Combining Multimedia Retrieval and Text Retrieval to Search Structured Documents in Digital Libraries

Andreas Henrich and **Günter Robbert**
Otto-Friedrich University of Bamberg, Faculty of Social and Economic Sciences
Feldkirchenstraße 21, D-96045 Bamberg, Germany
Email: {Andreas.Henrich|Guenter.Robbert}@sowi.uni-bamberg.de

Abstract

Digital Libraries usually contain a large collection of structured multimedia documents. At present text may be dominant in many applications, however the relevance of other media types such as image, audio and video increases steadily. An important functionality of a digital library in this respect is the retrieval of relevant multimedia documents or relevant parts of multimedia documents. To this end, an efficient combination of automatic text retrieval, retrieval in meta data (usually created manually) and content based retrieval on multimedia data is needed. In this position paper, we will argue, that a sophisticated query language for an (at least structurally) object-oriented database is a suitable basis for an application specific user-interface built on top of it. We will sketch the requirements for such a query language and present our research approach.

1. Motivation

Structured multimedia documents have to be maintained in many different application areas. Although most of the ideas presented in this position paper are applicable in all these areas it seems to be appropriate to sketch the concrete scenario we are concerned with in order to clarify the motivation behind our approach. The concrete scenario comprises the management of the multimedia teaching content of an open university. The situation is as follows: Many lecturers create teaching material for different target groups in a joint effort. This material is maintained in a common repository. In this context complete courses hardly fit for different target groups, however, smaller chunks may be reusable. These chunks can be modules covering a certain topic or even single images or animations. In order to support the lecturers with the creation of course material in the sketched scenario, we propose to support the retrieval of reusable components by a powerful retrieval service.

The architecture of the system is sketched in figure 1.

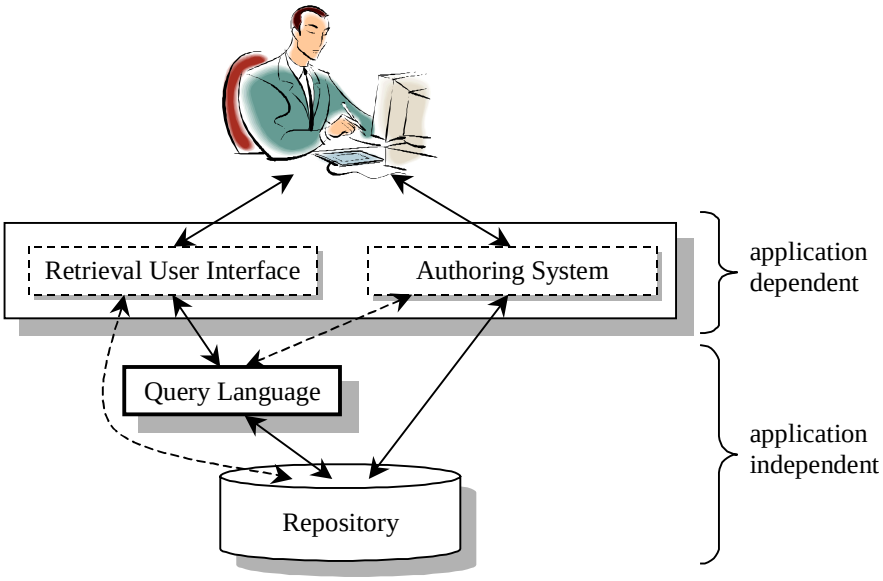

Figure 1 Rule of the query language in our scenario

The architecture is based on a object-oriented repository. This repository is responsible for object creation and deletion and aspects such as versioning, notification or the navigational access to the objects. The query

language enables a declarative set-oriented access to the objects. The repository and the query language are application independent. On top of them the application specific retrieval user interface is realized. In our context this user interface is integrated with the authoring system used to create the multimedia teaching content.

The present paper discusses the requirements for the application independent query language in the sketched architecture. Of course this query language (QL) firstly has to fulfil the usual requirements for query languages, such as descriptiveness, orthogonality, extensibility, computational completeness, … (cf. [HS91]). On the other hand a general purpose QL for digital libraries maintaining multimedia data in addition should fulfil requirements stemming from the specific character of the maintained structured multimedia data. Thereby the focus of the query language is more on the expressive power, the comprehensiveness and the genericity of the language. The adequacy and the usability for the end-user have to be assured by the application specific front-end.

In the following we will formulate seven important requirements for the query language and sketch potential solutions.

2. Dealing with Structured Documents

Requirement: Since multimedia documents in digital libraries usually are *structured documents*, the QL should allow to deal with structured documents in an appropriate way.

The fact that the maintained documents are complex structured objects brings up various interesting research issues. (1) The QL must allow to search for arbitrary granules ranging from whole documents over intermediate chunks to single media objects. Furthermore it might be appropriate in some situations to leave the result granule open – if a whole document is relevant, the whole document should be returned, but if only a small chunk of a document is relevant, this chunk should be returned. (2) Many properties of an object are not directly attached to the object itself, but to its components. For example, the text comprising a chapter will usually be stored in separate text objects associated with the chapter object via links or relationships and not as an attribute of the chapter object itself. (3) Additional information about an atomic media object can be found in its vicinity. Exploiting the structure of a multimedia document this "vicinity" can be addressed navigating one link up and then down to the sibling components.

In our opinion regular path expressions are extremely useful when dealing with structured documents. Generally spoken a regular path expression defines the set of all items which can be reached from the current object via a path matching the regular path expression. The technical details of the regular path expressions obviously depend on the underlying data model. However, there has to be a powerful means to define single matching links by *regular link definitions* and in addition means to define *sequences* and *iterations* of links are needed. To this end, all features of the underlying data model should be addressable. In PCTE [PCTE94] this does for example mean that the regular link definitions should allow to address link types as well as link categories and link attributes. Multiple link definitions can be concatenated and iteration facilities can be used. In our query language POQL^{MM} [Hen96], which is presently based on the PCTE data model, the regular path expression `{@c}/_.related/[{c}]*/->`. for example means that the first link of a matching path must be the traversal of a composition link in the reverse direction (`{@c}` matches all links for which the reverse link has category *composition*). Thereafter a link of type `related` with an arbitrary key attribute value has to be traversed. Finally an arbitrary number of composition links can be part of a matching path (`[{c}]*`). Although the complex syntax of the regular path expressions might be a bit confusing at first glance, they are extremely useful when dealing with structured documents. For example these expressions allow to define the text associated to a chapter via an expression defining a set of text attribute values. Furthermore we can easily address the text attributes in the vicinity of an image navigating one step up in the document structure and then down to the text objects.

In addition the use of rather generic link categories allows to deal with abstract object types in a flexible way. For example we can define the text associated with an object of type `object` as the concatenation of the string attributes of its direct an indirect components. Then we can use e.g. the vector space model to rank these objects and due to the normalization in the vector space model, which tries to eliminate the influence of the document length, this does automatically yield a ranking of different result granules.

One might argue that PCTE – which presently is the basis of POQL^{MM} – is not widely-used, but the same ideas can be easily applied to XML and in particular to XLink [XLi99] and XPath [XPa99] which have concepts such as roles and titles of links which can be addressed in a query language similar to link categories and link types in PCTE. Furthermore XLink knows bi-directional links which can also be exploited in regular path expressions.

3. Feature Extraction and Segmentation

Requirement: With multimedia data the semantics is usually given implicitly in the media objects, for example an image might show an important person or represent a certain mood. Therefore the QL should allow to *extract features* from the media objects potentially describing their semantics. Furthermore it should provide means to *subdivide media objects* such as images or videos into semantically coherent segments and to address the derived segments in a query.

At first glance it seems to be a good idea to maintain some descriptive information together with media objects in order to enable an efficient retrieval of these objects. Approaches in this direction range from the maintenance of a simple keyword list to sophisticated data models for the descriptive information as with MPEG-7 [Mar00]. Unfortunately a media object usually bears a nearly infinite semantics. Therefore each approach which tries to describe the semantics of a media object with some meta data must be fragmentary. Moreover practical experience shows that the quality of the meta data actually maintained in applications is usually unsatisfactory.

The consequence of this considerations is not that we do not need meta data – on the contrary meta data and especially manually created meta data will be the backbone of multimedia retrieval systems for the next decades. Nevertheless a query language for multimedia data should not restrict itself to meta data. In addition a broad variety of feature extracting operators for different media object types has to be integrated into the QL. For example colour histograms or texture information can be extracted from images, text can be extracted from audio data via speech recognition and captions can be extracted from video (cf. table 1). Besides these feature extracting operators segmentation operators should be provided segmenting an image into its potential conceptual subregions or subdividing a video into shots. These features are needed whenever meta data falls short. Especially segmentation allows to focus on details of media objects which might be disregarded when the media object is considered only as a whole. For example consider an image of a tennis match. The image might show Boris Becker and Andre Agassi in the 3rd set of a game in Rom. All these aspects are covered by the meta data. But in the background of the image there is a red Ferrari which is a bonus for the match winner. And now we are for some reasons searching for images in our archive which show the sponsoring activities of Ferrari in sports. In this case meta data is insufficient. But it might be possible to find the Ferrari when we search for segments with a similar colour and a similar texture compared to a given image of a Ferrari.

	feature extraction	*segmentation*	*conversion*
audio	musical instrument detection speaker recognition	interrupt detection	speech recognition
video	motion detection	shot detection	key frame extraction caption recognition
image	colour histogram texture	region based segmentation edge based segmentation	OCR

Table 1 Example operators or feature extraction, segmentation, and conversion for different media types

Although a lot of successful work on automatic feature extraction has been done in the areas of image, video and audio processing (see e.g. [RHC99,AY99,Sch97]), the integration of feature extraction and segmentation techniques in a QL still bears a lot of open questions with respect to retrieval quality, materialization of extracted features or query optimisation.

4. Partial Match and Similarity Queries

Requirement: Because of the vagueness in the interpretation of the media objects and in the expression of the users information need *partial match* and *similarity queries* should be facilitated.

When the QL provides operators for feature extraction it has to support similarity queries on these feature values as well. To this end, two different approaches are conceivable: The first approach would be to integrate the feature extraction and the similarity calculation in one operator. An operator `col_sim` which receives two images and returns a colour similarity value based e.g. on histogram intersection would be an example for this approach. The advantages of this approach are that it reduces the number of operators in the QL and that it prevents the application of unsuitable combinations of feature extractors and similarity measures. The second approach is to separate feature extraction and similarity calculation into different operators. In this case the QL would provide a variety of similarity measures implementing different metrics for similarity searches. This includes e.g. Euclidean distance, histogram intersection and the cosine coefficient (cf. [HR00]). In a concrete QL the calculation of the colour similarity may look like `hist_intersect(col_hist(A),col_hist(B))` in this case. Although this approach induces somewhat more lengthy query formulations, it has the advantage of

more freedom with the application of similarity measures. For example we could use some type of image transformation before applying the similarity measure or we can compare images with a constant vector.

As a consequence the second approach seems better suited with respect to our architecture sketched in figure 1, because here the QL is not directly used by humans, but via an API by higher level applications. From a rather extreme position we could argue that it is also in the responsibility of this application to use similarity measures and feature extraction techniques which have a high conformity with human perception. Nevertheless this remains an interesting research field, and the QL has to provide the required base operations for these purposes.

Another important aspect of the requirement sketched above is that at least part of the meta data maintained in the repository has to be regarded as vague data. Consequently we could use a probabilistic model (cf. [Fuh00]) for the complete retrieval process. However, this is not our research focus at the moment.

5. Integrating Text Retrieval Facilities

Requirement: Multimedia documents usually contain substantial textual parts and on the other hand comprehensive techniques for text retrieval have been developed, in the area of information retrieval (IR). Therefore a QL for multimedia data should admit the use of *text retrieval techniques* and especially the combination of text retrieval techniques with retrieval techniques for other media types.

In a structured document the most fertile information about an image, an audio or a video/animation can be found in the text objects associated with this media object. Consequently a QL for multimedia data must (1) allow to address the text in the vicinity of a media object in a flexible way, (2) incorporate advanced pattern matching facilities as well as more sophisticated IR models to deal with the text data and (3) integrate these text retrieval facilities in a homogeneous way to allow for a combination with attribute value based selection predicates (e.g. addressing meta data stored in "normal" attributes) as well as with the multimedia retrieval facilities.

For the first aspect regular path expressions as discussed in section 2 can be applied. A typical path expression in this case traverses "one step up and then down again". In POQL^{MM} this looks like `{@c}/[{c}]+`. Obviously such a path expression yields a heterogeneous result set. This problem can be solved with an additional type condition for the objects in the result set. In the next step IR techniques have to be applied to the result of the path expression. To this end, POQL^{MM} for example contains IR operations which can be applied to a set of objects. In fact the concatenation of all text attributes of these objects is considered as the corresponding text. If `A` represents a set with images, we can use `D_vector(A:{@c}/[{c}]+/->.)` to calculate a vector space representation for this image based on the text in the "vicinity" of the image. We plan to extend these facilities to exploit position information or sequence information present in the data model in the next version of POQL^{MM}.

Due to the integration of the information retrieval facilities into a declarative OQL-oriented query language, text-based retrieval can be combined with attribute value based selection predicates and multimedia retrieval facilities in a flexible way.

6. Combining Different Similarity Measures

Requirement: Due to the heterogeneous nature of multimedia applications there is no single combination of different similarity measures fitting well in all application areas. Therefore the query language must facilitate a *flexible combination of different similarity measures* trimmed well for application specific needs.

The integration of retrieval facilities for different media types into a closed descriptive query language allows to combine the facilities in a flexible way. For example we can combine a similarity search based on text with a similarity search on image features. This can be done in a rather straightforward way multiplying or adding similarity values calculated for the text similarity and the image similarity using the arithmetical operators of the QL. Another interesting approach is to use some type of fuzzy logic (cf. [Fag99]). All these approaches try to calculate new combined "similarity values" for the objects under consideration. Instead of this "object centred" approach we can also consider the different ranking lists resulting from the different similarity measures and combine these ranking lists into a combined ranking list. For the merging of the individual lists different strategies are conceivable. In general the problem can be interpreted as a selection or ranking problem for which all corresponding techniques from the area of decision theory can be applied. For example we could use one similarity criterion as the dominant one and consider the remaining criteria only when the first criterion yields the same value for a set of objects. An advantage of this approach is that it relieves us from the burden of performing some type of normalization among the different criteria. Nevertheless, the approach seems to be too extreme, because of the dominance of one criterion.

A more flexible approach, which does nevertheless not require a normalization among the criteria, is to use the ranks of the objects with respect to the single criteria. Let $r_{i,j}$ be the rank of object *i* with respect to criterion *j* ($j \in \{1,\ldots,m\}$ and $r_{i,j} \in \{1,2,\ldots\}$) and let w_j be the weight of criterion *j* representing its relative importance amongst the criteria. Then we can use the sum $\sum_{j=1}^{m} w_j \cdot \frac{1}{\sqrt{r_{i,j}}}$ to derive the combined ranking. In fact this is analogous to championships in sports where the results of different contests are combined giving a certain number of points for the different ranks. With our formula we assign 1 point for the 1st rank, 0.71 points for the 2nd rank, 0.58 points for the 3rd rank, and so forth. In addition the different "contests" (= similarity criteria) can be weighted with the values w_j.

To cope with situations where a single similarity criterion yields only a weak ordering – i.e. it yields the same similarity value for multiple objects – we can spread the points for these ranks equally over the objects with the same similarity value. Assume that the objects with numbers *a* to *b* in the ranking list for a single criterion have the same similarity value. In this case all these objects would get $\frac{1}{(b-a)+1} \cdot \sum_{k=a}^{b} \frac{1}{\sqrt{k}}$ points. In principle this allows to integrate Boolean conditions into the similarity considerations. To this end, we define that all objects fulfilling the Boolean condition have the similarity value 1 and all objects that doe not fulfil the condition have the similarity value 0.

The combination method sketched above is integrated in POQL^{MM} by the `combine`-operator. This operator can be interpreted as a generalized `sort`-operator. The `combine`-operator is applied to tuples typically resulting from a select statement, where the first components in the result tuples represent the similarity values for the single similarity criteria. After the `combine` keyword we have to define how the components of the tuples shall influence the sorting. For example `combine[(9,'-'),(7,'+'),(3,'+')]` defines, that the sorting with respect to the first component has to be done in descending order ("-"), whereas the sorting with respect to the second and the third component has to be done in ascending order ("+"). Furthermore the given numbers represent the values w_j.

7. Schema Independence

Requirement: For a general applicability the QL must not rely on a specific schema or a specific type of data modelling. Rather it should allow for precise queries *irrespective of the concrete underlying schema*. For example the query language should not assume a specific modelling of meta data or of the structure of a multimedia document.

The schema of a concrete multimedia system has to cover various aspects of the data, namely the structure of the documents, the different media types together with their registration data, the potential segmentation of media objects, interpretation data (meta data), potential user interaction, … At present there are various standardization efforts with respect to schemata for multimedia data – for example MPEG 7 [Mar00] or IMS [IMS00] to name only two prominent examples. From the perspective of a generic QL for multimedia data these schemata can be envisaged as benchmarks. The question is: Does the QL allow to address all features of the schemata in an appropriate way? On the other hand, a generic QL should not rely on a specific modelling of the data and especially of the meta data.

8. Performance

Requirement: Finally a good performance for all types of queries has to be assured.

With the requirements sketched in sections 2 to 7 in mind it might be tempting to design a sophisticated QL incorporating all types of sophisticated features. But the implementation aspects of such a QL must not be neglected. This comprises aspects such as index structures, query optimisation [HJ99] or the update of index structures (which is by no means trivial especially when derived attribute values have to be indexed [Hen97b]).

One main aspect in this direction are high dimensional index structures. There are various approaches to index structures for similarity searches. On the other hand, the combination of different similarity measures and the combination of similarity queries with standard fact conditions still form interesting research fields for index structures. It can be shown, that combined access structures can gain significant performance improvements (cf. [Hen98, Hen97a]).

With POQL^{MM} we are at present implementing an algorithm for the combine-operator based on the algorithms presented in [BL85], [PP97] and [GBK00]. Roughly spoken this algorithm performs parallel similarity searches on different access structures until the top positions in the resulting list are stable.

References

[AY99] Aslandogan, Y. Alp and Yu, Clement T.: *Techniques and Systems for Image and Video Retrieval*. TKDE 11(1): 56-63 (1999)

[BL85] Buckley, C. and Lewit, A.: *Optimization of inverted vector searches*. Proc. of the 8th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, pages 97-105, New York, 1985

[Fag99] Fagin, R.: *Combining Fuzzy Information from Multiple Systems*. JCSS 58(1): 83-99 (1999)

[GBK00] Güntzer, U. and Balke, W.-T. and Kießling, W.: *Optimizing Multi-Feature Queries for Image Databases*. In Proc. of the 26th Intl. Conf. on Very Large Databases (VLDB 2000), Cairo, Egypt, 2000

[Fuh00] Fuhr, N.: *Probabilistic Datalog: Implementing Logical Information Retrieval for Advanced Applications*. Journal of the American Society for Information Science 51(2), 2000, pages 95-110

[Hen96] Henrich, A.: *Document Retrieval Facilities for Repository-Based System Development Environments*; in: H.-P. Frei, D. Harman, P. Schäuble, R. Wilkinson (Eds.), Proc. 19th Annual Intl. ACM SIGIR Conference on Research and Development in Information Retrieval, p. 101-109, Zürich, 1996

[Hen97a] Henrich, A.: *A Common Access Structure for Standard Attributes and Document Representations in Vector Space*; in: P. Shoval, A. Silberschatz (Eds.), Proc. 3rd Intl. Workshop on Next Generation Information Technologies and Systems, p. 154-161, Neve Ilan, Israel, 1997

[Hen97b] Henrich, A.: *The Update of Index Structures in Object-Oriented DBMS*; Proc. 6th Intl. Conf. on Information and Knowledge Management (CIKM '97), ACM Press, p. 136-143, Las Vegas, 1997

[Hen98] Henrich, A.: *The LSD^h-Tree: An Access Structure for Feature Vectors*; in: Proc. 14th Intl. Conf. on Data Engineering (ICDE'98), p. 362-369, Orlando, Florida, USA, 1998

[HJ99] Henrich, A.; Jamin, S.: *On the Optimization of Queries containing Regular Path Expressions*, in: Ron Y. Pinter, Shalom Tsur (Eds.): Next Generation Information Technologies and Systems, Proc. 4th Intl. Workshop, NGITS'99, LNiCS, Vol. 1649, Springer, p. 58-75, Zikhron-Yaakov, Israel, 1999

[HR00] Henrich, A., Robbert, G.: *MARS: A Retrieval Service for Multimedia Authoring Environments*; in: Proceedings of Challenges of ADBIS-DASFAA Symposium on Advances in Databases and Information Systems, p. 88-98, Prague, Czech Republic, 2000

[HS91] Heuer, A. and Scholl, M.H.: *Principles of object-oriented query languages*; in: Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), GI-Fachtagung, volume 270 of Informatik Fachberichte, p. 178-197, Kaiserslautern, 1991. Springer, Berlin

[IMS00] *IMS Learning Resource Meta-data Information Model, Version 1.1 - Final Specification*, IMS Global Learning Consortium, Inc., 2000, `http://www.imsproject.org/metadata`

[Mar00] Martínez, J.M. (edt.): *Overview of the MPEG-7 Standard*, International Organization for Standardization, ISO/IEC JTC1/SC29/WG11, Coding of Moving Pictures and Audio, Geneva, May/June 2000

[PCTE94] *Portable Common Tool Environment – Abstract Specification/C Bindings/Ada Bindings*. ISO IS 13719-1/-2/-3, 1994

[PP97] Pfeifer, U. and Pennekamp, S.: *Incremental Processing of Vague Queries in Interactive Retrieval Systems*. in Fuhr, N; Dittrich, G. and Tochtermann, K. (eds.): Hypertext – Information Retrieval – Multimedia '97, p. 223-235, Dortmund, 1997, Universitätsverlag Konstanz

[RHC99] Rui, Yong and Huang, Thomas S. and Chang, Shih-Fu: *Image Retrieval: Current Techniques, Promising Directions and Open Issues*, Journal of Visual Communication and Image Representation, Vol. 10, 39-62, March, 1999

[Sch97] Schäuble, P.: *Multimedia Information Retrieval – Content-Based Information Retrieval from Large Text and Audio Databases*, Kluwer Academic Publishers, Boston, 1997.

[XLi99] *XML Linking Language (XLink)*, World Wide Web Consortium (W3C) Working Draft 26 July 1999, `http://www.w3.org/TR/xlink.html`

[XPa99] *XML Path Language (XPath)*, Version 1.0, World Wide Web Consortium (W3C) Recommendation 16 November 1999, `http://www.w3.org/TR/xpath.html`