FLEXIBLE QUERY FORMULATION
FOR MULTIMEDIA DOCUMENT RETRIEVAL

Panos Constantopoulos
Maria Theodoridou
Yannis Yeorgaroudakis

RCC/CSI/TR/1988/009

Computer Science Institute
Research Center of Crete
FLEXIBLE QUERY FORMULATION
FOR MULTIMEDIA DOCUMENT RETRIEVAL

Panos Constantopoulos
Maria Theodoridou
Yannis Yeorgaroudakis

January 1988

RCC/CSI/TR/1988/009
Technical Report Series
Computer Science Institute, Research Center of Crete
P.O.Box 1385, Heraklio, Crete, 711-10 Greece
Tel.: +30.81.221171, Telex: 262389 CCI GR
E-mail – UUCP: mcva!ariadne!maria

To be presented at EURINFO '88: First European Conference on Information Technology for Organizational Systems,
Athen, Greece, May 1988.
Flexible query formulation for multimedia document retrieval

P. Constantopoulos, M. Theodoridou, Y. Yeorgaroudakis

Institute of Computer Science
Research Center of Crete
Heraklion, Crete
GREECE

ABSTRACT

The design of a user interface for multimedia document retrieval is described. The interface provides flexible query formulation by exploiting all document components as well as associations between them and supports the resolution of user uncertainty by allowing dynamic query reformulation and extensive use of browsing.

1. Introduction

Storage and retrieval of multimedia documents is an increasingly important issue in information systems. In particular, strict requirements must be met in regard with retrieval effectiveness and flexibility.

Multimedia documents are structured collections of alphanumeric data (attributes and text), images, graphics and audio messages. Their size may vary from a few hundred kilobytes (mail messages, simple forms) to over a megabyte (complex documents with images and voice recordings) [Tsic83, IEEE85, Tsic85].

The filing of multimedia office documents can be considered to comprise three stages, corresponding to distinguishable important periods in the life cycle of the document. Initially, a document is private to a certain person, filed manually in that person's local space according to some private
organization scheme. During this stage, a document usually undergoes several changes. In the second stage, the document is made available to some authorized community and is subject to change by any member of this community. Finally, the document enters a stable, archival state where it may only be inspected but not modified; new versions, however, may be created [MUL86, Bert86].

The three filing stages have different storage requirements. The first and second stages require fast document access and updates and can be best served by magnetic disks. The archival stage, on the other hand, requires huge amounts of space, longevity and security, suggesting write-once optical disks as a very attractive solution.

The user interface is unquestionably a critical component of a multimedia filing system as it determines the level of user-system interaction and the community of prospective users. The objective of a good user interface design is to make the system accessible not only to a limited number of highly trained people but to everyone who may need its services, with minimum training.

There are several requirements that such a user interface should fulfill [Shne87, Vass82, Leea85]. First, it should support different kinds of dialogue techniques and levels of verbosity in order to accommodate both expert and novice users. In the office environment there are people who often perform the same sequence of operations. The user interface should be adjustable to the needs of such people by giving priority to frequent operations. Constant and incremental feedback as well as on-line help facilities are also essential for the better navigation of the user through the available operations. Last, but not least, the user interface should be consistent and uniform.

Query formulation is a very important and demanding function of the user interface [Gibb86]. Users tend to be vague about what they are looking for and often have difficulties in expressing it. Besides, multimedia documents can have very rich structure and content. Exploiting the document structure as well as the associations that exist between document components, and using flexible browsing techniques are two powerful tools for efficient, dynamic query formulation and resolution of user uncertainty.

In this paper we describe the design of a new user interface, in particular the query-related functions, developed at the Institute of Computer Science, Research Center of Crete, within the context of
ESPRIT (European Strategic Program for Research and Development in Information Technology) Project no. 28 (MULTOS). MULTOS stands for Multimedia Office Server and aims at developing a multimedia document filing system based on an open, distributed, client-server architecture, supporting storage and retrieval on the basis of the structure and content of documents, and using both magnetic and optical storage media [MUL86, Bert86]

In the next section we give a brief overview of multimedia document retrieval. In sections 3 and 4 we describe the design and the operations of the user interface for query formulation. In section 5 we outline the use of browsing. In section 6 we discuss some implementation issues and finally, we draw some conclusions and present directions for future work.

2. Multimedia document retrieval

Earlier work with multimedia documents has focused mainly on the creation and presentation of multimedia documents and on multimedia document exchange [Gibb87]. Systems addressing the issue of document retrieval include IBM's STAIRS, that handles only text documents, MULTOS [MUL86], MUSE [Gibb87, Cons86], and MINOS [Chri84, Chri86] that support multimedia documents. However, retrieval in these systems is mainly based on the attribute and text parts of the document. Work on retrieval of images has been done by [Chie80, Chan80, Tori80, Leee78, Leee77] but they do not address the problem in the context of an integrated multimedia document. The integration of retrieval techniques addressing all the components of a multimedia document is now attracting attention.

In the design described here our goal is, in addition to the general user interface requirements mentioned above, to provide for maximum flexibility in query formulation, to support the resolution of user uncertainty by allowing dynamic query reformulation and to exploit all the associations between the various document components. This is achieved by allowing the user to dynamically identify the query boundaries; making extensive use of browsing; and allowing bidirectional transitions between operating modes ("states"), i.e. query editing, browsing and query history examination.

From the retrieval standpoint, the set of all documents in the system, in any filing stage, may be divided into two partially overlapping sets: a search space and an auxiliary space, the content of which
is listed in Table 1. The search space is the set of documents where retrieval is to be made from, while the auxiliary space comprises documents which may be helpful to the user in formulating or altering a query.

<table>
<thead>
<tr>
<th>SEARCH SPACE</th>
<th>AUXILIARY SPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>• archival store (types, collections)</td>
<td>• query catalogue</td>
</tr>
<tr>
<td>• temporary collections (derived from query answer sets)</td>
<td>• temporary collections</td>
</tr>
<tr>
<td>• permanent private documents</td>
<td>• permanent private documents</td>
</tr>
<tr>
<td></td>
<td>• temporary documents</td>
</tr>
</tbody>
</table>

Table 1. Document spaces

Navigation through the document spaces is aided by using some organization scheme. In MUL-TOS, for instance, archived documents are classified according to a hierarchy of document types and also belong to collections of possibly heterogeneous documents (of different types) which are relevant to various affairs [Barb85, MUL86]. This organization is used during query formulation to restrict the search space. Also, a search may be restricted on some combination of the answer sets of previous queries (temporary collections).

Associations that may exist between document components can be used in formulating queries. Such associations are:

- **Image to text**: Image captions, text paragraphs referring to the image.
- **Voice to text**: Voice captions, text paragraphs referring to the voice recording.
- **Image to voice**: Recorded comment on the image.
- **Image to attribute**: Attributes describing semantic content or other features of the image or of the objects therein.
- **Voice to attribute**: Attributes describing content or other features of the voice recording.

Finally, queries may refer to one or more components of the document.
3. Design of the query interface

As mentioned above, our design goals for the query interface are to maximize flexibility, to support the resolution of user uncertainty and to exploit associations between document components so as to enhance the overall effectiveness of the retrieval process. The basic design choices made are:

- Use of Query By Example [Zloc75, ChFu80] which supports direct specification of restrictions as well as furnishing a "model", e.g. model image obtained through a scanner.

- Consistent use of dynamic pop-up menus combined with accelerators. Pop-up menus present all the possible choices in the current context and do not allow meaningless selections. Combined with accelerators [Sun85] that will facilitate expert users, they achieve both speed and accuracy.

- Display of the search and auxiliary spaces at the requested level of detail as well as of specific document components. The display of the search space (e.g. type, collection catalogues and private documents in MULTOS) informs the user about the document organization, while the display of the auxiliary space helps him to better define what he is looking for.

- Presentation of hierarchically organized information, such as classification of documents or individual document structures [MUL86], as a horizontal rather than vertical tree structure. The vertical tree representation of hierarchies has been established almost as a standard among the computer community. However, we believe that for the purposes of office work a horizontal representation is better.

- Provide fast browsing through sets of documents using miniatures as well as through individual documents using full documents. Browsing will help the user find something similar to what he is looking for, which to use as a template while formulating a new query. Also, browsing will be used for final document selection.

- Enable easy transitions as well as transfer of information among "query states" i.e. query editing, document browsing and query history examination. If, for instance, an interesting document is encountered while the user is browsing through local documents, this document may be inspected either in the current browsing state or after entering the query editing state where it will serve as
a guide for query formulation. Query execution may be initiated from any of the "query states".

- The query language is available to the user. This is necessary for two reasons. It acts as system feedback to whatever the user has edited in the query formulation; and it allows the expert user, who is familiar with the query language, to directly edit queries in textual form.

4. Operation of the query interface

In this section we describe the operation of the query interface assuming the document organization of MULTOS mentioned previously. In figure 1 the general layout of the query interface is presented. One part of the query interface is dedicated to display the search and auxiliary space that the user has already defined i.e. the types, collections, private documents and, previous queries chosen. The four components of a multimedia document (attributes, text, image, voice) are also displayed. We will call the components that are involved in a query active components while the ones that are not involved, passive components. By selecting any of the components, the user makes it active. The sequence in which components are activated is unimportant and reactivation of a component is allowed. In a specially reserved area the query is displayed in the syntax of the query language.

4.1. Query on attributes

In figure 2 the attribute specification interface is shown. In the attribute specification area, the user sees the structure of a type, or of a particular private document that he has specified. The components of the type or of the particular document can be edited using operators from a dynamic pop-up menu. The available operators depend on the type of the component i.e. operators for numeric components include EQUAL, LESS, GREATER, operators for string components include LIKE and operators for date components include BETWEEN [MUL86]. The user is also allowed to extend the structure of a document by adding nodes and links, possibly including unspecified intermediate nodes.

4.2. Query on text

Queries on textual content can require the presence or absence of a number of words, conjunctively or disjunctively. For editing such queries, a text specification window is opened in the query for-
mulation area where the required words are entered (Figure 3). A pop-up menu will display on request the available operators: AND, OR, NOT, END OF QUERY. The list of operators is extensible. For instance, a LIKE operator is needed if similarity retrieval is to be supported by the system.

Words entering the query specification may well have their origin in currently (or previously) inspected documents. Moreover, the decision about using a word in a query may be made just after having seen this word. So, we find useful to offer the possibility of opening a document presentation window during query formulation, from which words can be simply copied into the query specification area. Spelling errors, typing effort and having to remember words are thus minimized.

4.3. Query on image

In a multimedia system, images have both semantic and syntactic content and may have attributes associated with them. Images may belong to one of the image classes known to the system. Image classes are described by image class attributes and image class objects which also have attributes. Queries on images may refer to semantic content, syntactic content, image attributes or combinations of the above. The user can query images indirectly i.e. by using existing associations with text or voice components. Associated text includes image captions and in-text references to the image. Associated voice includes recorded comments on the image.

Queries on the image semantic content are formulated by specifying image semantic attributes, by giving a pictorial example (the user may either draw the image, or get the image from a scanner or from another document) and by specifying associated document components. Image syntactic content is specified by pictorial example. Image specification is accomplished through a uniform interface. The system provides an image specification canvas (Figure 4) where the user can specify the desired image in any of the above described ways.

4.4. Query on voice

Voice recordings, similarly to images, have both semantic and syntactic content. Voice captions and in-text references to a voice message are possible associations between text and voice in a multimedia document. Additionally a voice recording can be a comment to an image.
The technological advances in the area of voice recognition and synthesis, and the cost of such hardware unfortunately restrict queries on the audio part of documents, to attributes and associated document components (text, images).

5. Use of browsing

Inexact matching methods used for retrieval by content, such as signatures in the case of text [Fal85], usually produce, in addition to the desired answers, the so called "false drops". These are documents which really do not satisfy the query restrictions despite their being included in the query answer set. False drops can only be detected by document inspection and browsing provides a fast way of doing so.

Besides final document selection, browsing is useful in the query formulation stage. Inspecting the important features of documents accessed by a previous query or going through a local document, help the user decide what exactly to look for and formulate a new query. Thus, the combined use of browsing and dynamic query formulation enable the resolution of user uncertainty.

To provide effective browsing, both inter-document and intra-document browsing are supported.

5.1. Inter-document browsing

This is browsing through a set of documents and aims at giving the user the possibility to quickly inspect the important features of documents belonging to a query answer set or to a local dossier in order to decide which of these documents are interesting.

Inter-document browsing comprises two stages of feature presentation. In the first stage, the characteristic features of the document are presented. These include classification information such as document attributes, images and audio messages. For example, in MULTOS, characteristic features of a document are the values of its conceptual structure [MUL86]. This information has been inserted together with the document during its creation, either manually or automatically. Though possibly redundant, it is better to store this information separately from the document so that its retrieval will be fast and will not require the retrieval of the entire document. So, the purpose of this stage is to facilitate a first screening of the documents by the user.
During the second stage, the user sees more features of documents marked as "interesting" in the first stage. The extra features presented to the user in this stage are the ones on the basis of which a query was issued. For example, if a query requests documents that contain a specific image, the image is regarded as an important feature of the document. During this stage, documents are fully retrieved from the database.

In both stages, it is necessary for the user not only to see the presented information but also to be able to compare its occurrences in different documents. In order to fit more than one document on the screen of the workstation we present the information associated with each document in miniature form. Miniatures of text are created in real time by using a smaller point size for the letters. Miniatures of images are either created in real time or are created together with the document and stored in the database. Which of the two methods to use depends on the performance of the algorithm that creates the miniatures and the available storage capacity. Fast browsing through an audio recording can be achieved by playing a short part, e.g. three seconds, of the recording. This will be enough to give an idea about the speaker and the content of the recording.

5.2. Intra-document browsing

The final stage of browsing starts when the user decides that he wants to see a single document in its real form. During intra-document browsing, the user can move from one page to another (both forward and backward), use a scroll bar to define the approximate desired location, or define the exact page number. Additionally, the user can move to a page that contains a specific word or image or voice recording. Pages are presented in their real size unless they do not fit in the screen. In this case, scrolling inside the page is also provided. Pages may be folded if they exceed the size of the rest of the document and can be unfolded by the user. Finally, browsing inside a document can be based on its attribute values. The attribute values of the document are presented in a separate window next to the document presentation window. The user can point to an attribute value and see, in the presentation window, the page that contains it.
6. Implementation issues

Our implementation environment consists of six machines: a Vax-11/780, a Masscomp MC-500 workstation, and four Sun-2 workstations. The machines are connected by an Ethernet. A Microtek MS300-A scanner is also connected serially to one of the SUNs. The Vax and the Suns run UNIX 4.3 BSD and the Masscomp the Real Time Unix (RTU).

Two closely related multimedia document systems will serve as a starting point: MUSE [Gibb87] entirely developed at the Institute of Computer Science and MULTOS first prototype, developed in cooperation with the partners in the MULTOS project (1). Both are written in the C programming language and their user interface has been implemented on the SUNs using the facilities provided by the SunWindow operating environment. There were several problems regarding their user interfaces, so we intend to test the new user interface on both of them. This new implementation will follow the object oriented approach using the C++ programming language which does not create compatibility problems with the rest of the application code that is in C.

7. Conclusions, future plans

In this paper we have described a flexible query interface for multimedia document retrieval which supports a large spectrum of requirements as well as the resolution of user uncertainty. An implementation of this interface is under development at the Institute of Computer Science, Research Center of Crete. Our intention is to test the interface with documents from at least one application domain. One of our major concerns during the implementation is the maintainability of the interface.

The current design supports only queries on the document content. We would like to enhance the retrieval capabilities by allowing the user to specify characteristics of the documents that have to do with the layout form of the documents as well (e.g. layout structure of documents conforming with the ODA standard [ISO84]).

---

(1) Partners in ESPRIT Project 28 (MULTOS) are Olivetti (I), Battelle Institute (FRG), Consiglio Nazionale delle Ricerche (IEI) (I), EKIA (S), Mmemorica (GR), Research Center of Crete (GR) and Triumph Adler (FRG).
References

Barb85.

Bert86.

Chan80.

ChFu80.

Chie80.

Chri84.
S. Christodoulakis, J. Vandenbroek, J. Li, T. Li, S. Wan, Y. Wang, M. Papa, and E. Bertino, "Development of a multimedia information system for an office environment," Proc. of the Tenth Int. Conf. on Very Large Data Bases, Singapore, 1984.

Chri86.

Cons87.

MUL86.

Falo85.

Gibb87.

Gibb86.
IEEE85.


Leea85.

Leee78.

Leee77.

Shne87.


TorI80.

Tsic83.

Tsic85.

Vass82.

Zloc75.
<table>
<thead>
<tr>
<th>QUERY EDITING MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Types</td>
</tr>
<tr>
<td>Document Components</td>
</tr>
<tr>
<td>Collections</td>
</tr>
<tr>
<td>Local</td>
</tr>
<tr>
<td>Previous Queries</td>
</tr>
</tbody>
</table>

Query Language

Figure 1. Query Interface: General Layout
### Query Editing Mode

<table>
<thead>
<tr>
<th>Document Types</th>
<th>(root)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Components</td>
<td>(root_cc)</td>
</tr>
<tr>
<td>Collections</td>
<td>Archival</td>
</tr>
<tr>
<td>Local</td>
<td>Permanent</td>
</tr>
</tbody>
</table>

#### Previous Queries

<table>
<thead>
<tr>
<th>DOC.ATTR.</th>
<th>TEXT</th>
<th>VOICE</th>
<th>IMAGE</th>
</tr>
</thead>
</table>

#### Attribute Specification

- **QUIT**
- **DONE**

```
| document component specification |
```

---

**Query Language**

---

Figure 2. Attribute Specification Interface
Figure 3. Text Specification Interface
Figure 4. Image Specification Interface