand, is as follows:

1. When considering document d, turn each semantic entity $s\in^{0+}d$ into a singleton, i.e. into a cluster c of its own.
2. For each pair of clusters c_1, c_2 calculate a degree of association $CI(c_1,c_2)$. The CI is also referred to as cluster similarity measure.
3. Merge the pair of clusters that have the best CI. The best CI can be selected using the *max* operator.
4. Continue at step 2 until the termination criterion is satisfied. The termination criterion most commonly used is the definition of a threshold for the value of the best degree of association.

The two key points in hierarchical clustering are the identification of the clusters to merge at each step, i.e. the definition of a meaningful measure for *CI*, and the identification of the optimal terminating step, i.e. the definition of a meaningful termination criterion.

When clustering semantic entities, the ideal association measure for two clusters c_1, c_2 is one that quantifies their semantic correlation. In the previous we have defined such a measure: the intensity of their common context $h(K(c_1 \cup c_2))$. The process of merging should terminate when the entities are clustered into sets that correspond to distinct topics. We may identify this case by the fact that no pair of clusters will exist with a common context of high intensity. Therefore, the termination criterion shall be a threshold on the *CI*.

This clustering method, being a hierarchical one, will successfully determine the count of distinct clusters that exist in ${}^{0+}d$. Still, it is inferior to partitioning approaches in the following senses:

1. It only creates crisp clusters, i.e. it does not allow for degrees of membership in the output.
2. It only creates partitions, i.e. it does not allow for overlapping among the detected clusters.

Both of the above are great disadvantages for the problem at hand, as they are not compatible with the task's semantics: in real life, a semantic entity may be related to a topic to a degree other than 1 or 0, and may also be related to more than one distinct topics. In order to overcome such problems, we apply a method for fuzzification of the partitioning. Thus, the clusters' scalar cardinalities will be corrected, so that they may be used later on for the filtering of misleading entities.

Each cluster is described by the crisp set of semantic entities $c \subseteq {}^{0+}d$ that belong to it. Using those, we may construct a fuzzy classifier, i.e. a function C_c that measures the degree of correlation of a semantic entity s with cluster c. Obviously a semantic entity s should be considered correlated with c, if it is related to the common meaning of the semantic entities in it. Therefore, the quantity

$$Cor_1\ (c,s) = h(K(c \cup \{s\}))$$

is a meaningful measure of correlation. Of course, not all clusters are equally compact; we may measure cluster compactness using the similarity among the entities they contain, i.e. using the intensity of the clusters' contexts. Therefore, the aforementioned correlation measure needs to be adjusted, to the characteristics of the cluster in question:

$$C_c(s) = \frac{Cor_1(c, s)}{h(K(c))} = \frac{h(K(c \cup \{s\}))}{h(K(c))}$$

Using such classifiers, we may expand the detected crisp partitions, to include more semantic entities and to different degrees. Partition c is replaced by cluster c^{fuzzy}:

$$c^{fuzzy} = \sum_{s \in {}^{0+}d} s/C_c(s)$$

Obviously $c^{fuzzy} \supseteq c$.

The process of fuzzy hierarchical clustering has been based on the crisp set ${}^{0+}d$, thus ignoring fuzziness in the semantic index. In order to incorporate this information when calculating the 'final' clusters that describe a document's content, we adjust the degrees of membership for them as follows:

$$c^{final}(s) = t(c^{fuzzy}(s), I(s,d)), \forall s \in {}^{0+}d$$

where t is a t-norm. The semantic nature of this operation demands that t is an Archimedean norm [11]. Each one of the resulting clusters corresponds to one of the distinct topics of the document. Finally, once the fuzzy clustering of entities in a multimedia document's indexing has been performed, DTC and DECO can use the results in order to produce their own semantic output.

In order for DTC to determine the topics that are related to a cluster c^{final}, two things need to be considered: the scalar cardinality of the cluster $|c^{final}|$ and its context. Since context has been defined only for normal fuzzy sets, we need to first normalize the cluster as follows:

$$c^{normal}(s) = \frac{c^{final}(s)}{h(c^{final}(s))}, \forall s \in {}^{0+}d$$

Obviously, semantic entities that are not contained in the context of c^{normal} cannot be considered as being related to the topic of the cluster. Therefore:

$$R_T(c^{final}) \subseteq R_T{}^*(c^{normal}) = w(K(c^{normal}))$$

where w is a weak modifier. Modifiers, which are also met in the literature as *linguistic hedges*, are used to adjust mathematically computed values so as to match their semantically anticipated counterparts.

In the case where the semantic entities that index document d are all clustered in a unique cluster c^{final}, then $R_T(d) = R_T{}^*(c^{normal})$ is a meaningful approach. On the other hand, when multiple clusters are detected, then it is imperative that cluster cardinalities are considered as well.

Clusters of extremely low cardinality probably only contain misleading entities, and therefore need to be ignored in the estimation of $R_T(d)$. On the contrary, clusters of high cardinality almost certainly correspond to the distinct topics that d is related to, and need to be considered in the estimation of $R_T(d)$. The notion of 'high cardinality' is modelled with the use of a 'large' fuzzy number $L(\cdot)$. $L(a)$ is the truth value of the proposition 'a is high', and, consequently, $L(|b|)$ is the truth value of the preposition 'the cardinality of cluster b is high'.

The set of topics that correspond to a document is the set of topics that correspond to each one of the detected clusters of semantic entities that index the given document.

$$R_T(d) = c^{final} \in G(R_T(c^{final}))$$

where $\cup$ is a fuzzy co-norm and G is the set of fuzzy clusters that have been detected in d. The topics that are related to each cluster are computed, after adjusting membership degrees according to scalar cardinalities, as follows:

$$R_T(c^{final}) = R_T{}^*(c^{normal}) \cdot L(|c^{final}|)$$

It is easy to see that $R_T(s{,}d)$ will be high if a cluster c^{final}, whose context contains s, is detected in d, and additionally, the cardinality of c^{final} is high and the degree of membership of s in the context of the cluster is also high (i.e. if the topic is related to the cluster and the cluster is not comprised of misleading entities).

The DECO module, on the other hand, relies on a different view of the ontology that is constructed using only the specialization and example relations in order to take advantage of the findings of the fuzzy clustering of index terms. In short, DECO relates to each document the entities that are in the context of the detected clusters. In this framework the context is estimated using the same methodology as above, but instead of the T view of the knowledge we utilize one that contains only information extracted from the example, part and specialization relations.

12.5 Query Analysis and Processing

At the online mode of operation the system receives user queries from the end-user interfaces and serves them in a semantic and timely manner, based primarily on the information stored in the index and the ontology (Figure 12.17). Specifically, the semantic part of the query is analysed by the query analysis module in order to be mapped to a suitable set of semantic entities from the ontology; the entities of this set can then be mapped by the search engine to the corresponding multimedia documents, as the latter are indicated by the index. In the cases

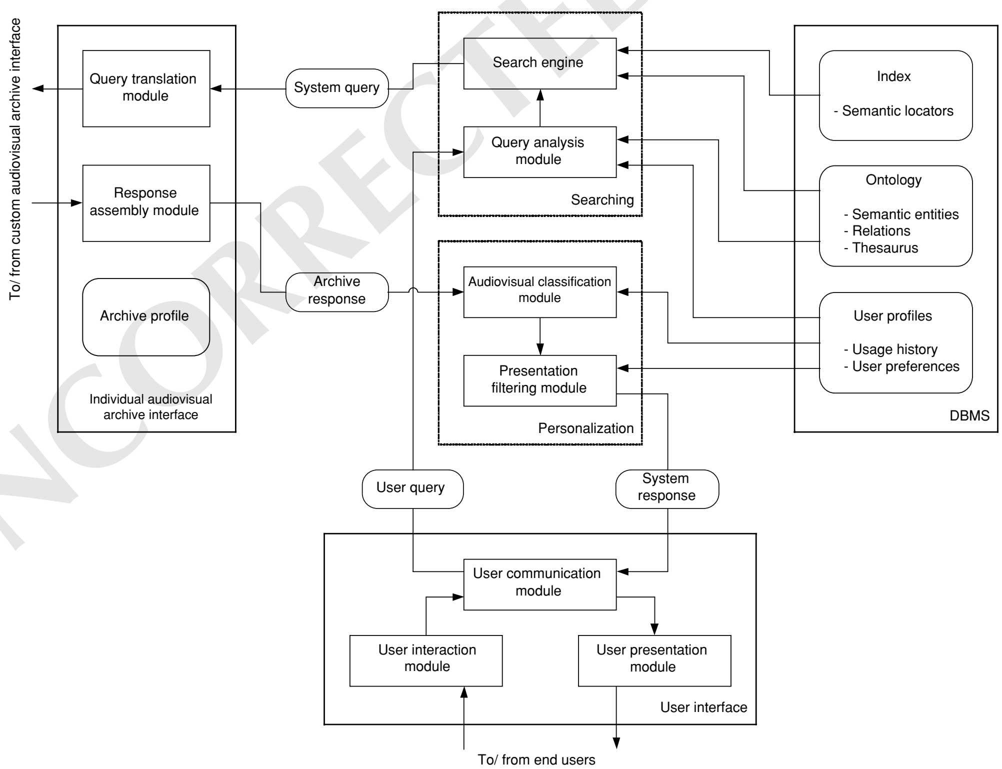

Figure 12.17 The system at query mode of operation

that the user query contains a structural part as well, or when the metadata part of the user profile is to be used during the personalization of the response, the audiovisual archives may have to be queried as well for the MPEG-7 annotations of the selected multimedia documents (it is the archive interface that takes care of the translation of the archive's custom DS to the MPEG standard, based on the information stored in the archive profile).

The weighted set of documents selected through this process is then adapted to the user that issued the query by the personalization subsystem, using the preferences defined in the active user profile. Of course, as the system aims to be the mediator for searches in audiovisual archives, it also supports the consideration of metadata in all the steps of searching and personalizing the results; still, the emphasis and novel contribution is found in the ability for semantic treatment of the user query, the multimedia documents and the user profiles, as it is exactly this characteristic that allows for the unified access to multiple and diverse audiovisual archives.

Focusing more on the searching procedure itself, we start by clarifying that both the user query and the index are fuzzy, meaning that the user can supply the degree of importance for each term of the query, and that the set of associated semantic entities for each document also contains degrees of association, as provided by the DECO and DTC modules. Consequently, the results of the searching procedure will also have to be fuzzy [12]; the selected documents are sorted by estimated degree of relevance to the user query, and in a later step according to relevance to the user preferences, and the best matches are presented (first) to the user.

It is possible that a query does not match a given index entry, although the document that corresponds to it is relevant to the query. For example, a generalization of a term found in a document may be used in the query. This problem is typically solved with the use of a fuzzy thesaurus containing, for each term, the set of its related ones. The process of enlarging the user's query with the associated terms is called query expansion; it is based on the associative relation A of the thesaurus, which relates terms based on their probability of coexisting in a document [13, 14].

To make query expansion more intelligent, it is necessary to take into account the meaning of the terms [15]. In order to be able to use the notion of context, as defined in the previous subsection, to estimate and exploit the common meaning of terms in the query, we need to map the query to the set of semantic entities in the ontology; this task is referred to as query interpretation, as it extracts the semantics of the terms of the user query. Finally, the utilization of a statistically generated associative thesaurus for query expansion, although a common and generally accepted practice in textual information retrieval, is avoided in this work, as this approach is known to overpopulate the query with irrelevant terms, thus lowering the precision of the response [16]; instead, we define and use a view of the ontology that is based strictly on partially ordering fuzzy relations, such as the specialization, the part and the example relation; the ordering properties of the considered relations make the resulting view more suitable for the definition and estimation of the context of a set of semantic entities.

12.5.1 Context-Sensitive Query Interpretation

As we have already mentioned, the definitions of semantic entities in the ontology contain sequences of labels, each one providing a different textual form of the semantic entity, possibly in more than one language. Matching those to the terms in the user query, we can acquire the semantic representation of the query. Of course, in most cases this is far from trivial:

the mapping between terms and semantic entities is a many-to-many relation, which means that multiple possible semantic interpretations exist for a single textual query. As a simple example, let us consider the case of the term ‘element’. At least two distinct semantic entities correspond to it: ‘element1’, which is related to chemistry, and ‘element2’, which is related to XML. Supposing that a user query is issued containing the term ‘element’, the system needs to be able to automatically determine to which semantic entity in the ontology the term should be mapped, in order to retrieve the corresponding multimedia documents from the index.

In the same example, if the remaining terms of the query are related to chemistry, then it is quite safe to suppose that the user is referring to semantic entity ‘element1’ rather than to semantic entity ‘element2’. This implies that the context of the query can be used to facilitate the process of semantic entity determination in the case of ambiguities. However, the estimation of the query context, as described in the previous section, needs as input the representation of the query as a fuzzy set of entities, and thus cannot be performed before the query interpretation is completed.

Consequently, query interpretation needs to take place simultaneously with context estimation. We follow the following method: let the textual query contain the terms $\{t_i\}$ with $i = 1, \ldots, T$. Let also t_i be the textual description of semantic entities $\{s_{ij}\}$ with $j = 1, \ldots, T_i$. Then, there exist $N_Q = \prod_i T_i$ distinct combinations of semantic entities that may be used for the representation of the user’s query; for each one of those we calculate the corresponding context.

As already explained, the intensity of the context is a semantic measure of the association of the entities in a set. Thus, out of the candidate queries $\{q_k\}$, where $k = 1, 2, \ldots, N_Q$, the one that produces the most intense context is the one that contains the semantic entities that are most related to each other; this is the combination that is chosen as output of the process of query interpretation:

$$q = q_i \in \{q_1, \ldots, q_{N_Q}\} : h(q_i) \geq h(q_j) \forall q_j \in \{q_1, \ldots, q_{N_Q}\}$$

This semantic query interpretation is exhaustive, in the sense that it needs to consider all possible interpretations of a given query. Still, this is not a problem in the framework where it is applied as:

- queries do not contain large numbers of terms;
- the number of distinct semantic entities that may have a common textual description is not large;
- the gain in the quality of the semantic content of the interpreted query, as indicated by the difference in the precision of the system response, is largely more important than the added computational burden.

12.5.2 Context-Sensitive Query Expansion

The process of query expansion enriches the semantic query, in order to increase the probability of a match between the query and the document index. The presence of several semantic entities in the query defines a context, which we use in order to meaningfully direct the expansion process, so that it generates expanded queries that provide enhanced recall in the result, without suffering the side effect of poor precision.

As will become obvious from the presentation of the process of matching the query to the index, optimal results can only be acquired if the origin of the new entities in the expanded query is known; in other words, we will need to know to which entity in the initial query each new entity corresponds. Thus, in query expansion, we replace each semantic entity s with a set of semantic entities $X(s)$; we will refer to this set as the expanded semantic entity.

In a more formal manner, we define the expanded entity as $X(S_i) = \sum_i s_{ij}/x_{ij}$, using the sum notation for fuzzy sets; the weight x_{ij} denotes the degree of significance of the entity s_j in $X(s_i)$. We compute it using the semantic query q, the context $K(q)$ of the query, and the *In* relation of the thesaurus; the *In* relation has resulted from the combination of the *Sp*, P and the *Ex* relations as $In = (Sp \cup P^{-1} \cup Ex)^{(n-1)}$.

In a query expansion that does not consider the context, the value of x_{ij} should be proportional to the weight w_i and the degree of inclusion $I(s_i, s_j)$. Therefore, in that case we would have $x_{ij} = w_{ij} = w_i In(s_i, s_j)$. In a context-sensitive query expansion, on the other hand, x_{ij} increases monotonically with respect to the degree to which the context of s_j is relative to the context of the query. We use the value:

$$h_j = \max\left(\frac{h(In(s_j) \cap K(q))}{h_q}, h_q\right)$$

to quantify this relevance. We additionally demand that the following conditions be satisfied by our query expansion method:

- x_{ij} increases monotonically with respect to w_{ij}
- $h_j = 1 \rightarrow x_{ij} = w_{ij}$
- $h_q = 0 \rightarrow x_{ij} = w_{ij}$
- $h_q = 1 \rightarrow x_{ij} = w_{ij} h_j$
- x_{ij} increases monotonically with respect to h_j.

Thus, we have:

$$x_{ij} = \max(h_j, c(h_q))w_{ij} = w_i In(s_i, s_j) \max(h_j, c(h_q))$$

The fuzzy complement c in this relation is Yager's complement with a parameter of 0.5.

12.5.3 Index Matching

The search engine, supposing that the query is a crisp set (i.e. entities in the query do not have weights) and that no expansion of the query has preceded, uses the semantic query q, which is a fuzzy set of semantic entities, and the document index I, which is a fuzzy relation between the set of semantic entities S and the set of documents D, to produce the result r; r is again a fuzzy set on D. When the query is comprised of a single semantic entity s, then the result is simply the respective row of I, i.e. $r(q) = I(s)$. When, on the other hand, the query contains more than one semantic entity, then the result is the set of documents that contain all the semantic entities, or, more formally:

$$r(q) = \bigcap_{s_i \in q} r(s_i)$$

Generalizing this formula to the case when query expansion has preceded, we should include in the result set only documents that match all the expanded entities of the query. Therefore, it is imperative that independent search results are first computed for each expanded entity separately, and then combined to provide the overall result of the search process.

Considering the way expanded entities are calculated, it is rather obvious that a document should be considered to match the expanded entity when it matches any of the terms in it. Moreover, the percentage of semantic entities that a document matches should not make a difference (it is the same if the document matches only one or all of the semantic entities in the same expanded entity, as this simply indicates that the document is related to just one of the entities in the original query). Consequently, we utilize the *max* operator in order to join results for a single expanded entity:

$$r\,(X(s_i)) = \bigcup_{s_j \in X(s_i)} r(s_j)$$

or, using a simpler notation:

$$r\,(X(s_i)) = X(s_i) \circ I$$

On the other hand, results from distinct entities are treated using an intersection operator, i.e. only documents that match all of the entities of the query are selected.

$$r(q) = \bigcap_{s_i \in q} r\,(X(s_i))$$

Unfortunately, this simple approach is limiting; it is more intuitive to select the documents that match all of the terms in the query first, followed by those documents that match fewer of the query terms. The effect of this limitation becomes even more apparent when the initial query is not crisp, i.e. when the absence of an entity that was assessed as unimportant by the user prevents an otherwise relevant document from being included in the results.

Thus, we follow a more flexible approach for the combination of the results of the matching of distinct entities with the semantic index. Specifically, we merge results using an ordered weighted average operator [17, 18], instead of the *min* operator. The selection of weights for the OWA operator is a monotonically increasing one. The required flexibility is achieved by forcing the degree of the last element to be smaller than one. Thus, the chosen family of OWA operators behaves as a 'soft' intersection on the intermediate results.

12.6 Personalization

Due to the massive amount of information that is nowadays available, the process of information retrieval tends to select numerous documents, many of which are barely related to the user's wish [19]; this is known as *information overload*. The reason is that an automated system cannot acquire from the query adequate information concerning the user's wish. Traditionally, information retrieval systems allow the users to provide a small set of keywords describing their wishes, and attempt to select the documents that best match these keywords. Although the information contained in these keywords rarely suffices for the exact determination of user wishes, this is a simple way of interfacing that users are accustomed to; therefore, we need to investigate ways to enhance retrieval, without altering the way they specify their request. Consequently, information about the user wishes needs to be found in other sources.

Personalization of retrieval is the approach that uses the information stored in user profiles, additionally to the query, in order to estimate the user's wishes and thus select the set of relevant documents [20]. In this process, the query describes the user's current search, which is the *local interest* [21], while the user profile describes the user's preferences over a long period of time; we refer to the latter as *global interest*.

12.6.1 Personalization Architecture

During the query mode (Figure 12.17), the audiovisual classification module performs ranking (but not filtering) of the retrieved documents of the archive response based on semantic preferences contained within the user profiles. The semantic preferences consist of user interests and thematic categories preferences. At the presentation filtering module further ranking and filtering is performed according to the metadata preferences such as creation, media, classification, usage, access and navigation preferences (e.g. favourite actors/directors or preference for short summaries).

The entire record of user actions during the search procedure (user query, retrieved documents, documents selected as relevant) is stored in the usage history of the specific user; this information is then used for tracking and updating the user preferences. The above actions characterize the user and express his/her personal view of the audiovisual content. The user profile update module takes these transactions as input during update mode (Figure 12.16) of operation and, with the aid of the ontology and the semantic indexing of the multimedia documents referred to in the usage history, extracts the user preferences and stores them in the corresponding user profile.

12.6.2 The Role of User Profiles

When two distinct users present identical queries, they are often satisfied by different subsets of the retrieved documents, and to different degrees. In the past, researchers have interpreted this as a difference in the perception of the meaning of query terms by the users [20]. Although there is definitely some truth in this statement, other more important factors need to be investigated.

Uncertainty is inherent in the process of information retrieval, as terms cannot carry unlimited information [21], and, therefore, a limited set of terms cannot fully describe the user's wish; moreover, relevance of documents to terms is an ill-defined concept [22]. The role of personalization of information retrieval is to reduce this uncertainty, by using more information about the user's wishes than just the local interest.

On the other hand, the user profile is not free of uncertainty either, as it is generated through the constant monitoring of the user's interaction; this interaction contains inherent uncertainty which cannot be removed during the generation of the user profile. Nevertheless, user profiles tend to contain less uncertainty than user queries, as long as the monitoring period is sufficient and representative of the user's preferences.

Therefore, a user profile may be used whenever the query provides incomplete or insufficient information about the user and their local interest. However, it is the query that describes the user's local preference, i.e. the scope of their current interaction. The profile is not sufficient on its own for the determination of the scope of the current interaction, although it contains valuable information concerning the user's global interest. Therefore, the user profile cannot totally dominate over the user query in the process of information retrieval.

The above does not imply that the degree to which the query dominates the retrieval process may be predefined and constant. Quite the contrary, it should vary in a manner that optimizes the retrieval result, i.e. in a manner that minimizes its uncertainty. We may state this more formally by providing the following conditions:

1. When there is no uncertainty concerning the user query, the user profile must not interfere with the process of information retrieval.
2. When the uncertainty concerning a user query is high, the user profile must be used to a great extent, in order to reduce this uncertainty.
3. The degree to which the user profile is used must increase monotonically with respect to the amount of uncertainty that exists in the user query, and decrease monotonically with respect to the amount of uncertainty that exists in the user profile.

The above may be considered as minimal guidelines or acceptance criteria for the way a system exploits user profiles in the process of information retrieval.

12.6.3 Audiovisual Content Classification

The audiovisual classification module receives as input a weighted set of documents. It is the set of documents retrieved based on the user's actions. Ranking has already taken place based on the query itself, producing an objective set of selected documents. The goal of the audiovisual classification module is to produce a subjective set of selected documents, i.e. a set that is customized to the preferences of the user who posed the query.

As has already been mentioned, preferences for thematic categories and semantic interests are utilized in different manners by the module of audiovisual classification. We elaborate on both in the following.

Exploitation of Preferences for Thematic Categories

Through the index, each document that the system handles has been related to the system's predefined thematic categories to various degrees by the DTC module. Moreover, for each user profile, degrees of preference are mentioned for all of the predefined thematic categories. These may be manually predefined by the user or by an expert when referring to static user profiles, or created based solely on the monitoring of user actions in the case of dynamic user profiles.

The audiovisual classification module examines each document in the set of result of the search independently. The user's degree of preference for each one of the thematic categories that are related to a document is checked. If at least one of them is positive, negative preferences are ignored for this document. The document is promoted (its rank is increased), to the degree that it is related to some thematic category and that thematic category is of interest to the user. The re-ranking is performed using a parameterized weak modifier, where the intensity of the preference sets the parameter. A typical choice is:

$$r'(q)_d = \sqrt[1+x]{r(d)}$$

where x is given by:

$$x = h\left(I^{-1}(d) \cap P_T^{+}\right)$$

and P_T^{+} are the positive thematic category preferences of the user. In other words, when the document is related to a thematic category to a high degree, and the preference of the user for that category is intense, then the document is promoted in the result. If the preference is less intense or if the document is related to the preference to a smaller extent, then the adjusting of the document's rank is not as drastic.

Quite similarly, when the document is only related to negative preferences of the user, then a strong modifier is used to adjust the document's ranking in the results:

$$r'(q)_d = (r(d))^{1+x}$$

where x is given by:

$$x = h\left(I^{-1}(d) \cap P_T^{-}\right)$$

and P_T^{-} are the negative thematic category preferences of the user.

Exploitation of Semantic Interests

Semantic interests offer a much more detailed description of user preferences. Their drawback is that they are mined with a lesser degree of certainty and they are more sensitive to context changes. This is why they are utilized more moderately.

The simple and composite entities contained in each document need to be compared with the ones contained in the user's profile. This comparison, though, needs to be performed in a context-sensitive manner. Specifically, in the case that no context can be detected in the query, the whole set of user interests is considered. If, on the other hand, the query context is intense, interests that do not intersect with the context should not be considered, thus eliminating inter-preference noise. Thus, semantic interests can only refine the contents of the result set moderately, always remaining in the same general topic. Ranks are updated based on similarity measures and relativity to context, as well as the preferences' intensities.

The relevance of an interest to the context of the query is quantified using the intensity of their common context, while the adjusting of ranks is performed similarly to the case of thematic categories, with x defined as:

$$x = \max(x_i)$$
$$x_i = h\left(I^{-1}(d) \cap P_i^{+}\right) \cdot h_{K(q) \cap P_i^{+}}$$

where P_i^{+} is one of the positive interests of the user.

As far as negative interests are concerned, there is no need to consider the context before utilizing them. What is needed, on the other hand, is to make sure that they do not overlap with any of the in-context positive interests, as this would be inconsistent. Within a specific query context one may demand, as a minimum consistency criterion, that the set of considered interests does not contain both positive and negative preferences for the same semantic entities.

When ‘correcting’ the view of the user profile that is acquired by removing out-of-context interests, the following need to be obeyed:

- Positive interests are generally extracted with greater confidence. Therefore, positive interests are treated more favourably than negative ones, in the process of creating a consistent view of the profile.
- Obviously, if only positive interests correspond to a specific semantic entity, then their intensities must not be altered. Likewise, if only a negative interests corresponds to a specific semantic entity, then its intensity must not be altered.
- In general, the intensities of positive preferences should increase monotonically with respect to their original intensity, and decrease monotonically with respect to the original intensity of the corresponding negative preference, and vice versa.

These guidelines lead to the generation of a valid, i.e. consistent, context-sensitive user profile [8].

12.6.4 Extraction of User Preferences

Based on the operation of the DECO and DTC modules, the system can acquire in an automated manner and store in the index the fuzzy set of semantic entities and thematic categories (and consequently topics) that are related to each document. Still, this does not render trivial the problem of semantic user preference extraction. What remains is the determination of the following:

1. Which of the topics that are related to documents in the usage history are indeed of interest to the user and which are found there due to coincidental reasons?
2. To which degree is each one of these topics of interest to the user?

As far as the main guidelines followed in the process of preference extraction are concerned, the extraction of semantic preferences from a set of documents, given their topics, is quite similar to the extraction of topics from a document, given its semantic indexing. Specifically, the main points to consider may be summarized in the following:

1. A user may be interested in multiple, unrelated topics.
2. Not all topics that are related to a document in the usage history are necessarily of interest to the user.
3. Documents may have been recorded in the usage history that are not of interest to the user in some way (these documents were related to the local interest of the user at the time of the query, but are not related to the user’s global interests.)

Clustering of documents

These issues are tackled using similar tools and principles to the ones used to tackle the corresponding problems in multimedia document analysis and indexing. Thus, once more, the basis on which the extraction of preferences is built is the context. The common topics of documents are used in order to determine which of them are of interest to the user and which exist in the usage history coincidentally.

Moreover, since a user may have multiple preferences, we should not expect all documents of the usage history to be related to the same topics. Quite the contrary, similarly to semantic entities that index a document, we should expect most documents to be related to just one of the user's preferences. Therefore, a clustering of documents, based on their common topics, needs to be applied. In this process, documents that are misleading (e.g. documents that the user chose to view once but are not related to the user's global interests) will probably not be found similar with other documents in the usage history. Therefore, the cardinality of the clusters may again be used to filter out misleading clusters.

For reasons similar to those in the case of thematic categorization, a hierarchical clustering algorithm needs to be applied. Thus, the clustering problem is reduced to the selection of merging and termination criteria. As far as the former is concerned, two clusters of documents should be merged if they are referring to the same topics. As far as the latter is concerned, merging should stop when no clusters remain with similar topics.

What is common among two documents $a,b \in D$, i.e. their common topics, can be referred to as their common context. This can be defined as:

$$K(a, b) = I^{-1}(a) \cap I^{-1}(b)$$

A metric that can indicate the degree to which two documents are related is, of course, the intensity (height) of their common context. This can be extended to the case of more than two documents, in order to provide a metric that measures the similarity between clusters of documents:

$$Sim(c_1, c_2) = h(K(c_1, c_2))$$

where $c_1,c_2 \subseteq H^+ \subseteq D$ and:

$$K(c_1, c_2) = \bigcap_{d \in c_1 \cup c_2} I^{-1}(d)$$

$$H = \{H^+, H^-\}$$

where H is a view of the usage history, comprising H^+, the set of documents that the user has indicated interest for, and H^-, the set of documents for which the user has indicated some kind of dislike. *Sim* is the degree of association for the clustering of documents in H^+. The termination criterion is again a threshold on the value of the best degree of association.

Extraction of interests and of preferences for thematic categories

The topics that interest the user, and should be classified as positive interests, or as positive preferences for thematic categories, are the ones that characterize the detected clusters. Degrees of preference can be determined based on the following parameters:

1. The cardinality of the clusters. Clusters of low cardinality should be ignored as misleading.
2. The weights of topics in the context of the clusters. High weights indicate intense interest. This criterion is only applicable in the case of user interests.

Therefore, each one of the detected clusters c_i is mapped to a positive interest as follows:

$$U_i^+ = L(c_i) \cdot K(c_i)$$
$$K(c_i) = \bigcap_{d \in c_i} I^{-1}(d)$$

where U_i^+ is the interest and $L(c_i)$ is a 'large' fuzzy number. When it comes to the case of thematic categories, they are generally extracted with higher degrees of confidence, but a larger number of documents need to correspond to them before the preference can be extracted. More formally, in the case of thematic categories the above formula becomes:

$$P_{T_i} = w(L'(c_i) \cdot K(c_i))$$

where w is a weak modifier and L' is a 'very large' fuzzy number.

The information extracted so far can be used to enrich user requests with references to topics that are of interest to the user, thus giving priority to related documents. What it fails to support, on the other hand, is the specification of topics that are known to be uninteresting for the user, as to filter out, or down-rank, related documents. In order to extract such information, a different approach is required.

First of all, a document's presence in H^- has a different meaning than its presence in H^+. Although the latter indicates that at least one of the document's topics is of interest to the user, the former indicates that, most probably, all topics that are related to the document are uninteresting to the user.

Still, topics may be found in H^- for coincidental reasons as well. Therefore, negative interests should be verified by the repeated appearance of topics in documents of H^-:

$$U^- = \sum s_i / u_i^-$$
$$u_i^- = L\Big(\sum_{d \in H^-} I(s_i, d)\Big)$$

Finally, both for positive and negative thematic category preferences and interests, due to the nature of the document analysis and document clustering processes, multiple semantic entities with closely related meanings are included in each preference and to similar degrees. In order to avoid this redundancy, a minimal number of semantic entities have to be selected and stored for each preference. This is achieved by forming a maximum independent set of entities in each preference, with semantic correlation (as shown by the height of the common context) indicating the proximity between two semantic entities. As initially connected we consider the pairs of entities whose common context has a height that exceeds a threshold.

12.7 Experimental Results

This section describes the quantitative performance analysis and evaluation of the audiovisual document retrieval capabilities of the proposed system, essentially verifying the responses to specific user queries against 'ground truth' to evaluate retrieval performance. In the sequel, the methodology followed for constructing the ground truth, carrying out the experiments and analysing the results is outlined. The overall results are presented and conclusions are drawn.

12.7.1 Methodology

Methodology for information retrieval performance evaluation

Performance characterization of audiovisual content retrieval often borrows from performance figures developed over the past 30 years for probabilistic text retrieval. Landmarks in the text retrieval field are the books [23], [24] and [25]. Essentially all information retrieval (IR) is about cluster retrieval: the user having specified a query would like the system to return some or all of the items, either documents, images or sounds, that are in some sense part of the same semantic cluster, i.e. the relevant fraction of the database with respect to this query for this user. The ideal IR system would quickly present the user some or all of the relevant material and nothing more. The user would value this ideal system as being either 100% effective or being without (0%) error.

In practice, IR systems are often far from ideal: the query results shown to the user, i.e. the finite list of retrieved items, generally are incomplete (containing some retrieved relevant items but without some missed relevant items) and polluted (with retrieved but irrelevant items). The performance is characterized in terms of precision and recall. *Precision* is defined as the number of retrieved relevant items over the number of total retrieved items. *Recall* is defined as the number of retrieved relevant items over the total number of relevant items:

$$p = precision = \frac{relevant\ retrieved\ items}{retrieved\ items}$$

$$r = recall = \frac{relevant\ retrieved\ items}{relevant\ items}$$

The performance for an 'ideal' system is to have both high precision and high recall. Unfortunately, they are conflicting entities and cannot practically assume high values at the same time. Because of this, instead of using a single value of precision and recall, a *precision–recall* (PR) graph is typically used to characterize the performance of an IR system. This approach has the disadvantage that the length, or *scope*, of the retrieved list, or visible size of the query results, is not displayed in the performance graph, whereas this scope is very important to the user because it determines the amount of items to be inspected and therefore the amount of time (and money) spent in searching. The scope is the main parameter of economic effectiveness for the user of a retrieval system. Moreover, even though well suited for purely text-based IR, a PR graph is less meaningful in audiovisual content retrieval systems where recall is consistently low or even unknown, in cases where the ground truth is incomplete and the cluster size is unknown. In these cases the *precision–scope* (PS) graph is typically employed to evaluate retrieval performance.

In [26], another performance measure is proposed: the *rank* measure, leading to *rank–scope* (RS) graphs. The rank measure is defined as the average rank of the retrieved relevant items. It is clear that the smaller the rank, the better the performance. While PS measurements only care if a relevant item is retrieved or not, RS measurements also care about the rank of that item. Caution must be taken when using RS measurements, though. If system A has higher precision and lower rank measurements than system B, then A is definitely better than B, because A not only retrieves more relevant images than B, but also all those retrieved images are closer to the top in A than in B. But if both precision and rank measurements of A are higher than those of B, no conclusion can be made.

Equally important is the degradation due to a growing database size, i.e. lowering the fraction of relevant items resulting in overall lower precision–recall values. A comparison between two information retrieval systems can only be done well when both systems are compared in terms of equal *generality*:

$$g = _{generality} = \frac{relevant\ items}{all\ items}$$

Although there is a simple method of minimizing the number of irrelevant items (by minimizing the number of retrieved items to zero) and a simple one to minimize the number of missed relevant items (by maximizing the number of retrieved items up to the complete database), the optimal length of the result list depends upon whether one is satisfied with finding one, some or all relevant items.

The parameterized *error* measure of [23]:

$$E = _{Error} = 1 - \frac{1}{a(1/p) + (1-a)(1/r)}$$

is a normalized error measure where a low value of a favours recall and a high value of a favours precision. E will be 0 for an ideal system with both precision and recall values at 1 (and in that case irrespective of a). The setting of $a = 0.5$ is typically chosen, a choice giving equal weight to precision and recall and giving rise to the normalized symmetric difference as a good single number indicator of system performance. Moreover, an intuitive best value of 1 (or 100%) is to be preferred; this is easily remedied by inverting the [1,0] range. Thus, *effectiveness* is defined as:

$$e = _{effectiveness} = 1 - E(a = 0.5) = \frac{1}{(1/2p) + (1/2r)}$$

Evaluation procedure

Based on the above methodology and guidelines for retrieval performance evaluation, a series of experiments was carried out to evaluate the system's retrieval performance. Evaluation was based on ground truth in a well-defined experimental setting allowing the recovery of all essential parameters. The evaluation test bed was the prototype of the experimental system developed in the framework of the FAETHON IST project, which served as a mediator for unified semantic access to five archives with documents annotated in different languages and using diverse data structures. The five archives were the Hellenic Broadcast Corporation (ERT) and Film Archive Greece (FAG) from Greece, Film Archiv Austria (FAA) and Austrian Broadcasting Corporation (ORF) and Alinary from Italy.

The first step was to develop the ground truth against which all retrieval results had to be compared in order to measure retrieval performance. The ground truth in general included a set of semantic test queries and the corresponding sets of 'ideal' system responses for each query. There are three actions involved in this process:

1. Since the content of the five participating archives belongs in general to varying thematic categories, the set of queries had to relate to concepts that were common in all, or most, archives, so that corresponding responses were sufficiently populated from all archives.

2. Once the set of test queries was specified, the 'ideal' set of responses (list of audiovisual documents to be returned) had to be specific with corresponding degrees of confidence, or equivalently ranked. This was repeated for each test query and for each participating archive.
3. Finally, in order to include personalization in retrieval performance evaluation, separate response sets were prepared for a limited number of pre-specified user profiles, differing in their semantic preferences.

Due to the large size of the archives, the existing knowledge base and user profiles, caution was taken to limit the number of test queries and user preferences, and even use a subset only of existing archive content. All ground-truth information was manually generated so the above selections were crucial in making the test feasible in terms of required effort.

Subsequently, the test queries were fed into the system and corresponding responses were recorded from all archives and for each specified user profile; these were automatically tested against ground-truth data in order to make comparisons. The latter were performed according to the performance evaluation criteria and measures specified above. In particular, for each test query:

1. the retrieved documents had been directly recorded;
2. the relevant documents, with associated degrees of relevance, were available from ground-truth data;
3. the relevant retrieved documents were calculated as the intersection of the two above sets of documents;
4. the total number of all documents in the system index was a known constant.

Thus, all quantities required for the calculation of precision, recall, generality error and effectiveness were available. Additionally to the above described methodology, wherever relevance or confidence values were available, such as in the list of retrieved documents, all cardinality numbers, or total number of documents, were replaced by the respective sums of degrees of relevance.

Finally, all precision and recall measurements were recorded for each experiment, i.e. for each test query and user profile. Average precision–recall values were calculated per query and user profile, and corresponding PR graphs were drawn and studied. The overall results are presented and conclusions on the system's retrieval performance are drawn.

12.7.2 Experimental Results and Conclusions

Experimental settings

Following the procedure described above, which is in turn based on the methodology of the previous subsection, we are going to calculate the quadruple $\{p, r, g, e\}$. The number of all audiovisual documents is a known constant, i.e. $d = 1005$. Because of the reasons mentioned in the evaluation procedure above, the parameterization could not be as extensive as theory demands. So the following compromises were made, in order to achieve reliable results within a feasible evaluation period of time:

1. Only three different user profiles were taken into account; one without any semantic preferences, a second with interest in politics and a third with interest in sports.

2. The ground truth was built manually for the following semantic queries: 'Navy', 'Football match', 'Elections', 'Aircraft' and 'Olympic games'.
3. Since the system returns the same audiovisual documents for a user query irrelative to the user's semantic preferences and only changes their degree of relevance (audiovisual classification re-ranks documents, it does not filter them), we consider that the retrieved documents are only these which have a degree above 30%. So, depending on the user's preferences, we retrieve a different number of documents.

Based on the above constraints, we executed the five semantic queries for each one of the three user profiles; thus we acquired fifteen different result lists. In Tables 12.1–12.5 we demonstrate the results, grouped by each query.

Retrieval results

Query 1: 'Navy'

For this specific query, we see that the results do not vary a lot among the three different user profile (Table 12.1). This is expected, since the word 'Navy' semantically is not related more to one of the two pre-selected semantic user preferences. We also notice that the system tends to respond with less accuracy in favour of better recall numbers.

Query 2: 'Football Match'

This time the query is related to the thematic category 'sports and athletics', which makes distinguishable better results for the user whose semantic preferences are set to 'sports'. This can be seen from the higher effectiveness number (Table 12.2).

Query 3: 'Elections'

The user who made the semantic query 'Elections' expects to retrieve some audiovisual content related to elections in the first place and to politics in extension. Consequently the user with preference in the topic 'politics' gets both higher precision and recall indices than the user

Table 12.1 The estimated parameters are demonstrated for each user profile for the semantic query 'Navy' against the ground truth

Profile	Relevant	Retrieved	Relative and retrieved	p	r	e	g
None	18	26	16	0.615	0.889	0.727	
Politics	18	50	15	0.750	0.833	0.789	0.018
Sports	18	22	15	0.682	0.833	0.750	

Table 12.2 The estimated parameters are demonstrated for each user profile for the semantic query 'Football match' against the ground truth

Profile	Relevant	Retrieved	Relative and retrieved	p	r	e	g
None	59	58	42	0.724	0.712	0.718	
Politics	59	48	41	0.854	0.695	0.765	0.059
Sports	59	86	52	0.605	0.881	0.717	

Table 12.3 The estimated parameters are demonstrated for each user profile for the semantic query 'Elections' against the ground truth

Profile	Relevant	Retrieved	Relative and retrieved	*p*	*r*	*e*	*g*
None	49	56	35	0.625	0.714	0.667	
Politics	49	51	40	0.784	0.816	0.800	0.049
Sports	49	39	31	0.795	0.633	0.705	

with no special interests, and the effectiveness index is higher than that of all the other users (Table 12.3).

Query 4: 'Aircraft'

In this query, we observed lower figures for the precision (Table 12.4). In other words, the system returned among the relevant documents many other irrelevant (according to the ground truth). This is depicted in Figure 12.18, with the points corresponding to this query being in the lower right part of the diagram.

Query 5: 'Olympic games'

In comparison to the previous queries this one, 'Olympic games', is related to more audiovisual documents in all five archives, a fact apparent from the generality index as well. This time we had results with higher precision compared to the recall for the queries performed by the user with no special interest and the user with interest in politics (Table 12.5), something which is also shown in Figure 12.18, where the corresponding points in the graph are in the left upper part of the diagram.

Figure 12.18 summarizes all of the above queries in the form of a PR graph. In the same figure we demonstrate the three different user profiles used. Five points were drawn, since we performed five queries. The lines were drawn after polynomial interpolation and with the use of statistical techniques, in order to show the exponential decrease of the PR graph. Although few sound conclusions can be drawn from a diagram that has resulted from so few queries, one can easily make at least two observations:

- System responses are generally better when user queries are also considered, as in this case more information is available to the system for the selection of relevant documents.
- The PR diagrams are generally located in the upper right corner of the 0–100 space, a fact that is not common in PR diagrams, thus indicating that the utilization of ontological semantic

Table 12.4 The estimated parameters are demonstrated for each user profile for the semantic query 'Aircraft' against the ground truth

Profile	Relevant	Retrieved	Relative and retrieved	*p*	*r*	*e*	*g*
None	17	29	16	0.552	0.941	0.696	
Politics	17	22	15	0.682	0.882	0.780	0.017
Sports	17	20	13	0.650	0.765	0.696	

Table 12.5 The estimated parameters are demonstrated for each user profile for the semantic query 'Olympic games' against the ground truth

Profile	Relevant	Retrieved	Relative and retrieved	*p*	*r*	*e*	*g*
None	95	66	55	0.833	0.579	0.683	
Politics	95	62	56	0.903	0.589	0.712	0.095
Sports	95	105	79	0.752	0.832	0.790	

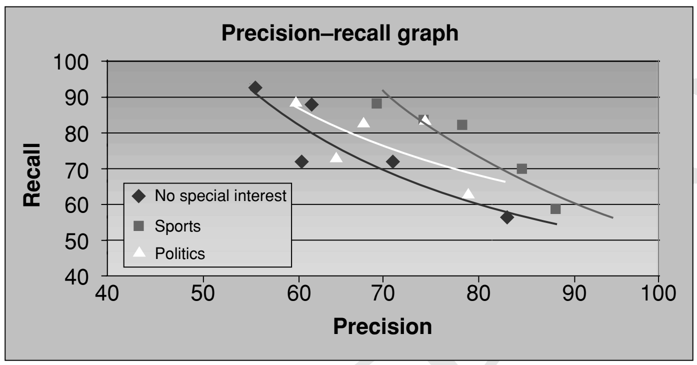

Figure 12.18 The estimated parameters from the five queries are demonstrated for each user profile against the ground truth

knowledge in the processing of user queries, documents and profiles can greatly contribute to the enhancement of the acquired results.

12.8 Extensions and Future Work

The key aspect of the FAETHON developments has been the generation and use of metadata in order to provide advanced content management and retrieval services. The web will change drastically in the following years and become more and more multimedia enabled, making already complex content management tasks even more complex and requiring solutions based on Semantic Web technologies. Unlike today, content itself will be a commodity in a future web, making the use of metadata essential. Content providers, for instance, will have to understand the benefits obtained from the systematic generation of metadata; service providers will have to accept metadata as the basis on which to build new services; and the producers of software tools for end users will redirect their imagination towards more appropriate integration of application software with web content, taking advantage of metadata. These developments clearly present some challenging prospects, in technological, economic, standardization and business terms.

Another interesting perspective of FAETHON's developments is the personalization, based on usage history, of the results of content retrieval. Personalization software is still in its infancy, which means there are no turnkey solutions. Solutions using agent technologies still have a lot of hurdles to overcome. To improve this scenario, additional technology approaches need to be evaluated and areas of improvement identified. In both perspectives, clearly FAETHON made some interesting steps on the correct route, and its developments are currently influencing the next research activities in the area of semantic-based knowledge systems.

The long-term market viability of multimedia services requires significant improvements to the tools, functionality and systems to support target users. *aceMedia* seeks to overcome the barriers to market success, which include user difficulties in finding desired content, limitations in the tools available to manage personal and purchased content, and high costs to commercial content owners for multimedia content processing and distribution, by creation of means to generate semantic-based, context- and user-aware content, able to adapt itself to users' preferences and environments. aceMedia will build a system to extract and exploit meaning inherent to the content in order to automate annotation and to add functionality that makes it easier for all users to create, communicate, find, consume and reuse content.

aceMedia targets knowledge discovery and embedded self-adaptability to enable content to be self-organising, self-annotating, self-associating, more readily searched (faster, more relevant results), and adaptable to user requirements (self-reformatting). aceMedia introduces the novel concept of the Autonomous Content Entity (ACE), which has three layers: content, its associated metadata, and an intelligence layer consisting of distributed functions that enable the content to instantiate itself according to its context (e.g. network, user terminal, user preferences). The ACE may be created by a commercial content provider, to enable personalized self-announcement and automatic content collections, or may be created in a personal content system in order to make summaries of personal content, or automatically create personal albums of linked content.

Current multimedia systems and services do not support their target users well enough to imagine their long-term market expansion, without significant improvements to the tools, functionality and systems available to the user. The aceMedia project sets out to offer solutions to the barriers to market success, which include:

- users being unwilling to sign up for commercial multimedia services when they are unable to readily find desired content, and are limited in the tools available to manage that content once purchased;
- commercial content owners unwilling to invest resources (usually staff) in content provision due to the high costs associated with multimedia content processing and distribution;
- individual users of multimedia acquisition and storage systems being unable to manage their ever-growing personal content collections, but the only tools available to assist them meet only a part of their needs, and the complexity of such tools usually sites them in the realm of the professional user

To address these problems, aceMedia focuses on generating value and benefits to end users, content providers, network operators and multimedia equipment manufacturers, by introducing, developing and implementing a system based on an innovative concept of knowledge-assisted, adaptive multimedia content management, addressing user needs. The main technological objectives are to discover and exploit knowledge inherent to the content in order to make content more relevant to the user, to automate annotation at all levels, and to add functionality to ease content creation, transmission, search, access, consumption and reuse. In addition, available user and terminal profiles, the extracted semantic content descriptions and advanced mining methods will be used to provide user and network adaptive transmission and terminal optimized rendering.

The current World Wide Web is, by its function, the syntactic web where structure of the content has been presented while the content itself is inaccessible to computers. The next generation of the web (the Semantic Web) aims to alleviate such problems and provide specific solutions targeting the concrete problems. Web resources will be much easier and more readily accessible by both humans and computers with the added semantic information in a machine-understandable and machine-processable fashion. It will have much higher impact on e-work and e-commerce than the current version of the web. There is, however, still a long way to go to transfer the Semantic Web from an academic adventure into a technology provided by the software industry.

Supporting this transition process of ontology technology from academia to industry is the main and major goal of *Knowledge Web*. This main goal naturally translates into three main objectives given the nature of such a transformation:

1. Industry requires immediate support in taking up this complex and new technology. Languages and interfaces need to be standardized to reduce the effort and provide scalability to solutions. Methods and use cases need to be provided to convince and to provide guidelines for how to work with this technology.
2. Important support to industry is provided by developing high-class education in the area of the Semantic Web, web services and Ontologies.
3. Research on ontologies and the Semantic Web has not yet reached its goals. New areas such as the combination of the Semantic Web with web services realizing intelligent web services require serious new research efforts.

In a nutshell, it is the mission of Knowledge Web to strengthen the European software industry in one of the most important areas of current computer technology: Semantic Web enabled e-work and e-commerce. Naturally, this includes education and research efforts to ensure the durability of impact and support of industry.

References

[1] A. Delopoulos, S. Kollias, Y. Avrithis, W. Haas, K. Majcen, Unified intelligent access to heterogeneous audiovisual content. In *Proceedings of International Workshop on Content-Based Multimedia Indexing (CBMI '01)*, Brescia, Italy, 19–21 September 2001.

[2] J. Hunter, Adding multimedia to the Semantic Web: building an MPEG-7 ontology. In *Proceedings of First Semantic Web Working Symposium, SWWS '01*, Stanford University, CA, July 2001.

[3] F. Nack, A. Lindsay, Everything you wanted to know about MPEG-7: part 1. *IEEE Multimedia*, **6**(3), 65–77, 1999.

[4] F. Nack, A. Lindsay, Everything you wanted to know about MPEG-7: Part 2. *IEEE Multimedia*, **6**(4), 64–73, 1999.

[5] T. Sikora, The MPEG-7 visual standard for content description—an overview. *IEEE Transactions on Circuits and Systems for Video Technology*, special issue on MPEG-7, **11**(6), 696–702, 2001.

[6] ISO/IEC JTC1/SC29/WG11 N5231, MPEG-21 Overview, Shanghai, October 2002.

[7] G. Akrivas, G. Stamou, Fuzzy semantic association of audiovisual document descriptions. In *Proceedings of Int errational Workshop on Very Low Bitrate Video Coding (VLBV),* 2001.

[8] M. Wallace, G. Akrivas, G. Stamou, S. Kollias, Representation of user preferences and adaptation to context in multimedia content-based retrieval. In *Workshop on Multimedia Semantics, Accompanying 29th Annual Conference on Current Trends in Theory and Practice of Informatics (SOFSEM)*, Milovy, Czech Republic, November 2002.

[9] M. Wallace, S. Kollias, Computationally efficient incremental transitive closure of sparse fuzzy binary relations. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Budapest, Hungary, July 2004. IEEE, 2004.

[10] G. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic, Theory and Applications*. Prentice Hall, Eaglewood Cliffs, NJ, 1995.

[11] M. Wallace, G. Akrivas, G. Stamou, Automatic thematic categorization of documents using a fuzzy taxonomy and fuzzy hierarchical clustering. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, St Louis, MO, May 2003. IEEE, 2003.

[12] D.H. Kraft, G. Bordogna, G. Passi, Fuzzy set techniques in information retrieval. In J.C. Berdek, D. Dubois, H. Prade (eds) *Fuzzy Sets in Approximate Reasoning and Information Systems*. Kluwer Academic, Boston, MA, 2000.

[13] S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis*. Kluwer Academic, Dordrecht, 1990.

[14] W.-S. Li, D. Agrawal, Supporting web query expansion efficiently using multi-granularity indexing and query processing. *Data and Knowledge Engineering*, **35**(3), 239–257, 2000.

[15] D.H. Kraft, F.E. Petry, Fuzzy information systems: managing uncertainty in databases and information retrieval systems. *Fuzzy Sets and Systems*, **90**, 183–191, 1997.

[16] G. Akrivas, M. Wallace, G. Andreou, G. Stamou, S. Kollias, Context-sensitive semantic query expansion. In *IEEE International Conference on Artificial Intelligence Systems (ICAIS)*, Divnomorskoe, Russia, September 2002. IEEE, 2002.

[17] R.R. Yager, J. Kacprzyk, *The Ordered Weighted Averaging Operators: Theory and Applications*, pp. 139–154. Kluwer Academic, Norwell, MA, 1997.

[18] R.R. Yager, Families of OWA operators. *Fuzzy Sets and Systems*, **59**, 125–148, 1993.

[19] P.M. Chen, F.C. and Kuo, An information retrieval system based on a user profile. *Journal of Systems and Software,* **54**, 3–8, 2000.

[20] C.L. Barry, User-defined relevance criteria: an exploratory study. *Journal of the American Society for Information Science*, **45**, 149–159, 1994.

[21] C.H. Chang, C.C. Hsu, Integrating query expansion and conceptual relevance feedback for personalized Web information retrieval. *Computer Networks and ISDN Systems*, **30**, 621–623, 1998.

[22] D.H. Kraft, G. Bordogna, G. Passi, Information retrieval systems: where is the fuzz? In *Proceedings of IEEE International Conference on Fuzzy Systems*, Anchorage, Alaska, May 1998. IEEE, 1998.

[23] C.J. van Rijsbergen, *Information Retrieval*, 2nd edn. Butterworths, London, 1979.

[24] G. Salton, M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1982.

[25] R.A. Baeza-Yates, B.A. Ribeiro-Neto, *Modern Information Retrieval*. ACM Press/Addison-Wesley, Reading, MA, 1999.

[26] J. Huang, S. Kumar, M. Mitra,W.-J. Zhu, R. Zabih, Image indexing using color correlogram. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1997.