# A User Interaction Language for Accessing Multimedia News Documents

Usha Vijayan

A Major Report

in

The Department

of

Computer Science

Presented in Partial Fulfilment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

August 1997

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

0-612-40236-3

# ABSTRACT

*A User Interaction Language for Accessing Multimedia News Documents*

Usha Vijayan

The conventional news media like News Paper, TV, Radio, and Internet provide news in bulk, of which only a certain limited topics may be of interest to an individual. Each of these media has some limitations in delivering the news in a personalized form. TV and Radio do not provide any control to the user in selecting a particular news, while the News Paper is not dynamic like TV or Radio. Internet has dynamism, good user control, and interactivity, but the available Internet browsers are not very user friendly in personalizing the news.

To overcome these limitations, an Electronic News Delivery System (ENDS), which delivers multimedia news documents that are personalized for each customer, is proposed here. The ENDS spreads over one Client and many Servers. The Client interacts with the user and collects the requirements while the Server stores the news documents. The system provides both *Conventional News Delivery* and *Personalized News Delivery* services.

To find a good language for communicating between User, Client, and Servers, a number of requirements were identified for the ENDS. Several existing query languages were studied to see whether any of them met these requirements, but none of them was found to be totaly suitable for personalized news delivery. Therefore, a new domain specific language called the User Interaction Language (UIL) was developed in this work. A Graphical User Interface for the ENDS was also designed and implemented to collect the user requirements and to convert them to appropriate UIL query.

# *Acknowledgements*

The author wishes to express her sincere gratitude to *Dr. T. Radhakrishnan*, Professor of Computer Science, Concordia University, for his supervision, encouragement, help, and the valuable discussions during the course of this project. His readiness and availability for discussions, whenever needed, are greatly appreciated. The author is also thankful to her supervisor for the financial support given during the initial stage of this project. His effort and time in reviewing this thesis and giving suggestions to improve it are also gratefully acknowledged.

The friendship and help provided by the fellow *Graduate Students* in the Department of Computer Science are also acknowledged. Association with them was a great opportunity to have experience in working in team which benefited me very much in my career.

Last but not least, the author expresses her sincere gratitude to her husband, *Surej*, for his love, support, understanding, and patience; to her little precious daughter, *Gayathri*, for all the loving moments and for the motivation to finish this report at the earliest; and to her *Parents* for their encouragement and support.

# *Table of Contents*

# *List of Figures*

# *List of Tables*

# *Nomenclature*

## Abbreviations:

| | |
|---|---|
| CBC | Canadian Broadcasting Corporation |
| CND | Conventional News Delivery |
| CQA | Cooperative Query Answering |
| CSQL | Cooperative Structured Query Language |
| DB | Database |
| DBMS | Database Management System |
| DB-QL | Database Query Language |
| ENDS | Electronic News Delivery system |
| FF | Fast Forward |
| GOMS | Goal, Operation, Method, and Selection |
| GUI | Graphical User Interface |
| GVSM | Generalized Vector Space Model |
| I/O | Input Output |
| IR | Information Retrieval |
| IRS | Information Retrieval System |
| OODB | Object Oriented Database |
| OOPC | Object Oriented Predicate Calculus |

| | |
|---|---|
| OOQL | Object Oriented Query Language |
| P | Precision |
| PND | Personalized News Delivery |
| PSQL | Pictorial Structured Query Language |
| QBE | Query-by-Example |
| QL | Query Language |
| R | Recall |
| REW | Rewind |
| SQL | Structured Query Language |
| SLS | Spoken Language System |
| TV | Television |
| UI | User Interface |
| UIL | User Interface Language |
| VCR | Video Cassette Recorder |
| VQL | Visual Query Language |

# 1

# *Introduction*

Conventionally, users get news from NewsPapers, TV, Radio, Internet, etc. Of the bulk news that is received through these media, only a certain limited topics may be of interest to one specific individual. Each news media has some advantages and some disadvantages in news delivery. TV and Radio do not provide any control to the user. User cannot change the linear ordering of  information and has to be physically near the TV or in the proximity to hear Radio. Even though newspapers provide this control to the user, they are not as dynamic like TV or Radio. The reader has to browse through the entire news paper so as not to miss any relevant points.

In some respect, the Internet is found to be the better medium for the news over these conventional media. The Internet can provide the dynamism of video as well as a good user control like in the newspaper. It also provides the interactivity of the personal computer [Hoff91]. But the currently available Internet browsers are not very user-friendly for the naive residential customers to personalize the news for individual users on her own.

The news access through Internet is a complex task, especially for most of the residential users who do not have much experience with computers and their jargons. Even though the information is ready, it is not easily retrieved. The main disadvantages of the existing web browsers are their general user interface and the standard query language which are not designed to incorporate the relevant features of the news domain. There is not much option given to the user except some searching, going to a known resource location, browsing through the available documents, etc. There is a lot of browsing required for selecting the personalized information. A user will not be interested in spending a lot of time and money to get information which is otherwise cheaper or free. Also these existing user interfaces require the user to specify the requirements clearly, which is a very difficult task in a complex domain like news.

Many of these problems can be overcome by developing an Electronic News Delivery System (ENDS), which delivers multimedia news documents that are personalized to satisfy each customer. With the design of a domain specific User Interaction Language (UIL) and a user interface, the personalized delivery can be made possible by the system.

## 1.1 Electronic News Delivery System

The Electronic News Delivery System (ENDS), we propose, spreads over one client site and many server sites. The client site is the personal computer of the user who uses the system. The server sites store news documents provided by different news producers. They can be located all around the world. Each of these documents is tagged with some attributes or meta-knowledge to give an overall picture of the corresponding documents. The system is being developed for mainly residential customers. They are normally naive users and the system assists the users in all phases of news delivery. The main purpose of this system is to deliver two types of services: Personalized News Delivery (PND) and Conventional News Delivery (CND). In electronic world, one of the essential difference between these two services would be in controlling the actual cost to the end user, that is no more expensive than buying a news paper daily.

**Personalized News Delivery (PND)**: The user specifies what she wants which are stored in a "User Profile". This avoids the time delay to browse through the entire news or to wait for a particular time. The user gets the required news whenever it is demanded. The system can also make use of an agent for each user, even within a single family. The personalized agent stores the user profile and helps the system in completing the missing parts of the requirements. The system delivers the personalized news to the user. While the agent is specifying the requirements of the expected news, the user can also interact with the system to alter or add any requirements. The agent can learn about the user needs and update the User Profile. Also the system provides complete control to the user while the news is being delivered. The user can change the way of presentation, or the presentation itself. For example, the user can access any of the VCR commands like FF, REW, PAUSE, etc. while viewing time dependent message. Also the media type can be changed from any one to another. The presentation of one document can be changed to another with different content. There are many more options provided to the user. These options are explained in the later chapters. Still in this system, the user has to spend time to specify the requirements. The news will be delivered periodically and regularly as in daily News Paper delivery. This would help plan one's daily routines.

**Conventional News Delivery (CND)**: Mainly the users demand the personalized news through the ENDS. Still, some users may want to stick with the routine of accessing the news through the conventional media. For example, some users may need to read a particular news paper in the morning. Some others may need to watch a prime time news of CBC at 10:00 PM. To satisfy those users, the system could deliver the news as a community news as it is delivered today. The content and the presentations will be the same as it is through that medium (TV, Radio, News Paper). In addition, the system provides all the useful control options as in the personalized delivery, which are not available with the conventional news media. This way, the system provides better service for all types of users. In this case, the user profile can be common to a family as opposed to one person in the family, thus resembling the News papers, Radio, or TV of today. In this case also, the system can make use of an agent. The personalized agent stores the user profile and helps the system in specifying the requirements. While the agent is assisting in completing the requirements of the expected news, the user can also interact with the system to alter or add any requirements. The agent can learn about the user needs and update the User Profile.

The system contains three major parts which are spatially separated: User, Client and Servers. Among these parts, client communicates with the user as well as servers. It is through this communication, system functions properly. Since the system is highly interactive, this communication language (UIL) plays a major role. This system is database oriented and so, we need a language which is similar to the query language for communicating with the system. We studied the existing query languages for this purpose, but none had all the requirements for effective user interactions in News Delivery. We propose a UIL as the communication language of the ENDS, that integrates many of the features found in database Query Languages (QLs).

## 1.2 Need for the Present Work

Most of the existing QLs were developed for alphanumeric (strings + numerals) data. The presentation of alphanumeric data is less complex than that of multimedia data which is inherent in News. The complexity of multimedia presentation was not taken care in the existing QLs. None of them provides control to the user during the presentation of the answer. For the news delivery application, the presentation could be of very long duration and the user can not be expected to be passive during this time. So the presentation control is a major requirement for the UIL.

None of the existing languages deals with the concurrent presentation of two items which may be very important for some of the users. Since some media types can be presented together, the system should be able to access them concurrently. It is a reasonable request from a user to view some photographs of an incident while she is listening to the audio news of that incident.

The existing languages do not provide any option to strengthen or weaken the specified conditions according to the number of matching documents found. When there is no matching documents found, the user may want to relax some of the specified conditions. Also, when there are a lot of matching documents, the user can select the most wanted by strengthening some of the conditions. This is a very desirable requirement for personalising the news delivery.

These are the major drawbacks of existing QLs in the news domain. Also, there are many more limitations found in them to adapt successfully into news application. So it was needed to develop a new domain specific communication language (UIL) for ENDS.

## 1.3 Objectives

The two main objectives of this project work are:

- To develop a domain specific UIL
- To design a User Interface (UI) for the Electronic News Delivery System

In the domain of seeking news or information, the user may not be able to give a clear specification of her requirements. She does not know what is available at the other end. Most of her requirements will be "vague". For the user to get full benefit, the system should provide the maximum information which is useful or interesting, even with the incomplete specification. This should not cost much more than the present means of news reception, nor the quantity of information should overwhelm the user. For example, a naive user will not be knowing about the location from where she can fetch a particular item with the minimum cost. If unspecified, the system should do that optimisation for her. The UIL help the system to carry all these vague or unspecified requirements to the server and process them.

User Interaction Language or UIL is developed for the communication in the electronic news delivery system. The user requirements are collected by the client site and sent to the server through a UIL query. UIL is a Domain Specific Query Language where the domain is multimedia news documents. It can be considered as an extended version of a traditional query language. There are some additional options present in UIL which are not present in the ordinary QLs.

The most important distinction of UIL from QL is that UIL is highly interactive and conversational. It provides a control over presentation as well as retrieval. The user can interact with the system even when the presentation is going on. There are lots of controls available in UIL: changing the presentation from one media to another, changing the reception from one mode to another, adding another media document concurrently with the ongoing presentation, adding another reception mode along with current mode, changing the document to another with different or same content in the midway of presentation, a variety of display controls which are meaningful to the current presentation (FF, REW, NextPage, Find, etc.), etc.

The UIL allows the user to strengthen some characteristics through its "Preference clause" and weaken some characteristics through the "Relaxation technique". By strengthening some of the specified conditions, the user can select the most wanted information when there are lots of documents of similar nature. In contrast, when there is no exactly matching documents, the user can get the nearest to his requirements by weakening some conditions. Giving different weights to different attributes is very important in the news domain where the user does not know what is available at the other end.

As the proof of concept demonstration, a menu driven user interface is developed for the system. This interface system collects the user input and converts them into a UIL Query.

## 1.4 Organization of this Report

An extensive review of the existing query languages was carried out among the old generation and new generation languages. The unique features found in them are provided in Chapter 2. The features that are useful in the news domain are also summarized in Chapter 2. Chapter 3 deals with the overall structure of the electronic news delivery system and the requirements of the UIL. The development of UIL is explained in Chapter 4. It provides the complete syntax and semantics of the UIL for each clause. The semantic is presented informally and no formal specification for semantics is given. The design of user interface for the system is given in Chapter 5. Finally the conclusions drawn from this work and the recommendations for future work are explained in Chapter 6.

# 2

# *Review of Query Languages*

A Query Language (QL) is defined as a high level computer language for the retrieval and modification of data held in databases or files [Jark85]. This is the only way of communication between the user and the database. Few years ago, query languages were evaluated on the basis of simplicity and expressive power. Most of the existing relational query languages like SQL. QUEL. etc. satisfy these two requirements [Jark85]. But the quest for new query languages still exists because of the limitations of the relational database and its QLs.

A relational database can only store the alpha numeric (strings and numerals) data. It cannot cope with the fast growth in multimedia. A multimedia database should be able to store both time dependent and time independent data [O'Doch91]. Object Oriented Database (OODB) systems are developed for this purpose. They can process and manage large amount of complex data like audio, video, statistical information, etc. For the real application, OODB like any other conventional database, requires a query language for a quick, set oriented retrieval of information about objects.

## 2.1 Types of Data Retrieval

The main function of the multimedia DBMS is to retrieve data for the user. Normally, users are not supposed to modify the database. O'Docherty *et al.* [O'doch91] classify the retrieval of the data into two categories: ***Presentation-Based*** and ***Content-Based.***

### 2.1.1 Presentation Based

Presentation based retrieval does not require any sophisticated analysis. It relates to the data type and data structure. It is commonly applied to composite media which contains more than one media. Two examples of such queries are "*Get me all the Voice data*" or "*Get me all the Text data*". This type of retrieval is very rarely used. It can be used only in small databases.

### 2.1.2 Content Based

Content based retrieval analyses the data stored in the database and retrieves data according to their semantic content. Content analysis range from very simple to extremely complex criteria. The system should have a reasonable response time in order to be interactive [O'doch91] which is harder to meet because of the probable complexity in the semantic analysis. Yoshitaka *et al.* [Yosh94] suggest two approaches to content-based retrieval of data. The first approach is to include some meta-information describing the contents or features along with each data-item. So, the system can retrieve data by referring to the attached information. This approach is very simple and processing could be faster. But it has got some disadvantages. The first disadvantage is the increase in storage space. The space for the meta-information is a waste. The second disadvantage is that any amount of meta-information cannot represent all features of the data. It is not practical to store all the features of the data in textual and/or numerical form. Further, when the standard of judgment changes, we must update all these meta-information. In the second approach, no information is attached to the data. When the query is processed, the system evaluates the entire data directly with respect to the query condition. This approach specifically implements content-based retrieval for a specific application. The disadvantages of this approach are complex analysis and application dependency.

In the content-based approach, there are two types of querying [Yosh94]: ***Direct Querying*** and ***Indirect Querying.*** They are explained below.

**Direct Querying**: In this type of querying, the data type in the conditional clause of the query will be the same as the type of data stored in the database. Normally multimedia queries are of this type. These queries can be processed by the system without the assistance of any other knowledge. It just has to compare the similar data type. In this type of querying, the heterogeneous types may have to be given as the condition of the query. The disadvantage of this type of querying is the difficulty of specifying it.

**Indirect Querying**: In this type, the query may not specify the same data type which is stored in the database. The condition in the query is specified only in a single media (say text) for retrieving the heterogeneous data type. This will simplify the query specification. The system needs assistance from the domain knowledge to interpret these queries. The domain knowledge prescribes a way of viewing the multimedia data in order to extract the semantics of the data. This also helps to translate an operator or a condition in the query into equivalent ones whose types are suited to internal query evaluation [Yosh94]. This method has several advantages over the direct querying. Even without adding the meta knowledge in the database, this method can specify the query in one media. The whole content of the data is used to interpret the queries. But the processing of indirect queries is more difficult than that of direct queries.

## 2.2 Different Ways of Querying

From the user's perspective, the querying can be done through a number of ways [O'Doch91].

### 2.2.1 Directly

This is the easiest way of querying and most of the QLs use this way. In this type, user is very much sure about what she wants and what the system stores and how it is stored. This is the oldest ways used in database querying. These queries can be very easily converted into the logical expressions or set operations.

### 2.2.2 Sketches

A "short hand" [O'Doch91] of an item is given on-line as the input. The system checks this input and compares with the stored data. If the given out line matches with any stored data, that will be selected. In these sketches, only the salient features are specified. An image sketch can be a rough outline with rectangular windows and multiple planes. A voice sketch can be a sentence spoken in a particular language or accent.

### 2.2.3 Similarity

The user specifies a particular item and the system checks in the data base for that item. Matching the finger prints in the crime laboratories is an example for this way of querying. Images can be compared using statistical measures [O'doch91]. Texts can be compared with the key words.

### 2.2.4 Confidence Level

If the queries are ambiguous or inexact, the system can relax the strict conditions and select the items which have the almost same meaning. These distance measures are referred to as confidence levels and could be discrete ("very good", "good", "not bad", etc.) or continuous (normalized to the range [0..1] for instance).

## 2.3 Previous Generation Query Languages

The query languages that are developed for conventional databases are called previous generation query languages [Jarke85]. In this section, a very short description of these type of languages is given. These languages can query the database only ***directly***. Other ways of querying explained in the previous section (sketches, similarity, confidence level) are not employed in these languages. The data model of the system should be very simple. Only alphanumeric data can be processed with these languages. There is no intelligent or cooperative processing available with these languages.

Figure 2.1 summarizes the trend analysis in developing these languages. The development of these QLs can be illustrated in two dimensions:

1. ***Functional Capability:*** What all things can be done with the system.
2. ***Usability:*** Effort of actually doing it.

According to these dimensions, previous generation QLs are classified into two groups: ***Programming Language Oriented*** which is better in functional capability and poorer in usability and ***Ergonomically Oriented*** which is better in usability and poorer in functional capability. They are explained in the following subsections.

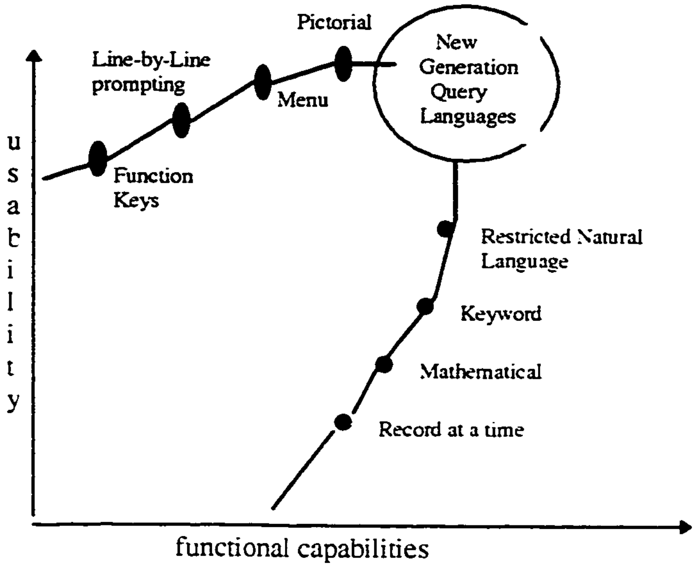


Figure 2.1 Trend Analysis in Query Language Generation [Jark85]

## 2.3.1 Programming Language Oriented

This group of QL originated from the disciplines of programming languages and database theory [Jark85]. They concern mainly with the syntactic structure and the semantic meaning of database interaction. The operations and specifications are explained by a command or a sequence of commands. This group of QL is not developed for casual users. The Programming Language Oriented queries have been classified into four classes [Jark85].

1. ***Record-at-a-time***: This is mainly used in the conventional file management and many early database systems for data retrieval. Currently some QLs use this for modification purposes.

2. ***Mathematical***: These languages are suited for high level user interfaces in set-oriented data retrieval in relational databases. This uses mathematical notation for short and succinct expression of powerful operation. Languages that use the position of the command operators and operands to convey the meaning are also included in this class.
3. ***Linear Keyword***: This group is the most popular group. Most of the QLs belong to this group. These QLs use English like statement to represent the query. The commands should be selected from a reserved list and they have a definite syntax and semantics.
4. ***Restricted Natural Languages***: This is the modification of the previous group, Linear Keyword. The user can use his native language for the interaction with the databases. Some of these QLs will interact with the user to resolve any ambiguity. Nevertheless, the natural language communication in all these QLs is still far from that used in person-to-person communication. This is the reason for using the prefix "***restricted***".

## 2.3.2 Ergonomically Oriented Languages

This group of languages originated from the ergonomic analysis of the interaction between naive users and computer systems. Development of these languages represents a trend for incorporating more functional capabilities, while their interface to users remaining novice oriented. Pictorial queries in the new generation of languages belongs to this group [Jark85]. This group of languages use the following user interaction styles [Jark85]:

1. ***Function Keys***: This is a very effective way of interaction between the system and the inexperienced user. By pressing a special key, a previously prepared transaction takes place. Even though this method is simple and effective, its usage is very limited. It can be used only for a very limited database operation, so cannot be used for large DBs.

2. ***Line-by-line-prompting***: This is a simple system driven dialogue. The user will be prompted to enter the requirement in each command. i.e., the name of the attribute, comparison value, etc. Then the system builds the query from these inputs and processes it. The disadvantage of this type is that, the user is supposed to know some of the inside details of the databases.
3. ***Menu Selection***: This is the modified way of Line-by-line-prompting. The user will be displayed all the choices and or required information as menus. The user has only to point her choices. Menus are structured hierarchically. Choice of one option may cause the representation of new menu. This has been the most sophisticated method in previous generation QL.
4. ***Graphical or Pictorial***: This would be considered as an early new generation query language that did not fully succeed because the necessary hardware and understanding of the user needs were not easily available. In this, user can manipulate visual symbols to formulate queries. The entities and relationships are represented in specific geometric shapes.

## 2.3.3 Disadvantages

Previous generation QLs provided only a restricted interactive environment. When they were developed, the user had a limited hardware interface (terminal and keyboard) and a relatively artificial and conceptual model of formatted data and their organization. The user also had a formal query language syntax, and used her own experience of the system to accomplish a task.

In these languages, the user's visual ability while interacting with the database is limited: the objects of interests are rarely displayed directly. Rather, they are represented by formatted text, thereby not giving the user any iconic clues (what the data look like) or spatial clues (where the data are) to help the querying process. Furthermore, the user does not fully employ her senses and cognition; for instance, query formulation does not use voice, touch, hearing, or gesture. Finally the interaction is "static". A previous generation query system shows little or no "intelligence" in deducing answers from incomplete, yet "obvious" representations of user interactions. The user may still perform a task, but with limited productivity and at the possible expense of more stress, less interest, and less pleasure.

## 2.4 New Generation Query Languages

New generation QLs combine the good things (*Functional capability* and *Usability*) of both groups of previous generation QLs: ***Programming Language Oriented*** and ***Ergonomically Oriented***. The developers try to integrate both these approaches into a functionally powerful query language meant for relatively unsophisticated users. These QLs can support complex data types and complex data models. This generation emerged from the advances in related areas [Jark85] such as hardware technology, developments in Graphics and Artificial intelligence, developments in Applied Psychology and Related Sciences, and success of Computer Games.

We divide the new generation QLs into several classes. They are Object Oriented QL, Pictorial QLs, Intelligent QLs, and Direct Manipulation QLs. They are explained in detail in the following subsections.

### 2.4.1 Object Oriented QL

In the introduction of this chapter, we have seen the importance of OODB in current applications. The important part of this new generation DB is its query language. There are a lot of OOQLs developed in past few years [Bert92], [Urba94], [Urba95], [Kife92], [Kim89]. The issues in designing a query model for object oriented databases are explained in [Bert92]. Among these OOQLs, many deal only with the retrieval aspects and does not consider updates. Since main operation in the multimedia database is retrieval, only that aspect is considered in this section.

#### 2.4.1.1 Main features

- ***Equality***: There are two types of equality in the O-O paradigm.
    - *Value equality*: Two objects are value equal if they have the same properties.
    - *Identity equality*: Two objects are identity equal if they are the same object.

- ***Predicates on Object Properties***: Since OODB support multi-valued properties as well as single valued properties, there should be two types of predicates.
    - *Membership predicate*: If an object or a single valued property of an object is a member of a multi valued property of another object, this predicates returns true.
    - *Inclusion predicate*: If a multi valued property of an object is included in that of another object, this predicate returns true.

- ***Navigation***: An important requirement of an OOQL is the capability of navigating through the object structure. This is similar to the notion of join in relational QL. Navigation is done with the use of a "***dot function***" in all the query languages of this type [Bert92], [Urba94], [Urba95], [Kife92], [Kim89]. "x.y.z" represent the object "z" which is a property of the object "y", which again is a property of an object "x". Thus, several dot functions can be sequenced to form a "Path Expression" [Bert92], [Kife92]. A path expression implicitly specifies a join between an object and a component object. Apart from this, there exists an "explicit join" similar to a relational join where two objects are explicitly compared by using either the value equality or the identity equality.

- ***Methods***: Another important feature of OODB is that it can support the methods with the object to express the local behavior of the object. In the query formulation, methods are considered similar to the properties. Therefore, in most of OOQLs, the methods are used just like attributes. But, unlike the attribute, with this method invocation, the specified method on a given object is executed. The methods can be included in the path expression also. A method can receive as input parameters not only constant values, but also values denoted by path expression [Bert92].

- ***Class Hierarchies***: The query can be applied on a class in two ways: only on that class or on that class and all its subclasses. Most of the query languages support both of these possibilities.

- ***Recursion***: QLs supporting advanced application include some form of recursion. The reason for introducing recursion in OOQL is that the schema in OODB may contain cycles. For example, the recursive operator of Object Oriented Predicate Calculus (OOPC) is represented by the operator P in [Bert92]. P is defined as follows. Given an object O, instance of a class C, and a property p of O, O.P(p) defines as a set of objects either are values of property p of object O, or are values of property p of some object O' which is a value of property p of object O, etc.

- ***Class of Operator***: This is not a very essential operator in many QLs. But it will be helpful for some application. The CLASS_OF [Bert92] operator supports the recognition of a class of a given object at runtime. This operator, applied to an object variable, returns the name of the class of the object.

### 2.4.1.2 Constructs of OOQL

Most of the proposed OOQLs have SQL like structure and constructs. Some of them have additional constructs to give better expressive power [Kife92]. The major constructs are explained below.

***Select Clause***: It is the target clause and specifies what must be retrieved by the query: (a) an object or (b) a property of an object or (c) output of a method. Urban *et al.* [Urba94] provide an interesting way of using "*" for not flattening the view of some multi-valued property at the output display. If a "*" is added with a multi-valued property at the select clause, those properties are collected into a set and displayed. Otherwise they will be flattened to provide a relational view.

***From Clause***: As in SQL, it is a range clause and it is optional. It defines the binding of the object variables. If the Class name is used in the other part. this clause can be avoided.

***Where Clause***: It is the qualification clause. This is used to specify a boolean combination of conditions that are to be met by the retrieval objects. All the predicates in the query languages (comparison predicate, equality predicate, membership predicate, inclusion predicate, quantification predicate) could appear in this clause.

***Object Creation Clause***: Instead of merely viewing the result of a query as an ordinary relation, we can also define the tuples produced by queries as a new object. For this, object ids are to be assigned to the produced tuples. In addition, since attribute names are crucial to the composition of a complex object, syntax should be extended to accommodate explicit assignment of values to attributes. Kifer *et al.* [Kife92] propose the OID FUNCTION OF clause for this purpose.

SELECT clause in the query gives the explicit names to attributes of the output relation. OID FUNCTION OF clause assigns an object id for each tuple in the result [Kife92].

***View Clause:*** Views are used for storing the objects temporarily. Most of the OOQLs do not provide the VIEW clause. It was first proposed by Kifer *et al.* [Kife92]. They define the view with the use of two clauses: CREATE VIEW and SIGNATURE. The CREATE VIEW clause is just to start the creation. The SIGNATURE clause provides the complete signature of the object class which is to be viewed.

## 2.4.2 Pictorial Query Language

Formulating queries in a formal query language, requires the user's knowledge of that query language and the underlying data. With the advent of OODB model, semantic content in data has increased [Moha93]. Accessing these systems involve a powerful query language and a large number of primitive query constructs. This complexity causes a heavier cognitive load on the user, since she has to visualize the whole DB schema to formulate a query. Because the knowledge in human long-term memory is imprecise, incomplete and often incorrect, user queries are subject to various types of failures [Cha91]: Spelling mistakes, Violation of language syntax and semantics, misconception of database schema, etc. All the text-based querying systems are found to experience this problem.

To handle this problem, Pictorial query languages have been proposed [Moha93], [Cha91], [Card93], [Berz93]. The main belief of this type of querying is that a pictorial form of communication increases the information content and significantly reduces the cognitive load on the database user [Moha93]. Most of the *domain specific query languages* proposed recently are pictorial. There are three types of pictorial query languages in the new generation QL: (i)Menu Guided [Cha91], [Card93], (ii) Tabular [Card93], and (iii) Visual [Moha93], [Berz93].

### 2.4.2.1 Menu Guided Querying

"Kaleidoscope" is the menu-guided query language which has been studied in one work [Cha91]. The menus reduce the granularity of user-system interaction and they create query incrementally with the user choices. They provide higher level user guidance (Intra-query guidance) to avoid various failures during query creation. To create a query, the user does not have to know the syntax of the underlying QL. She needs only to recognize or identify the constituents coming one after another that matches their intended query. A menu-guided querying system has two important parts:

1. ***Screen Organization***: This is the main part of the system. In Kaleidoscope [Cha91]. There are three objects on the screen: A base menu, a query status window and a system message window. The query status window presents the state of the partial query construction. The message window displays user requested and system driven messages during query creation. The base menu consists of varying number of choice windows, each of which lists choices. The content of these choice windows depends on the state of the partial query.

2. ***Guided Query Creation***: A query is created from left to right, guided by the system. The user adds a constituent to the partial query by selecting a base menu choice, and the system then updates the menu state. The system presents only choices that are syntactically and semantically valid for extending the partial query. This menu guided query languages can also provide a guidance in filling up the value for each attribute. To guide in this value creation, the

system can provide various helps: provide a pop-up menu for each value domain, provide the range of admissible values, provide the actual values itself, etc.

#### 2.4.2.2 Tabular Querying

The table driven queries are normally developed for applications in which the number of attributes of objects involved, is large. PICQUERRY+ [Card93] is one such type. The tabular querying system has two important parts:

- ***Screen Organization***: In tabular PICQUERRY+, five columns are used to specify the data to be retrieved.

  1. *Object*: This is the objects to be retrieved. This can be events or object attributes.
  2. *Relation Operator*: This indicates how the data should be correlated with the object value in order to satisfy the query condition.
  3. *Object Value*: This is the definition of what values for the specified objects constitute a truism with respect to the relation operator.
  4. *Logical Operator*: Logical operators AND, OR, NOT which allow construction of compound complex predicates.
  5. *Group:* Definition of parenthesized clauses in complex predicate.

- ***Query Creation***: In the tabular querying system, the query is created by filling up the required column(s) of the table given in the screen. There is no order for query creation, i.e., the columns can be filled in any order. During the filling-up of these columns the system checks each value that is filled in a column for the domain of that column. If there is a mismatch, the system provides an error message to the user. The tabular querying system can also provide a guidance for query creation by showing the possible values of each column to the user.

### 2.4.2.3 Visual Querying

A visual QL is graphically specified by using a set of visual primitives. There are formal syntactic and semantic structures underlying the creation of a graphical query. For retrieval purposes, the graphical primitives are mapped directly onto textual query primitives that are predicate expression.

VQL [Moha93], is a visual query language which has more advantages than other visual languages. VQL has a better visual content and expressive power. It provides a consistent mechanism by which queries can be specified pictorially to an object oriented database at multiple levels. In this, user can specify instance level queries, attribute level queries or recursive queries. The visual querying system has two major components:

- ***Query Primitives***: The VQL has three major query primitives as shown in Figure 2.2.
    1. ***Class Icon***: This is used for referring to any particular database class. The top portion of this primitive is for specifying the name of the dbClass while the shaded portion is used for referring to any particular instance of the class.
    2. ***Attribute Icon***: This is used for referring to any particular attribute. The attribute icon has a label portion (the oval shaped top part) for specifying the attribute name and a shaded rectangular bottom portion for specifying either (i) the value of the attribute, (ii) the range of values of the attribute, or (iii) a constraint on the value of the attribute, whose name is specified in the top portion.

3. ***Link Icon***: This refers to a special set of classes that correspond to the various database link objects (dbClassLinks and dbInstanceLinks). dbClassLink is the relationship that may exist between the dbClasses. dbInstanceLink links between specific dbInstances. In this primitive too, the bottom portion (shaded) is used for specifying individual instances of a particular link class, while the middle (un-shaded) portion is used for specifying a link class name. The shaded top portion of this icon is used for indicating the type of the link and for specifying details pertaining to the type of semantic operation that is to be performed.

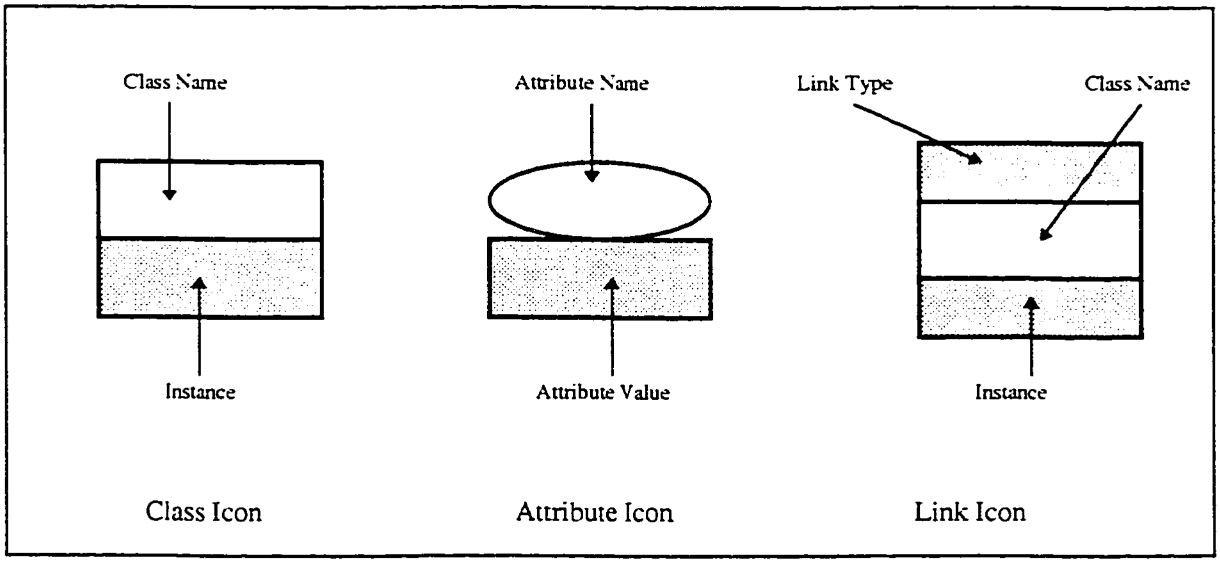


Figure 2.2 An Example of Visual Query Primitives [Moha 93]

- ***Query Creation***: In the visual querying system, query is created graphically using the visual query primitives. The primitives are filled with the required values and join together to make a query. The attributeIcon can have the values in its Value Field in different ways: as constant, as range, as unknown, etc. Two attributeIcons can be joined graphically in two ways: one for the AND connectives and the second for the OR connectives. The universal and existential quantifiers are denoted by special words in the Name Field of the Icons. The system can provide a guidance by providing the user with the possible values of the primitives.

An example query in VQL [Moha93]: Find the idNumbers of all the InletNeedles that have diameter greater than 0.25. (See Figure 2.3).

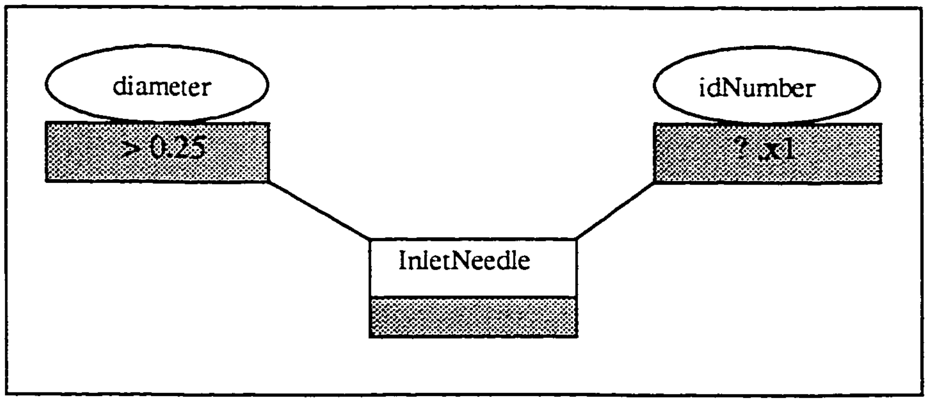


Figure 2.3 A Sample Query in VQL [Moha93]

### 2.4.3 Intelligent QL

Intelligent QL is another major class of new generation QL. Intelligence can be introduced in the query in many ways. Different ways of incorporating intelligence are first listed below and then detailed in the following subsections.

- Cooperative query answering [Chu94].
- Integration of DBMS with Information Retrieval [Saxt90].

- The ability to query the system without the proper knowledge of the schema [Laks93].
- Retrieving from incomplete information [Sutt95].
- Putting more knowledge in queries [Lacr87].
- Indirect Querying [Yosh94].

### 2.4.3.1 Cooperative Query Answering

Our daily life decisions are often made by asking and answering queries in an imprecise way. On the contrary, database querying requires the queries to be precise. User may not always specify queries precisely because of the limited knowledge of the structure and the contents of the data base. The conventional query processing does not give any information at all if the exact answer is not available. But in many cases, the user will appreciate a "close" answer instead of no answer. To remedy such restriction of precise query specification, the notion of Cooperative Query Answering (CQA) is introduced in QLs [Chu94].

Cooperative query answering provides neighborhood or generalized information relevant to the original query and within a certain semantic distance of the exact answer [Chu94]. A CQA process consists of enlarging the scope of the query by relaxing the search-range and refocusing the neighborhood range of the original query. Thus, CQA can provide an information of broader scope or information approximate to the "exact answer" when the latter is unavailable.

The query language CSQL [Chu94] is developed to support CQA. CSQL is an extension of SQL. It provides the following constructs for dealing with the query relaxation for CQA:

- ***Relaxation Symbol*** ^ : For marking the relaxable types, attributes and values on which an approximate query answer is tolerable.
- ***Relaxation Order***: Specifying the preference of attribute relaxation.
- ***Relaxable Predicates "Between" and "Within"***: "Between" is to specify a relaxation between a range such as Between(7, 11). "Within" specifies the relaxation among a group of items. Example: Within("Lax", "Burbank", "Long-Beach").
- ***Vague Predicates***: Which are relaxable based on given contexts, such as the following
  - close-to ("Rodondo-beach)
  - nearest-to ("Rodondo-beach)
  - approximate departure-time=9am
  - similar-to ("Chinese-food-style")
- ***Control Commands such as "Nearer" and "Farther"***: Which are used to control the relaxation scope interactively. These commands are also context sensitive.

CSQL also provides three types of CQA mechanisms.

1. ***Implicit Relaxation***: This is invoked automatically.
2. ***Explicit Relaxation***: This is invoked upon user's explicit specifications.
3. ***Interactive Relaxation***: This is interfered by the user interactively.

Unlike the traditional query processing, the CQA requires a domain knowledge to process the queries. This domain knowledge should contain the specifications of type abstraction hierarchy, as well as the abstraction domain values of atomic type which are required for the interpretation of queries.

The interpretation of CSQL is based on the knowledge of type abstraction hierarchy and is transparent to the users. Thus CSQL provides CQA not only by extending the functionality of SQL but also by modifying the query evaluation process to include logic inferencing, rewriting, and heuristic searching. With CSQL, a database system can provide approximate and conceptual query answering, tolerate imprecisely specified queries, and support object orientation.

### 2.4.3.2 Integrating Information Retrieval with DBMS

Information retrieval systems (IRS) provide well studied and well-understood models that lead to meaningfully ranked responses to queries. Saxton *et al.* [Saxt90] suggested a method to integrate IR with DBMS to make it more powerful database system. Saxton *et al.* argue that the abilities of IRSs in handling unprecisely described objects and relationships, handling imprecise user queries and ranking of responses from overwhelming amount of information according to their relevance will add an interesting dimension to the DBMS. In the Saxton's approach of integrated systems, IRS and DBMS are two special cases and the integrated system should have the capabilities to perform both DB type and IR type searches.

There are 4 types of queries available in this system.

1. Standard DBMS-type queries directed against traditional, formatted records.
2. Standard DBMS-type queries directed against document representations.
3. IR-type queries which, on the basis of descriptions available in the document representations and the conditions specified in the query, require the systems to provide estimates of the degree of usefulness of the corresponding documents to the user need.

4. IR-type queries which, on the basis of facts available in the traditional formatted records, require the systems to offer some judgment on the "relevance" of the corresponding entities to certain higher level user needs.

The type (4) queries will provide the "standard database" user with the ability to infer the relevance of the entity associated with a record to the concept implied by the query.

Saxton's integrated system has INGRES as the DBMS component and Generalized Vector Space Model (GVSM) as the IR model. To provide both kinds of retrieval facilities, there are some extensions added to the QL of the DBMS, which is QUEL. The keyword "RANK" in the place of "RETRIEVE" indicates the query is of IR-type. All other query constructs are same. In the IR-type queries, conditions are separated by parentheses. For IR-type searches to be meaningful, at least one condition term must be present. Furthermore, the IR-type searches will always connect the various conditions by the OR operation. However, within each condition we can have OR and AND connectives mixed together.

The user interface takes the query and determines its type DB or IR. If it is DB type it will be passed to INGRES. If it is IR type, each condition is made as a separate QUEL query and pass to INGRES. The set of results retrieved are ordered and presented to the user. The answer set is ordered according to the number of occurrences of an answer for these separate subqueries.

### 2.4.3.3 Querying the Database Schema

Lakshmanan *et al.* [Laks97], [Laks96], [Laks93] proposed a logic called SchemaLog which has a second order syntax and first order semantics. With this logic the queries can be developed with comparatively lower number of commands (rules and facts) than the other logical languages such as Prolog and Datalog. There are three major features of this language which make it intelligent: Schema Browsing, Schema Integration, Schema Evolution.

- ***Schema Browsing***: With the higher order syntax, SchemaLog can express queries ranging over the meta-information corresponding to the individual databases and their schema. So even without the sufficient knowledge of the underlying schema, the user can express complex queries in a natural way thus bringing programming closer to intuition. Also the queries can be written with unknown attribute name.

- ***Schema Integration*** : This is the process of unifying the representation of semantically similar information that is represented differently. SchemaLog provides this feature by creating a unified view of different schema in a heterogeneous database system, which can be conveniently queried by a user of any database in that federation. The use of logic rules in SchemaLog offers great flexibility in setting up such views.

- ***Schema Evolution*** : This is the process of assisting and maintaining the changes to the schematic information and content of database. SchemaLog provides evolution transparency to the users whereby they would be able to pose queries to the database, based on a possibly old version of the schema they are familiar with, even if the schema has evolved to a different state. The transformation program will take care of the relevant evolutionary relationship between the two schema, the querying schema and the current schema.

### 2.4.3.4 Retrieving from Incomplete Information

Some of the real world information is not complete to the database provider. Sometimes they are totally unknown or sometimes they are partially known. This information can be the answer for some queries. But, in the conventional database there is no way to model these data and query upon the model.

Sutton and King [Sutt95] suggested a model to treat unknown and partially known information in the functional model. An important aspect of their work is the abandonment of truth functional logic and the introduction of "certainty" and "possibility" operators into logical expression. To treat the unknown information, they introduced a value "***null***", whose semantics is "value at present unknown". Example: age Jane = *null*. Some times, we may have a partial knowledge about a data such that age of Jane is between 25 and 29. This knowledge can be represented by the function "Oneof". Example: age Jane = Oneof[25,26,27,28,29]. The introduction of the value "*null*" lead them to a three valued logic [Sutt95], consists of truth values True, False and *Maybe*. The truth value *Maybe* is the value given to the expression with *null* such as *null* = 10, which may be either true or false depending on the unknown value represented by the *null*.

For this logic, Sutton and King developed two types of operators [Sutt95]: Certainty and Possibility. Certainty Operator selects the data which are known with certainty. This is the operator similar to those used in the conventional databases. The Possibility operator is the specialty of this work, which select any possible data. Null values will be selected and Oneof[x1,x2,..] value may be selected, if any of the values inside the square brackets is satisfied with the given condition.

#### 2.4.3.5 Putting more Knowledge in Query Language

Normally, query languages provide a way to express mandatory qualifications on the data to be retrieved. They do not have facilities for expressing user preferences or desirable qualifications. But in some applications, the user is generally ready either to weaken the initially required characteristics if there is no objects satisfying them or to strengthen them if there are too many answers.

Lacroix *et al.* [Lacr87] developed an answer for this difficulty. They have added a preference clause in their query language. The qualification in the preference clause is similar to the qualification of a standard query. The preference is semantically defined as follows [Lacr87]: The query without the preference clause is evaluated. The preference clause is then applied to the answer set. It either turns the answer into an empty set in which case the preference clause is void i.e.. everything happens as if it were absent, or it hopefully reduces the cardinality of the answer.

### 2.4.4 Direct Manipulation QL

The last group in the new generation query languages is Direct Manipulation Query Languages. There are three basic features for these QLs [Jark85]: (a) object of interest visibility, (b) rapid reversible action, and (c) replacement of command language syntax by direct manipulation of objects. There are two types of direct manipulation languages were studied here. The first one is Query-by-Example, which is developed for the conventional alphanumeric databases [Ozso93]. The second is the Pictorial SQL for multimedia databases [Rous88], [Bach93]. These languages are briefly described below.

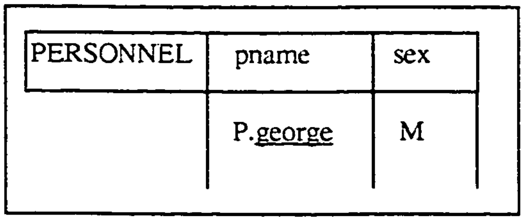
### 2.4.4.1 Query-by-Example (QBE)

This is a commercially available direct manipulation language that is based on the relational data model. QBE makes relations directly visible as objects (table schema) to be manipulated on the screen, and the user can move the cursor freely along the rows and columns of the tables. In this graphical language, the user creates the query by specifying an example output by directly making entries into relation skeletons instead of writing lengthy queries. There are many extensions to the QBE: Aggregate-by-Example, Summary-Table-by-Example, Query-by-Statistical-Relational-Table, Time-by-Example, Generalized-Query-by-Example, etc. [Ozso93].

To express a query involving more than a relation, more than one skeleton will be required. There will be an output window which will select the required output. The conditions on each relation are to be specified through examples in the appropriate relation skeleton. In the relation skeleton, a word that appears in capital letter is a constant: an underlined lowercase italic word denotes a free variable: and a doubly underlined lowercase italic word denotes a fixed variable, meaning that its value is the current value of another free variable with the same name in another window.

*Query Language Constructs*: As explained, a query in the example based language is a collection of object skeletons (such as relation skeletons), which are special purpose boxes representing constructs filled by the user with constants, variables, functions, and operators. The language constructor of the example based QL includes different types of boxes(Condition Box, Output Box, Range Box, Decision Box), aggregate functions, negation operator and set operator. The Condition Box is used to specify the conditions that can not be specified inside a relation skeleton. This box is used in all example based languages. The Output Box is a relation skeleton with one row filled with example elements. This box explicitly indicates the output of a query (or a subquery) and is used only in some of the example based languages. The Range Box is to define on the spot either a constant set (relation) or a union of relations over which a variable (or a set of variable ranges). This is also used only in some of the example based languages. The Decision Box functions as a condition statement (such as If-Then-Else or Case) of conventional programming languages. This is also not found in all example based languages. The aggregate functions (Max, Min, Avg, Cnt), negation operator and set operator are found in almost all example based languages.

A sample Query in QBE [Ozso93]: List the name of all male personnels. (See Figure 2.4).



| PERSONNEL | pname | sex |
|---|---|---|
|  | P.george | M |

Figure 2.4 A Sample Query with one Relation in QBE [Ozso93]

### 2.4.4.2 Pictorial SQL

Pictorial SQL (PSQL) is a direct manipulation language developed for pictorial database. In this QL, pictorial and alphanumeric databases are integrated to provide a uniform interface, but their representation and processing are clearly distinguished [Rous88]. In PSQL, user queries and searches on pictorial objects and spatial relationships among them must be directly specified in terms of the analog form using appropriate pointing devices. This allows the user to do direct manipulations on pictorial database. Alphanumeric data associated with pictures can be displayed on the picture to assist the user.

In PSQL, relations are defined over alphanumeric and/or pictorial domains. They model interdomain relationships. Every tuple models a relationship among those alphanumeric and pictorial elements of the domains. Each domain has a set of intradomain operators for creating, editing, modifying, deleting, selecting a particular element, etc. which directly manipulate the form and its internal representation. Intradomain operators are outside the relational model, and their interface is not a part of PSQL.

**PSQL Constructs:** PSQL has all the clause in SQL and one additional clause [< Picture-list >] which is optional. This specifies the picture on which the query is applied. The search area in the picture is specified in the qualification clause. It can be either a bound variable or a location given in absolute constant coordinates or variable coordinates. Furthermore, a search area in the qualification clause may be followed by a pictorial operator. There are many pictorial operators are available in the PSQL: *nearest, farthest, cover, not-cover, covered-by, not-covered-by, overlap, not-overlap, intersect, not-intersect, within, not-within, cross* and *not-cross*. The functions implemented in PSQL are *length, slope, area, perimeter* and *distance*.

An example Query in PSQL [Rous88]: Select all cities in the area {4+4,4-4,11+9,11-9} (eastern US entered by a mouse or using the window function) having a population greater than 450,000.

```
SELECT      city,state,population,location
FROM        cities
ON          us-map
WHERE       location within window(4+4,4-4,11+9,11-9) and
            population > 450,000.
```

## 2.5 Summary of the Review

A number of new generation query languages are reviewed in this chapter. All of them have some new features which make them suitable for some domain of application. Some of these new features can be useful for accessing the multimedia documents. The set of features which are useful to a news delivery system are explained as follows.

***Object Oriented Features***: The multimedia news documents are stored in Object Oriented databases. So the query language for news application should support the object oriented features.

***Pictorial Query creation***: A news delivery system will be used by a wide variety of residential customers. Not all of them will know much about computers. So it will not be easy for most of the customers to learn the syntax of the language. Some form of pictorial representation and/or menu guided query creation technique will help such users to use the system with ease.

***Co-operative query answering***: If the exact answer can not be found, this technique provides a neighborhood or generalized information relevant to the original query and within a certain "semantic distance" of the exact answer. This is a useful feature for a complex domain like news documents where the customer does not exactly know what is available at the server and which would be of interest to her. In this domain, it is very difficult to find an exact matching answer and the user may expect some cooperation from the system. The cooperation can be extended in three dimensions for the system if 3 types of abstractions were available at the document level. These abstractions are:

1. ***Spatial***: This is the relaxation on the conditions related to a place. The system can search among the nearby places.

2. ***Temporal***: This abstraction is based on time.
3. ***Semantic***: This abstraction is based on the semantic content of the document. Each content can be a subpart of a general item or subject. If the exact answer is not found another answer among this subject will be selected.

***Schema Browsing***: It is helpful in the news access if the queries could be applied on the meta-information. This is useful for the news delivery system, since the user may want to find the information regarding the meta-information itself. Also if the system allows the querying with unknown attribute, it will help the users who will not be knowing the attributes of the database.

***Adding Preference***: The addition of a preference clause can strengthen the preferred requirements. This will be helpful when the system finds lot of documents matching the requirements. Even if the system can not find any preferred documents, it can bring the other specified documents. Thus it does no harm even in the opposite situation.

Individually these features (suitable of news application) are researched by different people. But a language, integrating the suitable features that is intended for News Delivery application is not yet available.

# 3

# UIL: User Interaction Language for News Application

## 3.1 Electronic News Delivery System

The News domain is very large and complex. But only a fraction of that is relevant to an individual. Selecting that particular fraction of information from the vast collection is the main task of the news delivery system. Personalization of the news can be done only if the system can understand clearly, the user's preferences and constraints in all aspects. In the domain of news, a user may not be able to give a very clear specification of her needs. The user may not be knowing what is available at the storage sites. Yet, the user can specify the requirements in a somewhat vague and incomplete manner. A user should not be burdened by the details that she normally does not worry about. For example, a naive user will not be knowing about the location from where a particular news item can be fetched with the minimum cost. If the location is unspecified, the system should optimize the cost.

The front-end of an electronic news delivery system will acquire the user requirements through a suitable user interface and translate it to searchable queries. It will attempt to interpret the vaguely specified user requests and convey to the back end "database server". We assume that the system provides multimedia news documents and deals with two major services:

1. **Personalized News Delivery (PND):** This service is aimed at individual users and custom tuned for individual needs. The user specifies what she wants which are stored in a "User Profile". Mostly users demand content based news, i.e., news on a particular subject characterized by keywords or title. Other details like news-group, place, etc., also can be specified. Since most of the people select the news on the basis of the content, PND is a very popular type of news access. The user gets the required news whenever it is demanded. The system can also make use of a software agent for each user, or one for a single family. The personalized agent stores the user profile and helps the system in specifying the requirements. The system delivers the personalized news to the user. While the agent is specifying the requirements of the expected news, the user can also interact with the system to alter or add any requirements. The agent can learn about the user needs and update the User Profile. Also the system provides complete control to the user while the news is being delivered. The user can change the way of presentation, or the presentation itself. For example, the user can access any of the VCR commands like FF, REW, PAUSE, etc. while viewing time dependent message. Also the media type can be changed from any one to another. The presentation of one document can be changed to another with different content. There are many more options provided to the user. These options are explained in the later chapters. Still in this system, the user has to spend time to specify the requirements. The news will be delivered periodically and regularly as in daily News Paper delivery. This would help plan one's daily routines.

2. **Conventional News Delivery (CND):** Mainly the users demand the personalized news through the ENDS. Still, some users may want to stick with the routine of accessing the news through the conventional media. For example, some users may need to read a particular news paper in the morning. Some others may need to watch a prime time news of CBC at 10:00 PM. To satisfy those users, the ENDS could deliver the news as a "community news" as it is delivered today. The content and the presentations will be the same as it is through that medium (TV, Radio, News Paper). In addition, the system provides all the useful control options as in the personalized delivery, which are not available with the conventional news media. This way; the system provides better service for all types of users. In this case, the user profile could be common to a family as opposed to one person in the family, thus resembling the News papers, Radio, or TV of today. The personalized agent, in this case, stores the user profile and helps the system in specifying the requirements. While the agent is specifying the requirements of the expected news, the user can also interact with the system to alter or add any requirements. The agent could learn about the user needs and update the User Profile.

## 3.1.1 Different Parts of the System

The high level block diagram of the proposed system is shown in Figure 3.1. It has three major parts: ***Client site, Server site*** and ***User.***

### 3.1.1.1 Client

The Client of the news delivery system could be the Personal Computer of the customer. This is the front end of the ENDS. The user specifies the requirement and receives the news through this machine. The major components of the client site are explained briefly below.

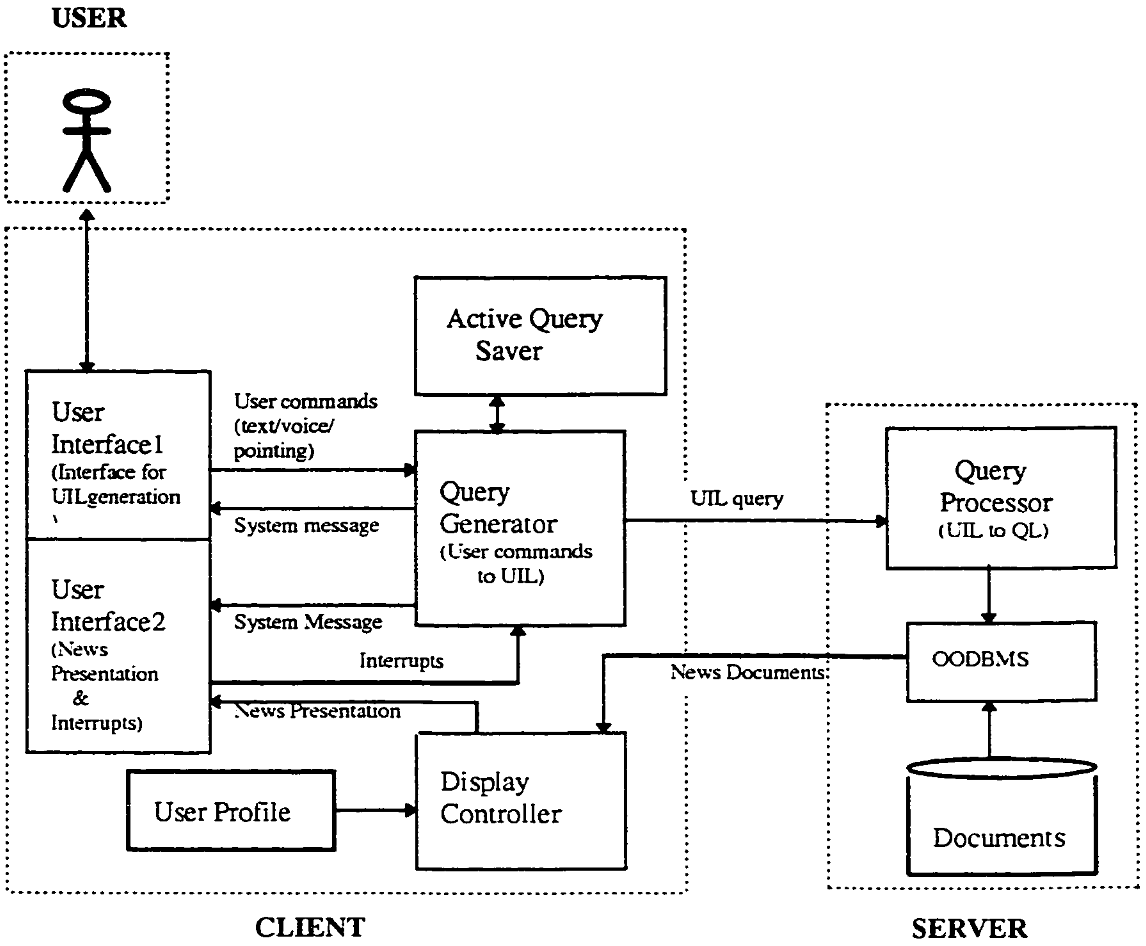

USER
Active Query Saver
User Interface1 (Interface for UILgeneration
User commands (text/voice/ pointing)
System message
Query Generator (User commands to UIL)
UIL query
Query Processor (UIL to QL)
User Interface2 (News Presentation & Interrupts)
System Message
Interrupts
News Documents
OODBMS
News Presentation
User Profile
Display Controller
Documents
CLIENT
SERVER

Figure 3.1 A High Level Block Diagram of the ENDS

**User Interface 1:** This part of the user interface collects the requirements for the required news from the user. This also provides some help to specify those requirements and constraints clearly.

**User Interface 2:** This part of the user interface displays the selected news document in the appropriate form. This also provides the user with the possible control of the media type of the news being presented.

**Query generator:** The Query Generator interacts with both parts of the user interface and analyzes the user inputs. These are converted into the UIL query here. If there is an error in the input, it sends corresponding message to the user through the user-interface. This message includes the type of error and the possible correction for that.

**Display Controller:** The server sends the documents to all clients in the same way. But users may need a personalized presentation. Display Controller controls the display according to the user profile.

**Active Query Saver:** The query which is generated in the last session could be saved for future use. After creating a query, the user may want to apply some controls on the presentation of its answer. To analyze these controls, the system should have the latest query. The Active Query Saver keeps the last generated query. This query can be used for future modifications also.

**User Profile:** This stores the user characteristics, user preferences, and user constraints.

### 3.1.1.2 Server

The Server is the back end of the ENDS. The news delivery system co-operates with many server sites. These are the resource locations where news documents are organized and stored in the news database. Apart from this news database, the server sites have two other major parts: Query Processor and the DBMS.

**Query Processor:** This translates the query from UIL to its local query language. For this translation, the query processor has to do some pre and post processing of the data along with its retrieval.

**DBMS:** The server stores the news documents along with some meta-information in the database. This database is managed by a database management system. This DBMS retrieves the data according to the query translated by the query processor. Such DBMS would be a multimedia oriented DBMS [Marc97].

### 3.1.1.3 User

The system caters to all types of residential users from school children to scholars, from laymen to intellectuals, etc. These users may fall in any of the categories between naive and expert in computer use. So, the system should provide interface which can support all these range of expertise.

### 3.1.1.4 Communication

There are three major parts in the ENDS which are spatially separated from each other and require to communicate often: User, Client and Server. They communicate through the system language. The communication takes place in two forms between these three parts.

1. *Between User and Client machine*: The user specify the requirements to the client. The Client delivers the news to the user. It also displays the error message whenever required.
2. *Between Client and Server machines*: Client sends the user request to server. The server sends back the matching news documents to the client.

The communication language of the ENDS should be able to do both of these communication, i.e., it should communicate between user, client, and server. The system is highly interactive and the communication language plays a major role in its success. A detailed study of most of the existing query languages has been done in Chapter 2 to find a suitable communication language for the system. Some of those languages had few features which are useful for the news domain, but none of them were found to be suitable for the news application. The limitations found on those languages are explained in the next section.

## 3.2 Limitations of Existing Query Languages

### 3.2.1 Standard Query Language

Currently most of the database related applications use query languages which are very similar to conventional SQL, Quel, or OSQL. These generic database languages do not contain any features to adapt to a specific domain. They require all the conditions to be specified clearly for retrieving the answer. Only with the clearly specified "search conditions", these queries can be processed. These conventional query languages work well with the old generation data types, which is alpha numeric. But they are not enough to fetch the personalized news from the servers in the Electronic News Delivery System, since the users may not be able to give clear search conditions.

### 3.2.2 Lack of Presentation Control

Database query languages provide control to the user only for selecting the answer. Displaying them is done by the I/O system in its own way. These existing languages do not provide any control to the user between a query dialogue, other than exiting the system. (A query dialogue starts with the submission of query and ends with the display of the last bit in the answer set.) User has to be passive after the submission of the query till the presentation is completed. Most of these QLs work fine with alphanumeric data since they are very short and can be presented in batches at the same time. In this case, the waste of time and money for browsing the data can be considered as negligible. But multimedia presentation is different from alphanumeric presentation. The multimedia presentation can be long and the data is presented only in a linear fashion. So, news services require interactive presentation. There should be some option provided in the language for the user to specify in what order the documents should be displayed or how to interrupt the presentation in a useful manner. Also there should be some control provided in the language for the user to change the presentation at any point of time to his choice. These options are not present in any of the existing database query languages.

### 3.2.3 Uniform Weight on Requirements

There are two worst conditions possible in the news application. In the first case, the number of matching documents can be large. In most of these cases, the user will not be interested in all of these documents. Bringing all these selected documents to the user will unnecessarily cost a lot of time to him. Also the user has to browse among all these documents. If the user is able to provide an option to strengthen some of the requirements in the language such that only those preferred documents will be displayed, this problem can be overcome. In the second case, the system may fail to find a perfectly matched item with the user's request. It is not a pleasing design to inform the user of this unsuccessful effort. The user might be ready to weaken some of the requirements for an answer rather than a failure message. These situations can be very frequent in the news application. To cope with these situations, the system should have a language which provides the user an option to weaken some of the specified requirements to get a neighborhood answer, in the case where system can not find a perfect answer. The conventional languages do not provide any option to strengthen or weaken the specified requirements.

### 3.2.4 No Concurrency

The existing languages do not present two documents concurrently. The user may want to get one audio document and some pictures of another document at the same time. Like Audio+Picture, there are many other possible combinations of the multimedia data type. Any of these combinations should be made available to the user. So the System Language should have a semantics for this type of concurrent presentation.

## 3.3 UIL: User Interaction Language

To overcome these limitations, a new communication language, called the User Interaction Language (UIL) is proposed in this work. This language can be used for communication between the client and server and also with the user. It is designed to meet the unique features of the news domain which are discussed later in this chapter. The system based on UIL has a user friendly interface for providing the personalized news. UIL is a Domain Specific Language where the domain is multimedia News Documents. In the ENDS, the multimedia documents are organized along with some textual meta-information, which are amenable for machine searching. UIL queries are applied on this meta knowledge to select the corresponding news documents.

UIL can be used to express all the requirements of a user which are unique to the news domain. The UIL query can be sent to each server site where it is converted into the local query language of the DBMS. To convert the special features of UIL into the existing DB-QL, some pre processing and post processing need to be done along with the data retrieval. The novelty of this work is this user interaction language, which can express all the complex type of querying for accessing the multimedia information. Even though, The UIL is converted to an existing query language at the server site, it can not be replaced by that server QL. It has some special syntax and semantics which are not available in any existing database query languages but are relevant for information access in the news domain.

### 3.3.1 Unique Features of UIL

UIL can be considered as an extended version of QL. There are many features that make UIL different from QL. These new features let UIL to be more useful in the news domain. They are outlined in the following subsections.

#### 3.3.1.1 Presentation Controls

UIL is a highly interactive language and provides control to the user throughout the use of the system. Unlike query languages, UIL allows the user interaction at any point of time even in between a query dialogue. Since presentation of certain documents can take a long time, providing the controls to the user during the query dialogue is a very important feature of UIL. The user can change the mode of presentation or the content of the presentation itself. All possible display controls (FF, REW, GOTO, etc.) are accessible to the users. The user can switch from one media to another or one reception mode to another. She can also change the presentation of one document to another, if the current one is not interesting. On the other hand, additional services or details can be demanded without changing the topic of the presentation if the document is interesting. All of these presentation controls are explained in detail in Sections 3.5 and 4.3.

#### 3.3.1.2 Concurrent Processing Capability

UIL applies concurrent processing in 3 ways: Concurrent Presentation, Concurrent Searching, and Concurrent Reception Modes.

***Concurrent Presentation***: UIL can present two media documents concurrently. For multimedia documents, it is possible and sometimes necessary to present two media types concurrently. When a user listens to an audio news, she may want to watch some photographs regarding that document. Also while reading a text, viewing the images (like street maps) will help to create a better feeling about that content.

***Concurrent Searching***: News documents are organized under different subjects depending upon their subjects. Some documents can be part of more than one subject. Mainly, users select the news on the basis of the subject(s). In some cases, users are interested in the content which is a combination of two or more subjects. So, they specify all of these subjects for concurrent processing. UIL provide a simple way to specify this concurrent searching.

***Concurrent Reception Mode***: UIL provides three modes of reception: On-line presentation, Down Loading, and Hard Copying. These modes can be accessed concurrently as well as individually.

### 3.3.1.3 Different Weights on Conditions

In the domain of information retrieval, we can define two terms, *Recall* (R) and *Precision* (P) as follows. *Recall* is defined as the ratio of the total number of items retrieved to the total number of matching items in the DB. *Precision* is defined as the ratio of the total number of items in the retrieved set to the total number of items relevant in the retrieved set. In ideal situation, R=1 and P=1. *Recall* is related to the retrieval of the data, while *Precision* is related to the user judgment of the data. In the database query languages, the qualification conditions carry equal weight. So the relevancy in DB-QLs is defined as pattern matching the specified search conditions. Those query processing assumes *Recall* and *Precision* to be 100%. They retrieve all the matching documents found in the DB and present all of them to the user. This will be a problem when the total number of matching items is too large or zero.

In the news application, the number of matching documents can be either too large or too small at the server site. When there is a large number of matching documents present in the database, the user may want to have a higher *Precision*. i.e., she wants to select only the most relevant items from that retrieved set. In contrary, when there is very few or no exactly matching documents found, the user may want to have a higher *Recall* by selecting the documents which are in the neighborhood of the specified conditions. The R and P can be changed by assigning different strength to the qualification conditions. In that case the relevancy of the document is judged by the user, not by the system.

UIL allows the user to assign different strength to different conditions. The users can strengthen condition(s) by specifying them as *preferred.* By strengthening a condition, the *Precision* is increased and the system can bring the most relevant documents to the user when it finds lots of documents matching the requirements. Also the users can weaken the condition(s) by specifying them as *relaxable.* By weakening a condition, the user gets an increased *Recall* when there are no exact match. The user can get an increased *Recall* and increased *Precision* by specifying both preferred conditions and relaxed conditions in the query. In the news application, the strengthening and weakening of conditions are very important, since the user does not always know exactly what is available at the server sites. Also this helps the user to determine the relevancy of the document found in the server when it is matching the initial specification which may be vague and incorrect. It is equivalent to a reconfirmation to fetch the best out of the set.

### 3.3.2 Phases of UIL

UIL generation has two phases.

1. ***Initial Query Creation or Selection Phase:*** In this phase, user specifies the attributes of the expected documents for the first time after invoking the system. The system arranges these requirements into a UIL query and sends to the server. The system also keeps it as the "active query" in the local client site. This active query can be processed for the next phase (presentation phase) or modified for the subsequent retrieval.
2. ***Interrupts Processing during Presentation or Presentation Phase:*** In this second phase, the presentation is controlled by the user commands. These commands are active while the

document is being presented and the user can invoke any of them at any time. This phase is not found in any of the other languages.

In the following two sections (3.4 and 3.5), the requirements for the User Interaction Language are discussed. These requirements can be divided into two: for ***Selection Phase*** and for ***Presentation Phase***

## 3.4 Requirements for Selection Phase

In the selection phase, the user initially specifies all the required conditions to be found in the retrieved documents. Selection phase exists in all query language. But in UIL, selection phase has to be extended to meet the needs of the news domain. The selection phase is divided into 4 subphases: ***Output Form, Range Control, Qualification Conditions*** and ***Presentation Details***

### 3.4.1 Output Form

In news application, there are four types of documents available at the server site: Video, Audio, Text and Image. The user should be able to select any one or any possible combinations of these types. Also the user should be able to select ant details or meta-information regarding any news documents. There are three modes, through which the user may want to collect the news. They are explained below.

1. ***On-Line Delivery:*** User may want to get the news on-line through the client I/O devices, (screen, speakers, etc.). She has to specify the type(s) of documents to be presented on-line which can be any of the above mentioned data type. Also the user may sometimes want to get some combination of these media type concurrently. For example, a user may be interested to

watch some pictures when she listens to the audio news. Like Audio+image, there are many other combinations: Text+Image, Audio+Text, Audio+Text+Image. In some cases, user may want to get one type after another. i.e., to listen to the audio news and then to read the textual news, then to watch the video part. In these cases, she may be able to specify a series of type(s) which should be presented one after the other.

2. ***Downloading the Documents*** : Through this mode, user may want to download the documents to her computer for the future use. She has to specify the type(s) of document to be downloaded along with this option. If technology permits, she should be able to specify all types of document, Video, Audio, Text, or Image. If the user needs the document for future use, then she can specify the expected time, when the document should be ready. The system can use this time interval to check when the traffic cost is the least and brings the news to the client.
3. ***Hardcopying the Documents*** : User should be able to hardcopy the document for mobile use. But only Text and Images can be asked for hardcopying. If the user needs the document for future use, then she can specify the expected time, when the document should be ready. The system can use this time interval to check when the traffic cost is the least and brings the news to the client.

In all these modes, the user should get ***advertisement free*** news delivery, if required.

### 3.4.2 Range Control

The news delivery system co-operates with many server sites. At each site, the documents are arranged according to the subject. The documents are classified into different section and they are stored under that particular subject. Searching through all these available sections in all the participating server sites will be a very time and cost consuming job. Also the documents of the same content will be similar at different sites. To reduce cost, time and some redundant documents, some users may prefer the search to be controlled to restricted sites(s) and/or subject(s). These search controls are explained as follows.

1. **Site Based Control** : The cost of retrieval of a document could be different in different server sites. Also the time for transmission can also be different depends on distance between the client and server machines. This difference makes the user to prefer some server sites over the others. Also she may want to select more number of sites at a time allowing the system to search all these sites.
2. **Subject Based Control :** Most of the Users may not be interested in many of the available subjects. So they should be given an option to specify their interested subject. With this clause, user can provide the required subject to be searched. Sometimes, users may want to get a document only if it is related to two or more subject. For example, the Olympic games can be found under the subjects SPORTS and WORLD. This type of concurrent processing can be specified in this clause. Also, users want to specify more than one subject for the system to first find the articles from the first subject, then from the second and so on.

### 3.4.3 Qualification Conditions

The documents should have met some conditions before their retrieval. These conditions should be given by the user. They are boolean expressions which can be connected by boolean operators AND and/or OR. Each expression is of comparison type, where a meta information is matched with the given value. The Qualification expression consists of three components. They are explained below.

- **Attribute :** Attributes are the meta information associated with each document. The conditions are specified on these attributes. The value of the specified attribute is to be compared with the given value. Only if this expression is true, the document is retrieved. The attribute can also be derived from a method associated with the meta information. Also some aggregate functions (Avg, Max, Min, Cnt) may have to be considered. In some special cases, the user may not be knowing exactly the attribute name. But she wants to compare this unknown attribute value with some known value. For example, if the user knows about an incident and does not know that appears as the title or as a keyword of the expected document. She will use the Attribute-less matching with the unknown attribute type.

**Equality Type:** The comparison type is very important in an expression. We need many types of comparison. Some of the ordinary comparison operators are =, #, <, >, <=, and >=. There are 5 types of matching in UIL. They are explained below:

1. **Strict Matching:** The value of the attribute is strictly matched with the given value.
2. **Approximate Matching**: The value of the attribute is approximately matched with the given value. This matching is used while comparing the numerical values. In this matching, an attribute is matched with a pre-defined range of values around the specified value.
3. **Relaxed Matching**: If the specified document is not found, the system can relax search according to the relaxed conditions. This type of co-operation between user and the system can bring the neighbouring answers if the correct answer is not found.
4. **Partial Matching:** When the user knows only partial details of the item, or he does not want to specify a long word, then this type of equality can be made use of. It is equivalent to the wild character matching.
5. **Attribute-less Matching**: If the user knows only a particular value and does not know under which attribute it appears, attribute-less equality can be made use of. In this type, the attribute is not specified. The system compares the specified value(s) with the values of all possible attributes and tries to find the answers for the users query.

- **Value:** In the boolean expression, the existing data has to be compared with a fixed value. This fixed value should be given by the user. The user can either give the complete value or the partial value with the wild character.

## 3.4.4 Presentation Details

Presentation of alpha numeral data is comparatively simple because, they consume less amount of time, space, and money. But presentation for UIL is a complex procedure, since it deals with multimedia data. Each multimedia document is large and they can be presented only in a sequential fashion. If the presentation is not interesting to the user, the system will fail to attract the customers. So there should be some option for the user to specify certain details that could be used for effective presentation.

### 3.4.4.1 Preference

In some cases, the system may find a large number of documents which meet the specified characteristics. In those cases, the user may be communicated with the largeness of the volume of retrieval. From this knowledge, the user may be ready to strengthen some of the selected documents through some new query constructs. User can express these preferences using some boolean conditions just like in the qualification clause. These conditions are added to the query in two situations: (i)when the system informs about the large number of matching documents and (ii) during the generation of the initial query. These conditions are processed only after the retrieval of the answer set. If there are some documents satisfying the ***preferred*** conditions they are to be selected as the answer to the query for the user from the retrieval set. If there are none satisfying the conditions, this clause can act as void and all the retrieved answer set can be brought to the user. The preference condition is specified similar to the Qualification condition. The components of this clause are ***Attribute, Equality Type***, and ***Value***. They are explained in the section of Qualification Conditions (Section 3.4.3) above.

### 3.4.4.2 Ordering

UIL should give the user some control on the order in which each document should be displayed. In SQL, ordering option gives a sorted order of data on presentation. Since we are not concerned with the alpha numerals, sorting is not the solution for ordering the display. In UIL, ordering is specified as boolean conditions similar to the Qualification condition. The components of this clause are ***Attribute, Equality Type***, and ***Value***. They are explained in the section of Qualification Conditions (Section 3.4.3) above. The documents which match the first ordering condition will be presented first, then the second and so on.

## 3.5 Requirement for Presentation Phase

Existing query languages provide user with the control only for selecting the required data items. But after that, the user does not have any control over presentation. The multimedia documents are comparatively very long and it is not a good design if the user is passive during the presentation. User should be given control over the presentation, so that the user may interact with the system at any time during the presentation. Unlike conventional query languages, the UIL provides an option for this type of interactive communication. The presentation controls are divided into three categories as follows and are explained in the following subsections.

1. ***Display Controls***
2. ***Change of Current Presentation***
3. ***Additional Services***

### 3.5.1 Display Controls

The controls which are used to alter just the presentation are called display controls. It does not change the document being presented or the mode of news delivery. It only manipulates the presentation to reach to another point in the document. These controls are different for each media type. Along with the news delivery all the corresponding controls are available for the user. These controls are commonly found in home Video and Audio cassette recorder.

**FastForward** : Fast-forward a video/audio presentation.

**REWind** : Rewind the video/audio document for replaying.

**Pause** : Stop (halt) the video/audio presentation for a short while.

**Play/Continue** : Start a new video/audio presentation or resume a halted presentation.

**Stop** : Stop the news presentation.

**StepFW** : Forward the video presentation frame by frame.

**StepREW** : Rewind the video presentation frame by frame.

**NextPage** : Flip the textual document to the next page.

**PrevPage** : Flip the textual document to the previous page.

**Start** : Start the presentation from the beginning again.

**Finish** : Go to the end of the presentation.

**GoToPage** : Flip the textual document to a specified page.

**GOTO(section):** News Papers are arranged in different sections, similar to the printed news papers. The news in each section is printed together. i.e., in the same page or in the adjacent pages. They are never scattered. Mostly people read news or at least make a first glance through the news according to the section of their interests. They read some sections carefully, and skip some others completely. So, going straight to a chosen section can be very useful for the users who opt for electronic news paper. This can replace a lot of page flipping.

**FIND(string)** : In some cases, the reader just wants to get some information like Loto649 result, What is on TV?, etc. She can just get them using a FIND option. It is a very essential requirement since searching a small portion in the computer monitor is much harder than searching it in a news paper.

## 3.5.2 Change of Current Presentation

Through these type of commands, the document being delivered can be changed. If the user is not interested in the content of the present document or the media type of the document, she may want to change the presentation to another document. The controls under this category allow the user to change the document to her choice without starting the selection from the beginning. The required controls are explained below.

**VideoVersion** : Switch to the video version from audio/text versions.

**AudioVersion** : Switch to the audio version for video/text versions.

**TextVersion** : Switch to the textual document from audio/video version.

**Skip** : Skip the current document, if not interested in that particular document. But still want to get the other selected documents of the same content.

**HeadLines** : Some readers are more interested in headlines. When there is a shortage of time, they skip the detailed news and listen only to the headlines.

**NO-AD** : Some users prefer news without interruptions from advertisement. If NO-AD option can deliver advertisement free news delivery, many readers can appreciate it, even for a higher cost.

**GetNewsOn(day):**After reading/listening to a news, the user may want to refer to another day's news. It is a frequently used scenario, since many people can not access news regularly. It is not advisable to ask them to start selection from the beginning, when only the date is to be changed in the original selection conditions. This can be applied for TV/Radio access also.

**GetNewsAt(time)**: After receiving the TV/Radio news, the user may want to receive another segment which was broadcasted at the specified time.

**Exit** : Exit the system at any point of time during the presentation.

### 3.5.3 Additional Services

In some cases the user may be so much interested in a particular document, she may want to have some additional services regarding that document. These controls are provided to seek such additional services without interrupting the current presentation.

**AddText** : Along with the audio news, the user may want to get to the textual part of it on a screen.

**AddPicture** : Along with the audio news, the user may want to see the photographs related to that incident.

**AddAudio** : Along with the textual news, the user may want to listen to the audio part of it.

**MoreOn**(subject/keyword): If the news is interesting, the user may like to have more details on some part or altogether. By default, the system assumes the previously specified subject and/or keyword. This option is similar to selecting the hyper-linked documents.

**Save** : User may want to download the document for the future use.

**Print** : User may want to hardcopy the document for the mobile use.

# 4

# *Development of UIL*

UIL is the user interaction language of the electronic news delivery system. Communication in UIL takes place between the user and her Personal Computer (Client site) and between the client site and remote resource locations (server sites). UIL, being a domain specific query language, includes many new features that are unique to the news domain. With these extended semantics. UIL helps the user to receive the personalized news document. This chapter deals with the development of UIL.

## 4.1 Meta Information for UIL

In the news delivery system, different news producers provide the news documents and they are stored in different sites at different locations. The documents are stored along with some meta information which is assumed to be in textual form. This meta knowledge provides the overall content description of the document. This meta information should be so constructed that it provides maximum knowledge about the related document. We are not concerned with the development of this meta information. But, we are developing the query language that can query this meta knowledge. The required meta knowledge for personalized news access and conventional news accesses are given in the Tables 4.1 and 4.2.

Table 4.1 Meta Information Required for Personalized News Delivery

| TV | RADIO | NEWS PAPER |
|---|---|---|
| KEYWORD(S) | KEYWORD(S) | KEYWORD(S) |
| - | - | TITLE |
| PLACE | PLACE | PLACE |
| TIME | TIME | TIME |
| NEWSGROUP | NEWSGROUP | NEWSGROUP |
| REP-DATE | REP-DATE | REP-DATE |
| REP-TIME | REP-TIME | REP-TIME |
| LANGUAGE | LANGUAGE | LANGUAGE |
| TELECAST-STATION | RADIO-STATION | PUBLISHED FROM |
| DURATION | DURATION | DURATION |
| ANCHOR-PERSON | READER(S) | - |
| ADVERTISEMENTS(YES/NO) | ADVERTISEMENTS(YES/NO) | ADVERTISEMENTS(YES/NO) |
| HEADLINES(YES/NO) | HEADLINES(YES/NO) | HEADLINES(YES/NO) |

Table 4.2 Meta Information Required for Conventional News Delivery

| TV | RADIO | NEWS PAPER |
|---|---|---|
| NEWSGROUP | NEWSGROUP | NEWSGROUP |
| DATE | DATE | DATE |
| TIME | TIME | - |
| LANGUAGE | LANGUAGE | LANGUAGE |
| TELECAST-STATION | RADIO-STATION | PUBLISHED FROM |
| DURATION | DURATION | NO-OF-PAGES |
| ANCHOR-PERSON | READER(S) | EDITOR(S) |
| ADVERTISEMENTS(YES/NO) | ADVERTISEMENTS(YES/NO) | ADVERTISEMENTS(YES/NO) |
| HEADLINES(YES/NO) | HEADLINES(YES/NO) | HEADLINES(YES/NO) |

For search purposes, the multimedia database has to be indexed [Foul97]. Some of the meta knowledge required for the Personalized News Delivery (PND) and Conventional News Delivery (CND) differ. In PND, we may have to add more meta knowledge regarding the content of the document like keywords, place of incident, etc. But in the conventional news access, the keyword is not very important. User asks for either the whole newspaper or some complete sections of the news paper.

Apart from this meta information, some of the domain knowledge also has to be stored at the server sites. This domain knowledge is used by the system for data abstraction to do a relaxed or neighborhood search. There are three types of abstractions required at the server site for the system to co-operate with the user: ***Temporal Abstraction, Spatial Abstraction*** and ***Content Abstraction.*** The temporal abstraction provides a neighborhood range for a time based information. The spatial abstraction decides how to find the relaxed answer with respect to a geographical location of the news (for example: India, Canada). The content abstraction provides the neighborhood answer for a particular content. This is on the basis of the semantic meaning of that particular word(s).

## 4.2 UIL Syntax & Semantics : Selection Phase

The selection phase in UIL consists of the creation of the query to access the news documents. The selection phase consists of 4 clauses: ***Target clause, Range clause, Qualification clause*** and ***Presentation clause***. The syntax and semantics of the commands in each clause are explained below.

### 4.2.1 Target Clause

In this clause, UIL provides the selection of the required media type for presentation. The system provides 4 primitive media types for document: ***VIDEO, AUDIO, TEXT*** and ***IMAGE***. User can select one or any possible combination of these types. Some of these documents can be displayed on line or down-loaded to the client machine or hard copied onto the client's printer. Among Display, Save, and Print options, at-least one option is necessary for the query processing. The user can specify all of them, if required. Apart from these actual news documents, the user can also ask for any details (textual meta information) of the news documents.

### 4.2.1.1 Display Option

This option delivers the news online on the client machine. The news documents are delivered through the monitor screen and/or through the attached speakers. The syntax and semantics for the possible commands are given below. This can be followed by the NO-AD command for an advertisement free access which is explained in Section 4.2.1.5.

**UIL Syntax** : *DISPLAY* video
**UIL Semantics** : This will deliver the video document(s) on the client machine. In place of video, any other type can be used

**UIL Syntax** : *DISPLAY* audio+text
**UIL Semantics** : "+" sign corresponds to the concurrent delivery. This will provide audio document and the text document concurrently. Audio+text can be replaced by any possible combination of the primitive types like Audio+Image, Audio+Text+image.

**UIL Syntax** : *DISPLAY* audio, video
**UIL Semantics** : The video document will be followed by the audio document for presentation. Any number of items can be specified in this way. They will be presented in the specified order.

**UIL Syntax** : *DISPLAY* newspaper
**UIL Semantics** : This will provide the syntax for conventional type news delivery for accessing the whole newspaper. Other conventional news delivery media types TV and RADIO can also be used instead of news paper.

**UIL Syntax** : *DISPLAY* details
**UIL Semantics** : As details, any information regarding the system or documents can be specified. For example, Available sites, cost of retrieval from each site, meta information of the documents, etc.

### 4.2.1.2 Save Option

This option down-loads the news documents on the client machine. If the document is not in urgent demand, this command can be followed by the *BEFORE* command for an efficient operation. *BEFORE* command is explained in the Section 4.2.1.4. With this time specification, the document is down loaded before the specified time. Also this can be followed by the NO-AD command for an advertisement free access which is explained in 4.2.1.5. The syntax and semantics for the possible options with *SAVE* are given below.

**UIL Syntax** : *SAVE* text
**UIL Semantics** : This will down-load the text document on the client machine. In place of text, any other type can be used, if technology exist to store them in the client machine.

**UIL Syntax** : *SAVE* text, image
**UIL Semantics** : The text and image document will be down-loaded. Any number of items can be specified in this order.

**UIL Syntax** : *SAVE* newspaper
**UIL Semantics** : This will down-load the whole newspaper to the user. Conventional news delivery media types (TV, RADIO) can also be used instead of news paper, if the client site has the technology to store them.

If technology permits any type of documents can be downloaded with SAVE command. In that case, any attribute that appears in the DISPLAY option can be included here also.

### 4.2.1.3 Print Option

This option prints the news documents on the client printer. If the document is not in urgent demand, this command can be followed by the BEFORE command for an efficient operation. BEFORE command is explained in the Section 4.2.1.4. Also this can be followed by the NO-AD command for an advertisement free access. That is explained in Section 4.2.1.5. The syntax and semantics for the possible options with this command are given below.

**UIL Syntax** : *PRINT* text
**UIL Semantics** : Prints the text files to the clients printer.

**UIL Syntax** : *PRINT* text, image
**UIL Semantics** : Prints the text and image files to the clients printer.

**UIL Syntax** : *PRINT* newspaper
**UIL Semantics** : This will print the whole newspaper in the client printer.

### 4.2.1.4 Time Specification

This option increases the efficiency of the system externally. User can demand this for cost efficient retrieval. If the user is not in an urgent need of getting the news then she can ask the system to get the news ready before a specified time through this command. The system can fetch the documents when the transmission cost is the least in that specified time interval. If there is cooperation from agents, the system can search with the agents in the nearby sites for these documents. If they happen to bring the same item before this specified time, the system can bring it from them. This way the cost of transmission can be reduced. This command can followed only by SAVE or PRINT commands. The DISPLAY command needs the documents to be delivered on iine. The syntax and semantics for this command are given below. This is an ***optional*** command.

UIL Syntax : *BEFORE* time
UIL Semantics : The news will be saved or printed before the specified time. This option is only related to SAVE or PRINT commands.

### 4.2.1.5 Advertisement Free Delivery

This is used for an advertisement free news environment and can be followed by any type of news delivery. The syntax and semantics for this command are given below. This is an ***optional*** command.

UIL Syntax : *WITH NO-AD*
UIL Semantics : Provides documents, removing all the advertisement attached with it.

## 4.2.2 Range Clause

In the news delivery system, news documents are divided according to the subject in separate sections. The range clause specifies to the search controls for the UIL by directing the search to required section(s) and/or to required site(s). These control options are optional for the

UIL query. If they are included, the system restricts the search in these sections. But if they are not included in the query, the system searches all possible sections at all possible places. There are two ways of search control available in UIL: ***Database or Site based*** and ***Subject based.***

### 4.2.2.1 Database Oriented Search

Database oriented search control is similar to the relation based search control in the existing query languages. The only difference is that UIL is applied on the complete site whereas other QLs are applied on the relation of a database. This is an ***optional*** command. The syntax and semantics for the available search control are given below.

**UIL Syntax** : *DB* Montreal-site
**UIL Semantics** : Only the Montreal site is to be searched for the news access

**UIL Syntax** : *DB* Montreal-site, Toronto-site
**UIL Semantics** : Montreal and Toronto sites are to be searched. Any number of sites can be included in the search in this way.

### 4.2.2.2 Subject Oriented Search

The second search control is based on subject of the news. Upon this control, only the documents of specified subjects are searched. This is an ***optional*** command. The syntax and semantics for the available search control are given below.

**UIL Syntax** : *SUBJECT* world
**UIL Semantics :** Only the section WORLD is to be searched

**UIL Syntax** : *SUBJECT* world, entertainment
**UIL Semantics** : The section WORLD and ENTERTAINMENT are to be searched. If any matching document found in any one of these subjects, they have to be retrieved. Any number of subjects can be specified in this way.

**UIL Syntax** : *SUBJECT* world+sports
**UIL Semantics** : Here, '+' operator stands for concurrent processing. In this search, the document is retrieved only if it is related to both of these subject. Any number of subjects can be specified in this way.

## 4.2.3 Qualification Clause

Qualification clause consists of one or more boolean conditions to qualify the document for retrieval. Atleast one condition is mandatory for the system to start the search. If there are more than one condition, they are connected with the logical operators like AND or OR. These conditions are of comparison type. In each condition the value of a meta-information (attribute) is matched with the specified value. This attribute can be replaced by a method in the object or any aggregate function applied on any attribute. There are five different types of matching in UIL. They are Strict Matching, Partial Matching, Approximate Matching, Relaxed Matching and Schema-less matching. They are explained in the previous chapter in Section 3.4.3.

UIL syntax and semantics for the qualification clause are given below. In those examples, attribute can be replaced by an object or a return value of a method or an aggregate function. Also instead of =, any other comparison operators can be used from Less than (<). Greater than (>). Not equal to (#).

**UIL Syntax** : *WHERE* attribute = value
**UIL Semantics** : The attribute is strictly matched with the value.

**UIL Syntax** : *WHERE* att1 = val1 AND att2 = val2
**UIL Semantics** : Here two condition are connected with AND. Instead of AND, OR also can be used as a connector. Any number of conditions can be connected using AND and/or OR connectors.

**UIL Syntax** : *WHERE* attribute = "xxx*"
**UIL Semantics** : The value of the meta-information (attribute) is partially matched with the string "xxx".

**UIL Syntax** : *WHERE* attribute ~= 10
**UIL Semantics** : The meta-information (attribute) value is matched approximately with 10.

**UIL Syntax** : *WHERE* attribute ^= "xxx"
**UIL Semantics** : A relaxed search is performed for meta-information(attribute) to match a neighborhood answer with "xxx".

**UIL Syntax** : *WHERE UNKNOWN* = "xxx"
**UIL Semantics** : A strict search is performed for to match the specified value, "xxx" with the value of any available attribute.

### 4.2.4 Preference Clause

This consists of one or more boolean conditions to specify the user's preferences. These conditions are applied after the selection of the qualified documents, but before presentation. If the preferred items are present at the server, they will be sent first to the user. Only if the user asks for more, the rest will be sent. If there are no preferred documents found among the selected items, this clause acts as void and all the selected items are sent to the user.

The preference conditions have the similar syntax and semantics of the qualification conditions. They differ only at the command word level where qualification conditions start with ***WHERE*** and preference starts with ***PREFER***. All the 5 types of matching can be included here also. UIL syntax and semantics for the preference clause are given below. In those examples, attribute can be replaced by an object or a return value of a method or an aggregate function. Also instead of =, any other comparison operators can be used from Less than (<), Greater than (>), Not equal to (#). This is an ***optional*** command.

**UIL Syntax** : *PREFER* attribute = value
**UIL Semantics** : The meta-information(attribute) is strictly matched with the value.

**UIL Syntax** : *PREFER* att1 = val1 AND att2 = val2
**UIL Semantics** : Here two condition are connected with AND. OR also can be used as a connector. Any number of conditions can be connected using AND and/or OR connectors.

**UIL Syntax** : *PREFER* attribute = "xxx*"
**UIL Semantics** : The value of the meta-information (attribute) is partially matched with the string "xxx"

**UIL Syntax** : *PREFER* attribute ~= 10
**UIL Semantics** : The meta-information(attribute) value is matched approximately with 10.
**UIL Syntax** : *PREFER* attribute ^= "xxx"

**UIL Semantics** : A relaxed search is performed for meta-information(attribute) to match a neighborhood answer with "xxx"

## 4.2.5 Ordering

This clause specifies the order in which the selected items are to be presented. The qualified or retrieved document(s) are presented in the ordering option. Even among the preferred documents, they are ordered according to the ordering conditions. We can have a number of ordering statements. The related documents are ordered in the specified order. The ordering conditions have the same syntax and semantics as the qualification conditions and they differ only at the command level. The qualification conditions start with ***WHERE*** and ordering conditions starts with ***ORDER***. All the 5 types of matching can be included here also. Ordering is an ***optional*** command.

**UIL Syntax** : *ORDER* attribute = value
**UIL Semantics** : The meta-information(attribute) is strictly matched with the value for ordering.

**UIL Syntax** : *ORDER* att1 = val1 *AND* att2 = val2
**UIL Semantics** : Here two condition are connected with AND. OR also can be used as a connector. Any number of conditions can be connected using AND and/or OR connectors

**UIL Syntax** : *ORDER* att1 = val1, val2
**UIL Semantics** : In this first the documents matching val1 is presented and then that matching val2. Any number of values can be given to match an meta-information (attribute) for ordering the presentation.

**UIL Syntax** : *ORDER* attribute = "xxx*"
**UIL Semantics** : The value of the attribute is partially matched with the string "xxx".

**UIL Syntax** : *ORDER* attribute ^= 10
**UIL Semantics** : The meta-information(attribute) value is matched approximately with 10.

**UIL Syntax** : *ORDER* attribute ~= "xxx"
**UIL Semantics** : A relaxed search is performed for attribute to match a neighborhood answer with "xxx".

### 4.2.6 Complete UIL Syntax for Selection Phase

```
DISPLAY       Output
SAVE          Output
PRINT         Output1
BEFORE        Time
NO-AD
FROM-DB       Db
SUBJECT       Subject
WHERE         Condition ((AND|OR) Condition)*
PREFER        Condition ((AND|OR) Condition)*
ORDER         Condition ((AND|OR) Condition)*
Output        : op (,op)* | conv-op
op             : TEXT | IMAGE | AUDIO | VIDEO | TEXT+IMAGE |
                AUDIO+IMAGE | TEXT+AUDIO | AUDIO+IMAGE+TEXT
conv-op       : NEWSPAPER | TV | RADIO
Output1       : op1 (,op1)* | conv-op1
op1           : TEXT | IMAGE | TEXT+IMAGE
conv-op1      : NEWSPAPER
Time          : time, day
time          : String (example 09:00 PM)
day           : String (example 11-NOV-1996)
Db            : (db, (,db)*
db            : String  (Name of site)
Sub           : (sub. (.sub)*, (+sub)*)*
sub           : String (Name of Subject or Section)
Condition     :  Attribute   EqualityType   Value
EqualityType  : = | < | > | # | ~= | ^=
Attribute     : String  (Name of the Attribute)
Value         : String
```

## 4.3 UIL Syntax & Semantics : Presentation Phase

The system delivers the documents to the user after matching them with the conditions specified in the selection phase. The delivery of these documents can take a long time if many documents are selected. The system does not expect the user to be passive during this interval. It provides many different controls for the user for interacting with the system. These controls help the user to adapt the presentation of the news to her own choice. The controls depend on the media type of the document to be presented. Along with the presentation of a document the corresponding controls are active and visible to the user to operate on them. These controls are classified in four different sections as follows and they are explained in the following sections.

1. *Display Controls during Video/Audio Presentation during PND*
2. *Display Controls during Text/Image Presentation during PND*
3. *Display Controls during TV or Radio presentation during CND*
4. *Display Controls during News Paper presentation during CND*

## 4.3.1 Display Controls of Video/Audio Presentation (PND)

**UIL Syntax** : *FF*
**UIL Semantics** : The current display of the document is fast-forwarded.

**UIL Syntax** : *REW*
**UIL Semantics** : The current display of the document is fast rewound.

**UIL Syntax** : *PAUSE*
**UIL Semantics** : The display is halted for a short duration. User will be able to view the still image of the current frame.

**UIL Syntax** : *PLAY*
**UIL Semantics** : The display is started or the halted display is resumed with this command

**UIL Syntax** : *STOP*
**UIL Semantics** : The display is stopped for a long duration or permanently. The current document is closed at the sever site.

**UIL Syntax** : *SAVE*
**UIL Semantics** : The current document is to be down loaded to the client machine.

**UIL Syntax** : *EXIT.*
**UIL Semantics** : Exit the news delivery system

**UIL Syntax** : *STEPFW*
**UIL Semantics** : The current document is paused and forwarded by frame by frame.
(Only for Video)

**UIL Syntax** : *STEPREW*
**UIL Semantics** : The current document is paused and rewound by frame by frame
(Only for Video)

**UIL Syntax** : *AUDIO-VERSION*
**UIL Semantics** : Switch the current presentation to Audio presentation. (Only for Video)

**UIL Syntax** : *VIDEO-VERSION*
**UIL Semantics** : Switch the current presentation to Audio presentation. (Only for Audio)

**UIL Syntax** : *TEXT-VERSION*
**UIL Semantics** : Switch the current presentation to Text or Image presentation.

**UIL Syntax** :*ADD-PICTURES*
**UIL Semantics** : Add image document together with the current presentation. This will provide a concurrent presentation of audio and image documents. (Only for Audio)

**UIL Syntax** : *ADD-TEXT*
**UIL Semantics** : Add text document together with the current presentation. This will provide a concurrent presentation of audio and text documents. (Only for Audio)

**UIL Syntax** : *MORE-ON*(subject/keyword)
**UIL Semantics** : System brings other documents which provides more details of the specified keyword or subject.

**UIL Syntax** : *SKIP*
**UIL Semantics** : The current document is skipped from the presentation and other selected documents, if any will be presented next.

**UIL Syntax** : *CHANGE-CONTENT*(*string*)
**UIL Semantics** : System stops the presentation of the current document and switch to other documents with the specified keywords.

**UIL Syntax** : *HEADLINES*
**UIL Semantics** : Only the headlines of the current document is presented.

**UIL Syntax** : *NO-AD*
**UIL Semantics** : Provides an advertisement free news delivery from now on.

## 4.3.2 Display Controls of Text/Image Presentation (PND)

**UIL Syntax** : *NEXT-PAGE*
**UIL Semantics** :The displayed document is flipped to the next page.

**UIL Syntax** : *PREV-PAGE*
**UIL Semantics** : The displayed document is flipped to the previous page.

**UIL Syntax** : *START*
**UIL Semantics** : The current document is flipped to the first page.

**UIL Syntax** : *FINISH*
**UIL Semantics** : The current document is flipped to the last page.

**UIL Syntax** : *GOTO*(page)
**UIL Semantics** : The system turns the document over to show the specified page.

**UIL Syntax** : *PRINT*
**UIL Semantics** :The current document is printed at the client site.

**UIL Syntax** : *STOP*
**UIL Semantics** : The display is stopped permanently. The current document is closed at the server site.

**UIL Syntax** : *SAVE*
**UIL Semantics** : The current document is to be down loaded to the client.

**UIL Syntax** : *EXIT*
**UIL Semantics** : Exit the news delivery system

**UIL Syntax** : *VIDEO-VERSION*
**UIL Semantics** : Switch the current presentation to Video presentation. The document will be of the same content.

**UIL Syntax** : *AUDIO-VERSION*
**UIL Semantics** : Switch the current presentation to Audio presentation.

**UIL Syntax** : *ADD-AUDIO*
**UIL Semantics** : Add audio document together with the current presentation. This will provide a concurrent presentation of audio and text documents.

**UIL Syntax** : *ADD-PICTURES/TEXT,*
**UIL Semantics** : Add image/text document together with the current presentation. This will provide a concurrent presentation of image and text documents.

**UIL Syntax** : *MORE-ON*(subject/keyword)
**UIL Semantics** : System brings other documents which provides more details of the specified keyword or subject.

**UIL Syntax** : *SKIP*
**UIL Semantics** : The current document is skipped from the presentation and other selected documents, if any will be presented next.

**UIL Syntax** : *CHANGE-CONTENT*(*string*)
**UIL Semantics**: System stops the presentation of the current document and retrieve other with the specified keywords.

**UIL Syntax** : *GOTO*(section)
**UIL Semantics** : The system presents the specified section

**UIL Syntax** : *FIND*(*string*)
**UIL Semantics : System finds the specified string and show that page to the user with that particular string highlighted.**

**UIL Syntax** : *NO-AD*
**UIL Semantics** : Provides an advertisement free news delivery from now on.

## 4.3.3 Display Controls of TV/Radio Presentation (CND)

**UIL Syntax** : *FF*
**UIL Semantics** : The current display of the news is fast-forwarded.

**UIL Syntax** : *REW*
**UIL Semantics** : The current display of the news is fast rewound.

**UIL Syntax** : *PAUSE*
**UIL Semantics** : The display is halted for a short duration. User will be able to view the still image of the current frame.

**UIL Syntax** : *PLAY*
**UIL Semantics** : The display is started or the halted display is resumed with this command

**UIL Syntax** : *STOP*
**UIL Semantics** : The display is stopped permanently. The current document is closed at the server site.

**UIL Syntax** : *SAVE*
**UIL Semantics** : The current document is to be down loaded to the client machine.

**UIL Syntax** : *EXIT.*
**UIL Semantics** : Exit the news delivery system

**UIL Syntax** : *MORE-ON*(subject/keyword)
**UIL Semantics** : System brings other days news which provides more details of the specified keyword or subject.

**UIL Syntax** : *SKIP*
**UIL Semantics** : The current document is skipped from the presentation and other selected documents, if any will be presented next.

**UIL Syntax** : *CHANGE-CHANNEL*(*string*)
**UIL Semantics** : System stops the presentation from the current station and switch to the specified station.

**UIL Syntax** : *GET-NEWS-AT*(time)
**UIL Semantics** : System presents the news which was broadcasted at the specified time of the day of the receiving news.

**UIL Syntax** : *GET-NEWS-ON*(day)
**UIL Semantics** : System presents all the news which wasbroadcasted on the specified day.

**UIL Syntax** : *HEADLINES*
**UIL Semantics** : Only the headlines of the current news segment is presented.

**UIL Syntax** : *NO-AD*
**UIL Semantics** : Provides an advertisement free news delivery from now on.

## 4.3.4 Display Controls of News Paper Presentation (CND)

**UIL Syntax** : *NEXT-PAGE*
**UIL Semantics** :The displayed document is flipped to the next page.

**UIL Syntax** : *PREV-PAGE*
**UIL Semantics** : The displayed document is flipped to the previous page.

**UIL Syntax** : *START*
**UIL Semantics** : The current document is flipped to the first page.

**UIL Syntax** : *FINISH*
**UIL Semantics** : The current document is flipped to the last page.

**UIL Syntax** : *GOTO*(page)
**UIL Semantics** : The system turns the document over to show the specified page.

**UIL Syntax** : *PRINT*
**UIL Semantics** :The current document is printed at the client site.

**UIL Syntax** : *STOP*
**UIL Semantics** : The display is stopped permanently. The current document is closed at the server site.

**UIL Syntax** : *SAVE*
**UIL Semantics** : The current document is to be down loaded to the client.

**UIL Syntax** : *EXIT*
**UIL Semantics** : Exit the news delivery system

**UIL Syntax** : *MORE-ON*(subject/keyword)

**UIL Semantics** : System brings other days newspaper section which provides more details of the specified keyword or subject.

**UIL Syntax** : *SKIP*
**UIL Semantics** : The current news paper is skipped from the presentation and other selected item, if any will be presented next.

**UIL Syntax** : *CHANGE-NEWSPAPER*(*string*)
**UIL Semantics**: System stops the presentation of the current NEWS PAPER and retrieve other with the specified name.

**UIL Syntax** : *GOTO*(section)
**UIL Semantics** : The system presents the specified section

**UIL Syntax** : *FIND*(*string*)
**UIL Semantics** : System finds the specified string and show that page to the user with that particular string highlighted.

**UIL Syntax** : *GET-NEWS-ON*(day)
**UIL Semantics** : System presents all the news which wasbroadcasted on the specified day.

**UIL Syntax** : *HEADLINES*
**UIL Semantics** : Only the headlines of the current news paper is presented.

**UIL Syntax** : *NO-AD*
**UIL Semantics** : Provides an advertisement free news delivery from now on.

## 4.4 Sample Queries in UIL

There are two ways, users access the news: *Conventional* access and *Content-based* access. The sample queries for these two cases are shown in the following sections.

### 4.4.1 Conventional News Delivery (CND)

In the conventional access, the user wants to use the system as a conventional news media. She just demands a conventional news group or any of the properties of the conventional news media. The user does not want the system to select the news, but she will browse through. She just wants to continue the way she was receiving news. Some of the possible queries of conventional news access are shown in this section. The queries are only for the selection of the document. The presentation control will be activated and shown during the entire presentation.

*Query* **A1)** Get me today's "GLOBE & MAIL"

```
UIL :   DISPLAY      NEWSPAPER
        WHERE        NEWSGROUP = "GLOBE & MAIL"
```

**Note:** If the date, time, etc. are not specified, UIL will assume it as the latest or most recent one.

*Query* **A2)** Get the BBC channel

```
UIL :   DISPLAY      TV
        WHERE        NEWSGROUP = BBC
```

*Query* **A3)** Get me any French newspaper dated on Oct 17 which is published from Montreal.

```
UIL :   DISPLAY      NEWSPAPER
        WHERE        LANGUAGE = 'FRENCH" AND
                     DATE ="17-OCT-1996" AND
                     PUBLISHED FROM = "MONTREAL"
```

*Query* **A4)** Display the ENTERTAINMENT section of today's THE GAZETTE and save it.

```
UIL :   DISPLAY      NEWSPAPER
        SAVE         NEWSPAPER
        SUBJECT      ENTERTAINMENT
        WHERE        NEWSGROUP = 'THE GAZETTE" AND
                     DATE = "11-25-1996"
```

*Query* **A5)** Save today's THE GAZETTE before 8:00 PM.

```
UIL :   SAVE         NEWSPAPER
        BEFORE       TIME = "20:00"  AND
                     DATE = TODAY
        WHERE        NEWSGROUP = 'THE GAZETTE"   AND
                     DATE = TODAY
```

*Query* **A6)** Get me the yesterdays CBC news if it is available in the MONTREAL site.

```
UIL :   DISPLAY      TV
        FROM-DB      MONTREAL
        WHERE        NEWSGROUP = "CBC" AND
                     DATE = YESTERDAY
```

***Query* A7)** Get the details of all available ENGLISH Video news from Montreal and Toronto sites.

```
UIL :     DISPLAY        NEWSGROUP,DURATION,STATION  SITE, cost (SIT E)
          FROM-DB        MONTREAL,TORONTO
          WHERE          LANGUAGE - "ENGLISH" AND
                         TYPE - VIDEO
```

***Query* A8)** Get today's prime-time news from ABC without advertisement.

```
UIL :     DISPLAY        VIDEO
          WITH NO-AD
          WHERE          NEWSGROUP - "ABC"  AND
                         DATE - TODAY  AND
                         TIME - PRIMETIME
```

***Query* A9)** Get the SPORTS section of all available ENGLISH newspapers from Montreal site.

```
UIL :     DISPLAY        NEWSPAPER
          FROM-DB        MONTREAL
          SUBJECT        SPORTS
          WHERE          LANGUAGE - "ENGLISH"
```

## 4.4.2 Personalized News Delivery (PND)

In the content based news access, the user asks the system to search for a particular content. The user does not require the whole news from a news provider. The system filters the document on the basis of content. Many other attributes are also specified along with the content to narrow down the search to a "manageable number" of documents. Some of the possible queries of content based news access are shown in this section. The queries are only for the selection of the document. The presentation control will be activated and effective during the entire presentation.

***Query* B1)** Get the video on "XX" from subject "YY1" or "YY2"

```
UIL :     DISPLAY        VIDEO
          SUBJECT        YY1, YY2
          WHERE          KEYWORD - "*XX*"
```

***Query*** **B2)** Get the video news on "XX" from Montreal site. Prefer CNN.

```
UIL:   DISPLAY      VIDEO
       FROMDB       MONTREAL
       WHERE        KEYWORD = "*XX*"
       PREFER       NEWSGROUP = "CNN"
```

***Query*** **B3)** Get the AUDIO and IMAGES at the same time regarding the  Earthquake.

```
UIL:   DISPLAY      AUDIO+IMAGE
       WHERE        KEYWORD = "*EARTHQUAKE*"
```

***Query*** **B4)** Get every news, if it is somehow related to the city, Montreal.

```
UIL:   DISPLAY      VIDEO,AUDIO,TEXT
       WHERE        * = "*MONTREAL*"
```

***Query*** **B5)** Get the VIDEO on Canadian politics from  BBC with relaxation on place. (i.e. If Canadian politics is not available, North American politics will do)

```
UIL:   DISPLAY      VIDEO
       SUBJECT      POLITICS
       WHERE        PLACE ^= "CANADA"  AND
                    NEWSGROUP = "BBC"
```

***Query*** **B6)** Get the 8'O clock ABC news on US election with the  relaxation on the time.

```
UIL:    DISPLAY     VIDEO
      SUBJECT       POLITICS
      WHERE         KEYWORD = "* US ELECTION *"   AND
                    TIME ^= "8:00 PM"
```

***Query*** **B7)** Get the VIDEO news of US election if the cost is  approximately 2$.

```
UIL:   DISPLAY      VIDEO
       FROM         X
       SUBJECT      POLITICS
       WHERE        KEYWORD = "* US ELECTION *"   AND
                    cost(X) ~= 2
```

# 5

# *Implementation of User Interface*

User Interaction Language (UIL) is the communication language used in the News Delivery System. In the Client-Server architecture of the ENDS, the client communicates with the user and the server sites through the UIL. It has some additional features which are not found in a traditional query language. UIL is a domain specific language which makes use of domain knowledge to interpret the user inputs. These domain knowledge can be stored in server and/or client sites. The main difference of the UIL from the standard database query language is that the user can process on the retrieved data. The UIL provides different options to control the presentation of the retrieved data along with their retrieval. This is very useful in the case of multimedia presentation where cost of retrieval, time taken or the visual space required for presentation are important factors.

UIL is oriented towards two phases: ***Initial Query Generation*** and ***Command (Interrupt) Processing during Presentation.*** These commands (Interrupts) were explained in Chapter 4. In this chapter, the design of a Graphical User Interface (GUI) based on UIL is described. The inputs are based on a keyboard and mouse, but they can be adapted for speech or multimodal input.

## 5.1 User Characteristics

The main purpose of the user interface design is to improve the user's "performance". One way to measure the "performance" of a user is to use the amount of time and effort consumed to complete a task [Mayh92]. There are many factors which affect the performance. Knowledge of these factors help to create a good "user profile" for an application system. In the literature, these factors are classified into 3 different user characteristics: Psychological Characteristics; Knowledge and Experience Characteristics; and Job and Task Characteristics. To understand the user population and improve the user performance, we analyse these classes of factors as per the News application. We use a three point scale (Low, Moderate, and High) as a measure for several of these factors.

### 5.1.1 Psychological Characteristics

**Motivation:** Motivation of the user is an important factor in the performance of tasks requiring motor, cognitive, or perceptual skills [Mayh92]. In the news delivery system, all types of users are expected. For the users who have low motivation, the system should be easy to learn. For the moderately motivated users, the system should provide some benefits for using it. Highly motivated users should feel that the system is not very complex and not difficult to learn and is natural to use. To satisfy all these users, UI should be consistent, predictable, and simple to understand. The news application expects all types of users: from students to scholars, from laymen to intellectuals, etc. There will be users who are highly motivated out of their personal interest. Also there will be users who are with low motivation due to the fear of changing to electronic news delivery.

$\textit{Motivation} \quad \in \{\textit{Low, Moderate, High}\}$

### 5.1.2 Knowledge and Experience Characteristics

The user's knowledge and experience can be measured by different factors. These factors are relatively independent and important to design a good UI. They are explained below.

**Computer Literacy :** Computer Literacy corresponds to the prior experience with computer. It includes the use of mouse, familiarity with the different computer keys, and knowledge of some common operation such as saving, opening a file, etc. It is very important to know the level of computer literacy expected from the user's before designing the UI. In the news application, computer literacy can vary from low to high in different users.

*Computer Literacy* $\in$ *{ Low, Moderate, High }*

**Task Experience:** Task experience corresponds to the knowledge of the task domain. Our domain is news documents and the users will have low to high level of knowledge about them.

*Task Experience* $\in$ *{ Low, Moderate, High }*

**System Experience:** System Experience corresponds to the knowledge of a particular language or mode of interaction of the system. In the news delivery system, the knowledge of UIL varies from low to high in the users. The beginners will not have any idea about the syntax of the language. But after a number of use, they can become experts in the syntax.

*System Experience* $\in$ *{ Low, Moderate, High }*

**Interaction Skill:** This skill corresponds to the typing skill and the familiarity with the mouse for various selection. In the case of speech I/O, this will correspond to the user's understanding of the limit of speech recognizers. Since we have users with computer literacy from low to high, we assume the level of interaction also vary from low to high.

*Interaction Skill* $\in$ *{ Low, Moderate, High }*

### 5.1.3 Job and Task Characteristics

These characteristics determine the relative emphasis to be put on ease of learning versus ease of use. These factors help to determine the amount of syntactic and semantic assistance required in the interface. Different factors of this category are analysed below.

**Frequency of Use:** Frequency of use affects the interface design in two ways. The users who use the system frequently are usually willing to spend more time in learning. So efficiency of operation is more important than ease of learning. But for low frequency users, the system should be designed for ease of learning. In the news delivery system, most of the users get news daily or atleast weekly.

*Frequency of use* $\in$ *{ Moderate, High }*

**Primary Training:** The amount of primary training determines how easy to learn the interface must be. In the news delivery system users will not have any primary training.

*Primary Training* $\in$ *{ None }*

**System Use:** Since there are many ways for receiving the news, the use of the electronic news delivery system is discretionary. When the system use is discretionary, ease of learning is very important to create motivation. When the system is mandatory, users get training to learn the system.

*System Use* $\in$ *{ Discretionary }*

**Task Importance:** The task importance will influence how much time, the users are willing to give to learn the system. The task importance in news delivery can range from low to high. Some of the users will not find this task very important, since there are many other ways to get the news. Some other users will find the system very important since it can save a lot of time for personalised news delivery (for example, following their portfolios in a stock market).

*Task Importance* $\in$ *{Low, Moderate, High}*

## 5.2 Different Tasks in UIL Query Creation

In order to specify the dialog between the user and the interface, different jobs performed by the users are to be identified. Each job consists of a number of Goals (Tasks). A goal can be defined as something that the user tries to accomplish [Kier88]. Sometimes accomplishing a goal might require accomplishing one or more sub-goals. Therefore, the goal analysis has to be conducted before designing the dialog.

Users have some procedural knowledge to perform their jobs. These knowledge are modelled using GOMS method. GOMS model has 4 components [Kier88]: (i) G: Specified goals, (ii) O: Operation that constitute methods, (iii) M: Methods performed by user to achieve goals, and (iv) S: Selection rules when multiple methods exist.

We identify the user tasks in the news delivery system as a number of high level goals. Each of these high level goals consists of many sub-goals. These goals and sub-goals are shown in Table 5.1. Different methods for these goals and operation that constitute these methods are explained in Section 5.4.

Table 5.1 User Goals in the Electronic News Delivery System

| Goal | Description |
|---|---|
| Goal 1 | Specify the retrieval details<br>1.1 Specify the reception mode(s) (display, save, print) and<br>1.2 Specify the media type(s) of the document |
| Goal 2 | Specify the Search Options<br>2.1 Specify the Database oriented search<br>2.1.1 Specify the Site(s) to be searched and/or<br>2.1.2 Specify the connectors (AND or OR) between each sites<br>2.2 Specify the Subject based search<br>2.2.1 Specify the Subject(s) to be searched and/or<br>2.2.2 Specify the connectors (AND and OR) between subjects |
| Goal 3 | Specify the Qualification details<br>3.1 Specify the Attribute and<br>3.2 Specify the Equality Type and<br>3.3 Specify the Value for that Attribute<br>3.4 Repeat the selection as many times as required |
| Goal 4 | Specify the Presentation Details<br>4.1 Specify the Preference<br>4.1.1 Specify the Attribute and<br>4.1.2 Specify the Equality type and<br>4.1.3 Specify the Value for that attribute<br>4.1.4 Repeat the above 3 specifications as required<br>4.2 Specify the Ordering<br>4.2.1 Specify the Attribute and<br>4.2.2 Specify the Equality Type and<br>4.2.3 Specify the Value for that attribute<br>4.2.4 Repeat the above 3 specifications as required |
| Goal 5 | Specify the Presentation Controls<br>5.1 Specify the required Presentation Control(s) which is active with the presentation |

## 5.3 Design of Dialogue Style

After the goals and sub-goals are identified, the next step in the UI design is to define the specific action or action sequence that are to be performed by the user and the interface in order to accomplish these goals. The action sequence is influenced by the "dialog design". The first step in dialog design is *identifying potentially appropriate dialog styles* using the given the user characteristics.

### 5.3.1 Appropriate Dialog Style

The *Cell Matrix Method* [Mayh92] proposes a strategy for selecting an appropriate set of dialog styles for an application. The Cell Matrix with the different dialog styles and user profile is shown in Table 5.2. In this matrix, the top most row lists the different dialog styles from left to right, and the left-most column lists the user characteristics from top to bottom. Each Cell in the matrix holds a particular value of the user characteristics in that row, for which the dialog style in that column would be appropriate.

Table 5.2 Appropriate Dialog Styles - Cell Matrix Method [Mayh92]

| | **Menu** | **Fill-in forms** | **Question& Answers** | **Command language** | **Function keys** | **DirectMan-ipulation** | **Natural Language** |
|---|---|---|---|---|---|---|---|
| **Motivation** | Low | Low, Moderate | Low | High | Low<br>High | Low | Low |
| **Typing Skill** | Low | Moderate, High | Moderate<br>High | Moderate<br>High | Low | Low | High |
| **System Experience** | Low | Low, Moderate | Low<br>Moderate | High | Low<br>High | Moderate | Low |
| **Task Experience** | Low | Moderate, High | Low | High | Moderate<br>High | Moderate<br>High | High |
| **Computer Literacy** | Low | Moderate, High | Low | High | Moderate<br>High | Low | Low |
| **Frequency of use** | Low | Moderate, High | Low | High | Low<br>High | Low | Low |
| **Primary Training** | Little or None | Little or None | Little or None | Formal | Little or None | Some | Little or None |
| **System Use** | Discretionary | Discretionary | Discretionary | Mandatory | Discretionary | Discretionary | Discretionary |
| **Task Importance** | Low | Moderate | Low | High | Low<br>High | ]Low | Low |
| **Total** | | | | | | | |

After finding the user characteristics of the users in the news delivery system, we read across each row in the matrix by marking every cell that matches that particular user characteristics. All the cells are read in this way. Then the number of marks obtained for each dialog style are added. The total for each dialog style is tallied at the bottom of each column. The marking for the news delivery application is shown in Table 5.3.

Table 5.3 Appropriate Dialog Styles for UIL - First Pass

| | Menu | Fill-in forms | Question& Answers | Command language | Function keys | DirectManipulation | Natural Language |
|---|---|---|---|---|---|---|---|
| Motivation | Low | Low, Moderate | Low | High | Low<br>High | Low | Low |
| Typing Skill | Low | Moderate, High | Moderate<br>High | Moderate<br>High | Low | Low | High |
| System Experience | Low | Low, Moderate | Low<br>Moderate | High | Low<br>High | Moderate | Low |
| Task Experience | Low | Moderate, High | Low | High | Moderate<br>High | Moderate<br>High | High |
| Computer Literacy | Low | Moderate, High | Low | High | Moderate<br>High | Low | Low |
| Frequency of use | Low | Moderate. High | Low | High | Low<br>High | Low | Low |
| Primary Training | Little or None | Little or None | Little or None | Formal | Little or None | Some | Little or None |
| System Use | Discretionary | Discretionary | Discretionary | Mandatory | Discretionary | Discretionary | Discretionary |
| Task Importance | Low | Moderate | Low | High | Low<br>High | Low | Low |
| Total | 8 | 9 | 9 | 7 | 9 | 8 | 9 |

Once this initial marking is done, a second pass is made to inspect each unmarked cell further. This refined marking is shown in Table 5.4. This second pass is done to see whether the dialog style for users other than those indicated in the cell for each row or user characteristics, has any serious disadvantage. If the dialogue style does not have any serious disadvantages for that user characteristics, then the cell should be marked. For example, if the expected frequency of use is moderate or high, then the cell under "Menu" will not be marked at the initial phase. However menus do not carry any penalty for the frequent users. Therefore menus will get a mark on the low frequency use for the next pass. Even though the user characteristics noted in each cell do not match with the user characteristics in the user profile, they should still be marked if they do not pose any particular disadvantage for the users with characteristics listed in the user profile [Mayh92].

At the end of second pass, the total for each dialog styles are tallied at the bottom of each column as shown in Table 5.4. We have dialog styles Menus, Fill-in Forms, Question and Answers, Function Keys, and Natural Languages with the same total indicating that all these dialog styles match with the user profile. From these styles, we have to select the *most appropriate style(s)* considering the factors like Cost of implementation, relative importance of user characteristics, etc.

Table 5.4 Appropriate Dialog Styles for UIL - Second Pass

| | **Menu** | **Fill-in forms** | **Question& Answers** | **Command language** | **Function keys** | **DirectManipulation** | **Natural Language** |
|---|---|---|---|---|---|---|---|
| **Motivation** | Low | Low, Moderate | Low | High | Low<br>High | Low | Low |
| **Typing Skill** | Low | Moderate, High | Moderate<br>High | Moderate<br>High | Low | Low | High |
| **System Experience** | Low | Low, Moderate | Low<br>Moderate | High | Low<br>High | Moderate | Low |
| **Task Experience** | Low | Moderate, High | Low | High | Moderate<br>High | Moderate<br>High | High |
| **Computer Literacy** | Low | Moderate, High | Low | High | Moderate<br>High | Low | Low |
| **Frequency of use** | Low | Moderate, High | Low | High | Low<br>High | Low | Low |
| **Primary Training** | Little or None | Little or None | Little or None | Formal | Little or None | Some | Little or None |
| **System Use** | Discretionary | Discretionary | Discretionary | Mandatory | Discretionary | Discretionary | Discretionary |
| **Task Importance** | Low | Moderate | Low | High | Low<br>High | Low | Low |
| **Total** | 9 | 9 | 9 | 7 | 9 | 8 | 9 |

Among these selected dialog styles, further checking is done for the feasibility of the style in the news delivery system. The Question and Answer style and the Function key style are found to be not appropriate for the news delivery system. For the news delivery system, there may be a lot of information provided by the user for the personalised selection. Question and Answer make the application tedious by asking a lot of input. The answer should be specified when the question is asked. It does not give the flexibility in ordering the input. Since there are a lot of input for the query, the number of function keys will be large. These function keys can not provide an intuitive meaning. So the more the number of function keys, the more errors made by the user. Thus the Question and Answer and the Function Key styles are not appropriate dialog styles for the communication in the electronic news delivery system.

Based on the above discussions, the ***Menus***, ***Fill-in Forms***, and ***Natural Language*** styles are found to be appropriate for the UIL communication. In this work, a Menu based and a Fill-in Forms based user interfaces are developed. The Spoken Language System (SLS) for the UIL communication is being developed in the future work.

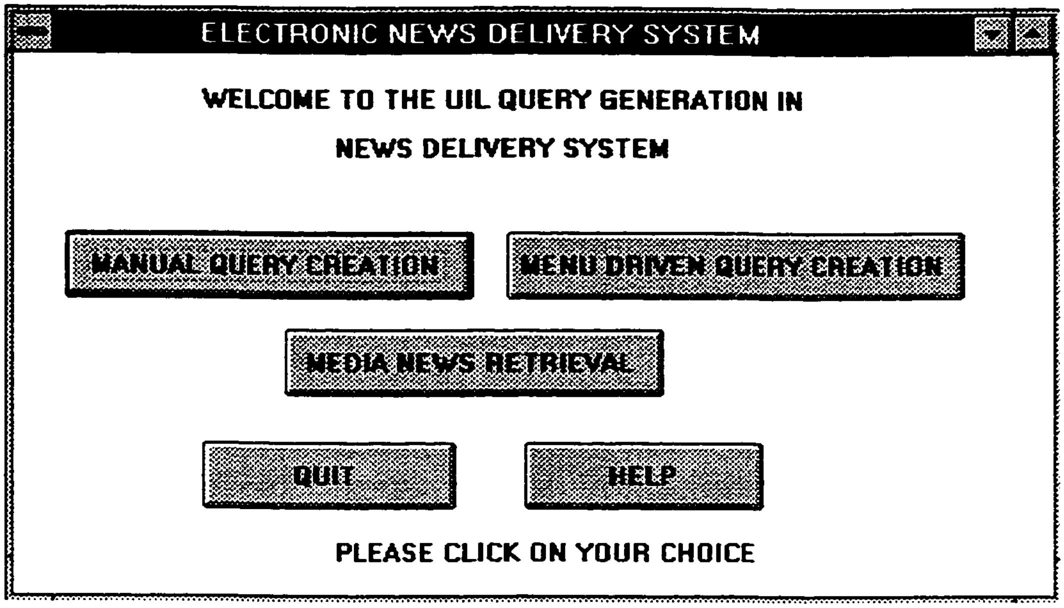
## 5.4 Proposed User Interface

This interface provides an easy and reliable option to specify all possible requirements of a user for news selection. There are two types of dialog styles available in the interface: Menu based query creation (using Menus) and Manual query creation (using Fill-in Forms). The opening Window for the Electronic News Delivery System is shown in Figure 5.1. The user can select the required query creation styles using the appropriate command buttons in this window. Also, the conventional type news delivery can also be demanded through this window.

For the naive users, the menu driven technique will be appropriate. At each stage, the system provides help. All possible values for each field are displayed in the window. The user need only to identify what she wants and *select* it. If there is any error in selection, the system provides an on line error message, so that user can *verify* the procedure using the help window. At any stage, user can verify the partially or fully generated query. If any *modification* required, that can be done very easily. The opening window for the ***Menu-based*** UIL generation is shown in Figure 5.3.

The disadvantage of the menu based system is that the user has to go through many steps in terms of windows, sub-windows, menus, etc. to finalise the query. An expert user may not like to waste time like that. Expert user may be knowing the syntax and possible values for the query constructs. To help those users, this interface provides a manual query creation window. In that window, the constructs will be displayed with their fields. The required values can be typed on those fields and the system generates the final query in UIL. In this way, user has to interact with only one window. If required, there is provision to go to the menu based creation at any time and continue the selection in that mode. In this way, all types of users can be comfortable with the system. Since the generated query can be displayed at any instant, the user can very easily go from the naive status to the expert status. The ***Fill-in-Form*** for the manual query creation is shown in Figure 5.2.

The interface for the different "Clauses" in the proposed UIL are given in the following subsections. The Procedure (Operation sequence of GOMS Model) for selecting each option in UIL is also explained in detail in those subsections.

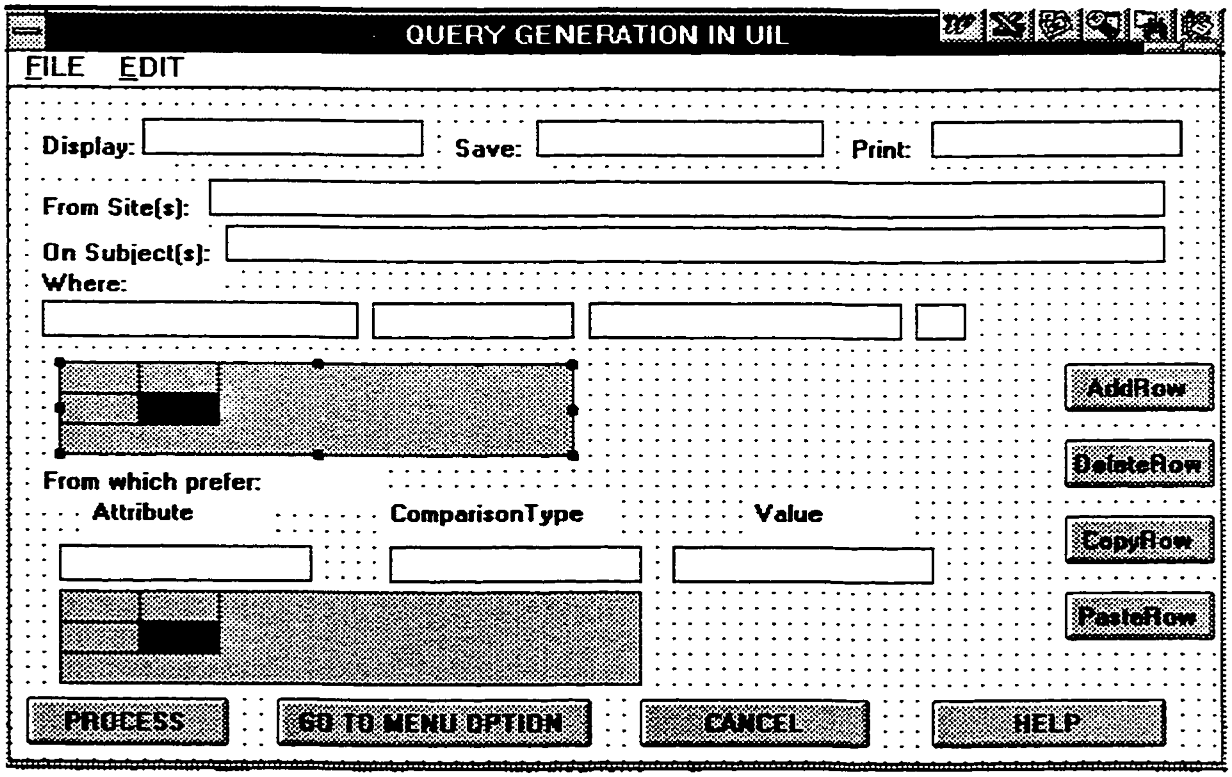


ELECTRONIC NEWS DELIVERY SYSTEM

WELCOME TO THE UIL QUERY GENERATION IN
NEWS DELIVERY SYSTEM

PLEASE CLICK ON YOUR CHOICE

Figure 5.1 The Opening Window for UIL Query Generation



QUERY GENERATION IN UIL

Figure 5.2 Fill-in-Form for Manual Query Creation

## 5.4.1 Target Clause

The following subsections explain how different retrieval options of UIL can be incorporated by this GUI. For each option, the required methods (procedure) are also explained.

### 5.4.1.1 Procedure for Selecting the Item for Display

In menu driven generation:

1. Select the "RETRIEVAL" from the window of Menu-based query generation (Figure 5.3).
2. A pull-down sub-menu appears (Figure 5.3).
3. Select the "DISPLAY" option from the pull-down sub-menu.
4. A pull-down sub-menu appears (Figure 5.3).
5. Select the required news media from the sub-menu.

In textual query generation:

1. Fill in the field for DISPLAY with the required media type(s) of news item. (Figure 5.2).

### 5.4.1.2 Procedure for Selecting the Item for Save

In menu driven generation:

1. Select the "RETRIEVAL" from the window of Menu-based query generation (Figure 5.3).
2. A pull-down sub-menu appears (Figure 5.3).
3. Select the "SAVE" option from the pull-down sub-menu.
4. A pull-down sub-menu appears.
5. Select the required news media from the sub-menu.

In textual query generation:

1. Fill in the field for "SAVE" with the required media type(s) of news item. (Figure 5.2).

### 5.4.1.3 Procedure for Selecting the Item for Print

In menu driven generation:

1. Select the "RETRIEVAL" from the window of Menu-based query generation (Figure 5.3).
2. A pull-down sub-menu appears (Figure 5.3).
3. Select the "PRINT" option from the pull-down sub-menu.
4. A pull-down sub-menu appears.
5. Select the required news media from the sub-menu.

In textual query generation:

1. Fill in the field for "PRINT" with the required media type(s) of news item. (Figure 5.2).

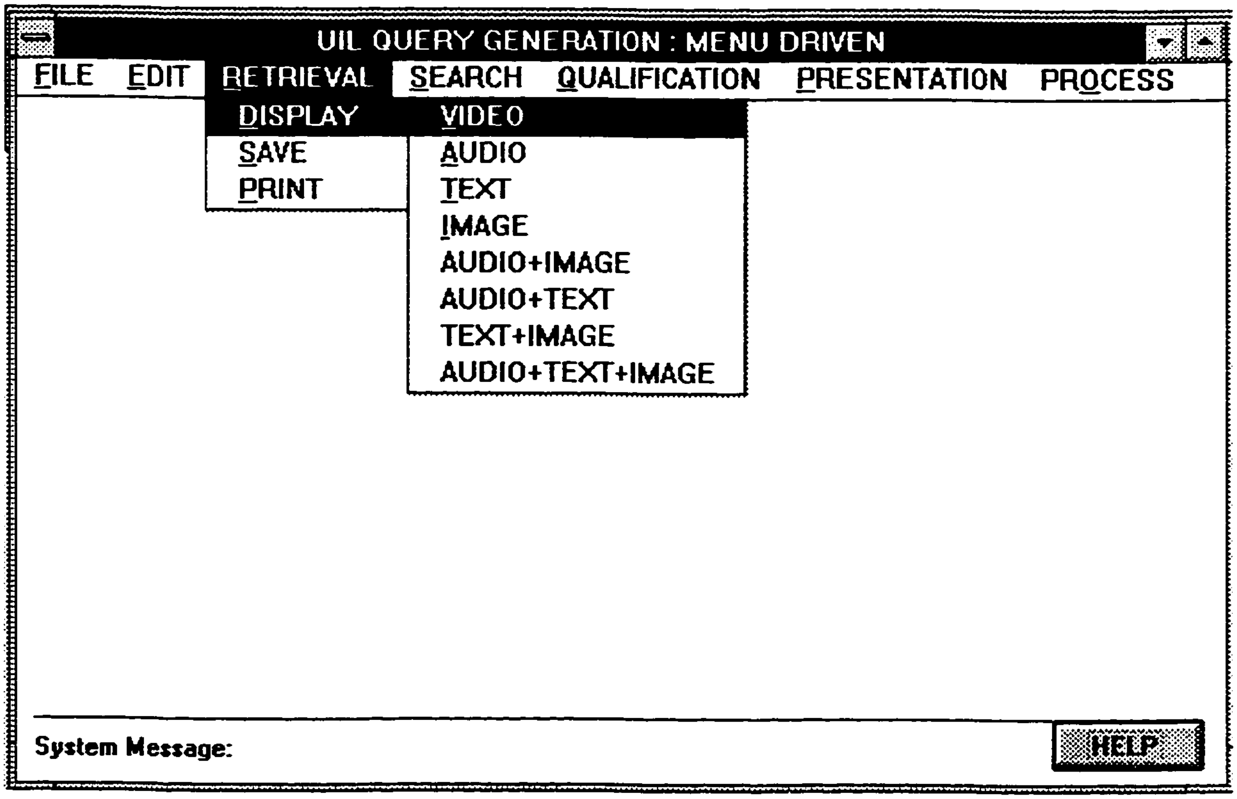


Figure 5.3 Opening Window for the Menu based Query Generation and the Retrieval Option

## 5.4.2 Search Clause

### 5.4.2.1 Procedure for Selecting the Subject

In menu driven query generation:

1. Select the "SEARCH" option from the first window of Menu-based query generation.
2. A pull-down sub-menu appears.
3. Select the "SUBJECT" from this menu.
4. A new Window for Subject Search appears (Figure 5.4).
5. Select the required subject from the list shown in the window (Figure 5.4).
6. Select the "ADDSUBJECT" button.
7. Select another subject if required from the list.
8. Select the appropriate connector.
9. Click on "ADDSUBJECT".
10. Repeat step 7-9 as many times as required.
11. Press OK to set.

In manual query creation:

1. Fill the field for "SUBJECT" with the required name(s) (Figure 5.2).

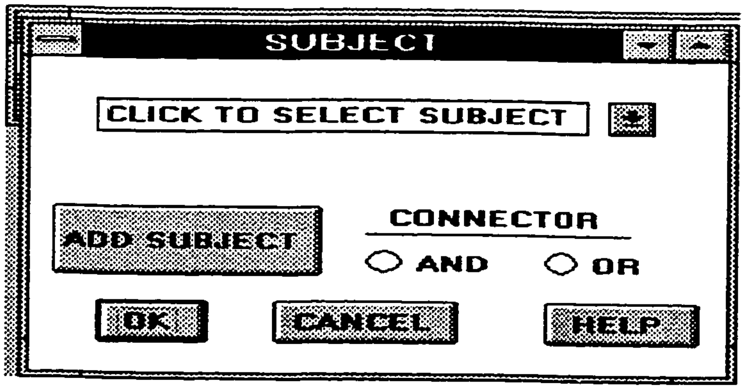


Figure 5.4 Window for Restricting the Search to Subject(s)

### 5.4.2.2 Procedure for Selecting the Server Site

In menu driven query generation:

1. Select the "SEARCH" option from the first window of Menu-based query generation.
2. A pull-down sub-menu appears.
3. Select the "DATABASE" from this menu.
4. A new Window for Database Search appears (Figure 5.5).
5. Select the required database from the list shown in this pop-up window(Figure 5.5).
6. Select the "ADDSITE" button.
7. Select another site if required from the list.
8. Select the appropriate connector.
9. Click on "ADDSITE".
10. Repeat step 7-9 as many times as required.
11. Press OK to set.

In textual query creation:

1. Fill the field for database with the required names. (Figure 5.2).

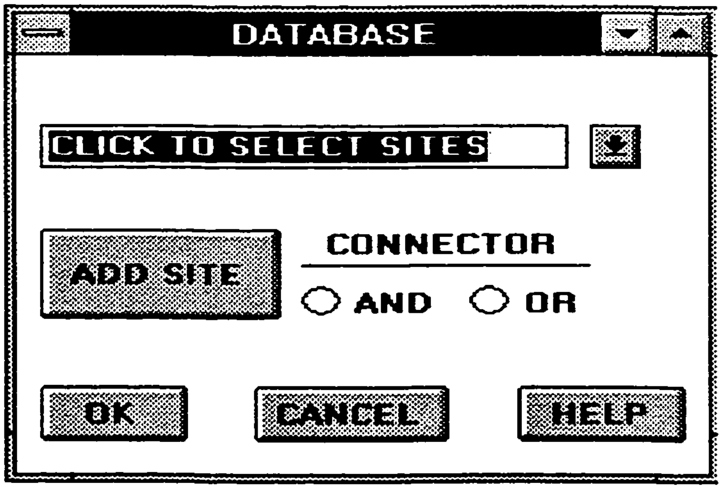
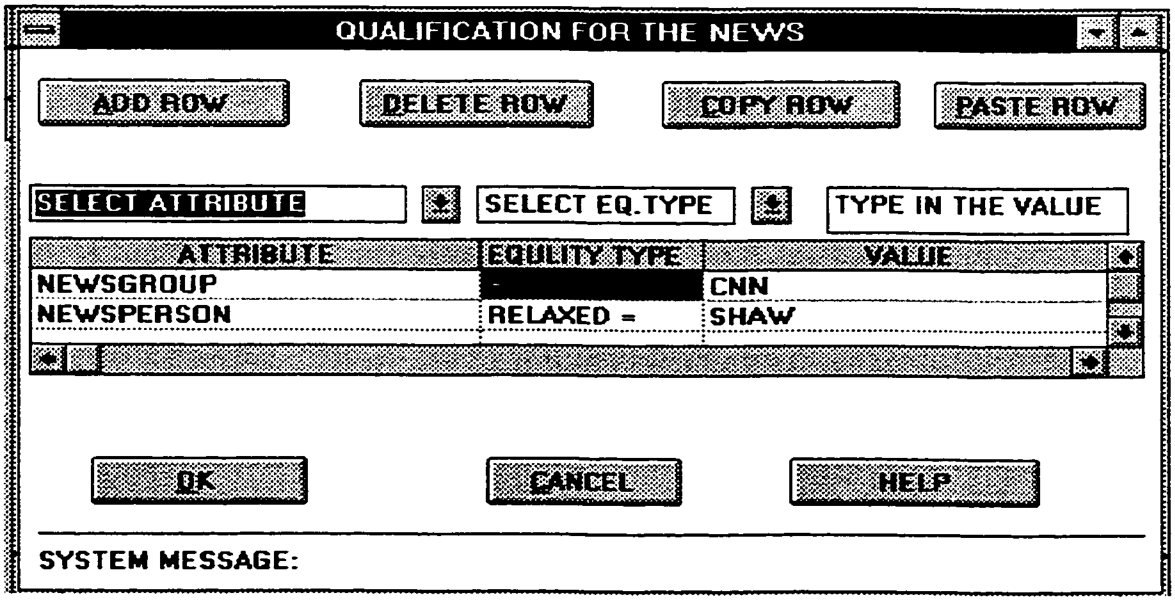


Figure 5.5 Window for Restricting the Search to Site(s)

### 5.4.3 Qualification Clause

In menu driven query generation:

1. Select the "QUALIFICATION" option from the first window of Menu-based query generation (Figure 5.3).
2. "QUALIFICATION" window pops out as shown in Figure 5.6.
3. Double Click on the "ATTRIBUTE" and a list of all attributes is displayed.
4. Select the required attribute from the attribute list.
5. Double Click on the "EQUALITY TYPE" and a list of all equality types is displayed.
6. Select the required equality type from the list.
7. Type in the required value for the field for value.
8. If more conditions are to be specified select the appropriate connector in the field.
9. Repeat Step 3-8 as many times as required.
10. Press "OK" to finish selection.



| ATTRIBUTE | EQULITY TYPE | VALUE |
|---|---|---|
| NEWSGROUP | - | CNN |
| NEWSPERSON | RELAXED = | SHAW |

SYSTEM MESSAGE:

Figure 5.6 Window for Specifying the Qualification Conditions for the Documents

In manual query generation:

1. Type in the required attribute for the corresponding field (Figure 5.2).
2. Type in the required equality for the corresponding field (Figure 5.2).
3. Type in the required value for the corresponding field (Figure 5.2).

## 5.4.4 Presentation Clause

### 5.4.4.1 Procedure for Specifying the Preference

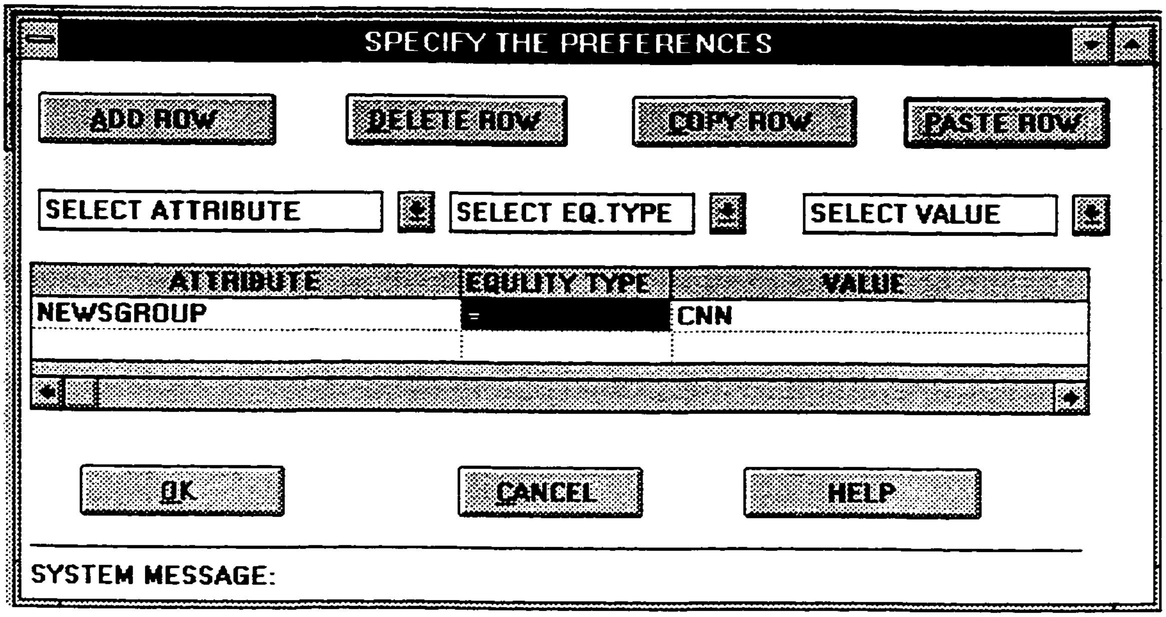
In menu driven query generation:

1. Select the "PRESENTATION" option from the first window of Menu-based query generation (Figure 5.3).
2. A pull-down menu appears.
3. Select "PREFERENCE" from this menu.
4. "PREFERENCE" window pops out as shown in Figure 5.7.
5. Double Click on the "ATTRIBUTE" and a list of all attribute is displayed.
6. Select the required attribute from the attribute list.
7. Double Click on the "EQUALITY TYPE" and a list of all equality types is displayed.
8. Select the required equality type from the list.
9. Type in the required value for the field for value.
10. If more conditions are to be specified select the appropriate connector in the field.
11. Repeat Step 5-10 as many times as required.
12. Press "OK" to finish selection.

In manual query generation:

1. Type in the required attribute for the corresponding field (Figure 5.2).

2. Type in the required equality type for the corresponding field (Figure 5.2).
3. Type in the required value for the corresponding field (Figure 5.2).



| ATTRIBUTE | EQUILITY TYPE | VALUE |
|---|---|---|
| NEWSGROUP | = | CNN |

Figure 5.7 Window for Specifying the Preferred Conditions

### 5.4.4.2 Procedure for Specifying the Ordering

In menu driven query generation:

1. Select the "PRESENTATION" option from the first window of Menu-based query generation (Figure 5.3).
2. A pull-down menu appears.
3. Select "ORDERING" from this menu.
4. "ORDERING" window pops out as shown in Figure 5.8.
5. Double Click on the "ATTRIBUTE" and a list of all attribute is displayed.
6. Select the required attribute from the attribute list.
7. Double Click on the "EQUALITY TYPE" and a list of all equality types is displayed.

8. Select the required equality type from the list.

9. Type in the required value for the field for value.

10. If more conditions are to be specified select the appropriate connector in the field.

11. Repeat Step 5-10 as many times as required.

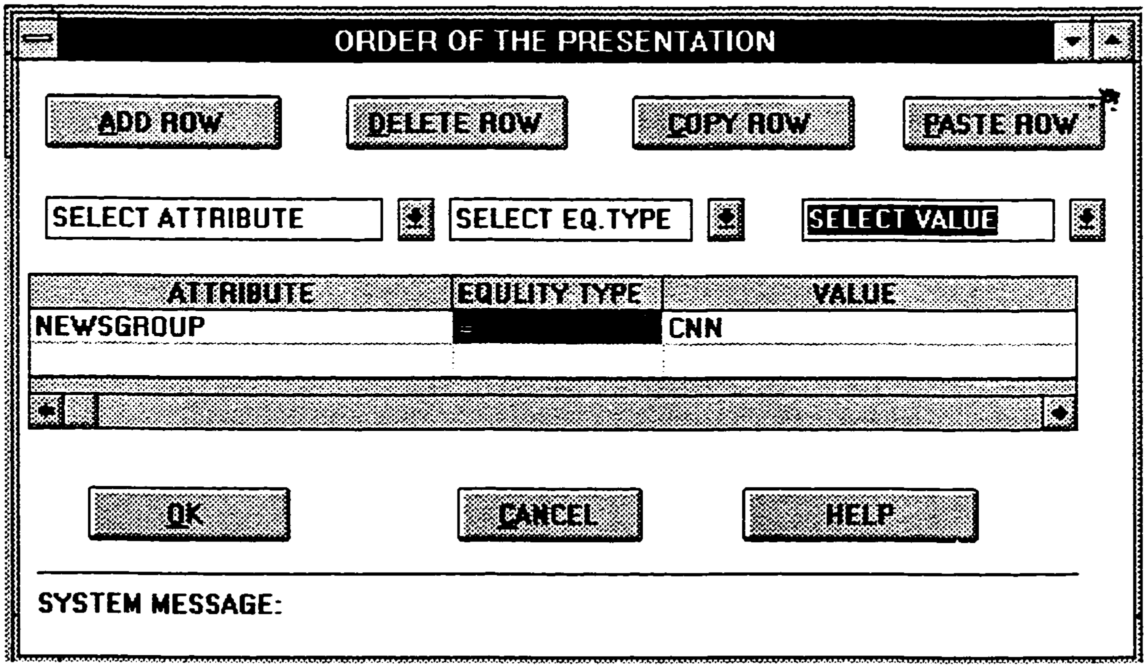
12. Press "OK" to finish selection.

In manual query generation:

1. Type in the required attribute for the corresponding field (Figure 5.2).

2. Type in the required equality type for the corresponding field (Figure 5.2).

3. Type in the required value for the corresponding field (Figure 5.2).



SELECT ATTRIBUTE | SELECT EQ.TYPE | SELECT VALUE

| ATTRIBUTE | EQULITY TYPE | VALUE |
|---|---|---|
| NEWSGROUP | = | CNN |
| | | |

SYSTEM MESSAGE:

Figure 5.8 Window for Specifying the Order of the Documents to be Presented

## 5.4.5 Procedure for Processing the Request

### 5.4.5.1 Procedure for Bringing the News

1. Select the "PROCESS" option from the first window of Menu-based query generation.
2. "PROCESS" window pops out as shown in Figure 5.9.
3. Select the "BRING THE NEWS ITEM" button.
4. The required documents will be brought to the use.

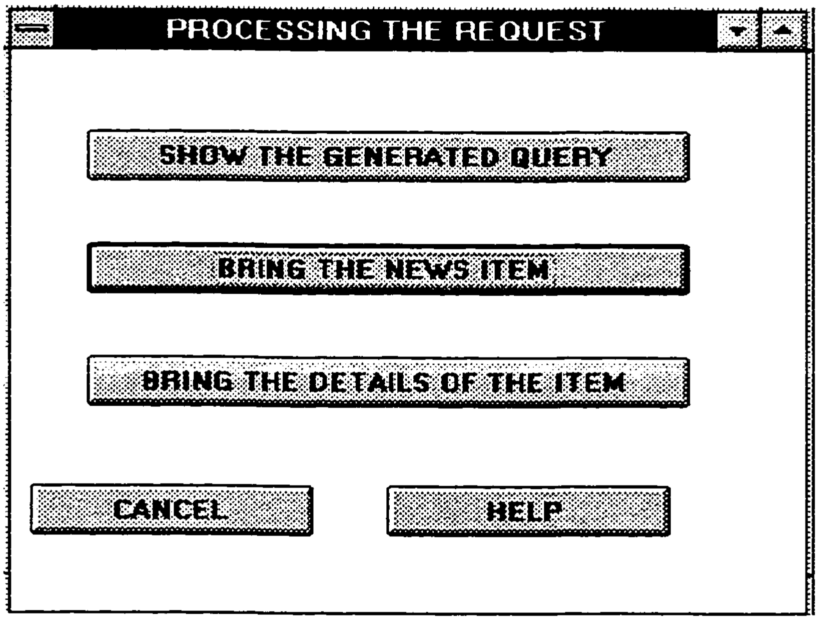
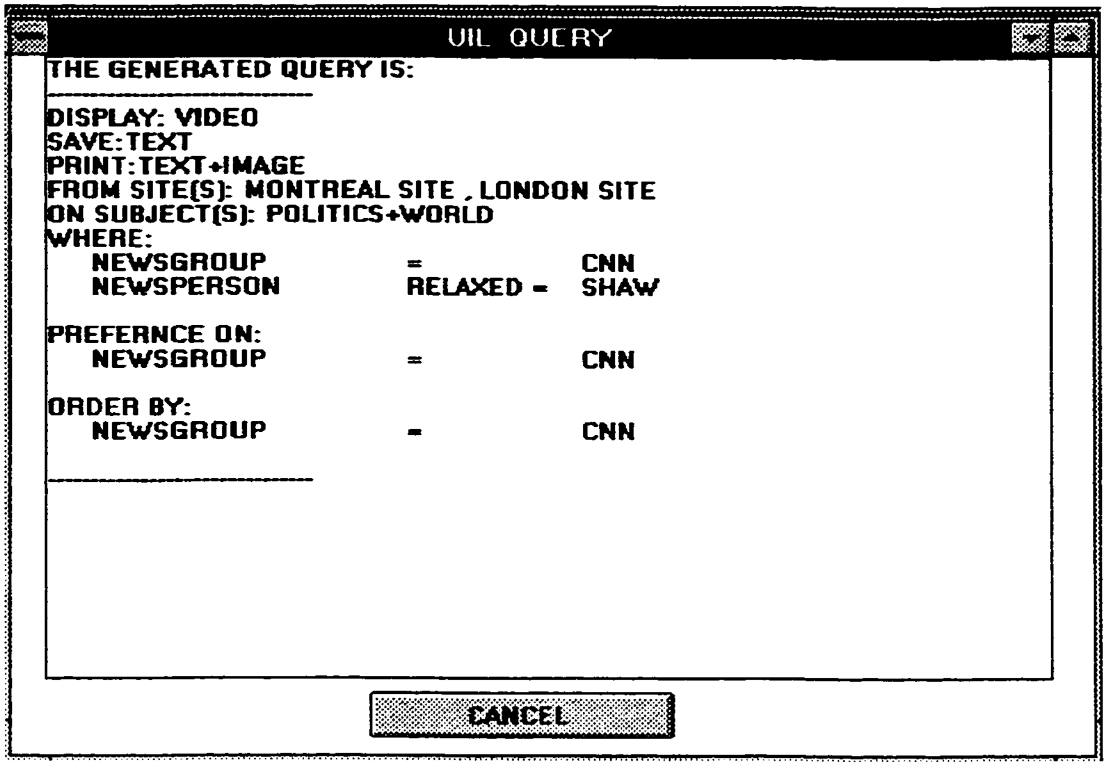


Figure 5.9 Window for Processing Different Options after Query Generation

### 5.4.5.2 Procedure for Viewing the System Generated Query

1. Select the "PROCESS" option from the first window of Menu-based query generation.
2. "PROCESS" window pops out as shown in Figure 5.9.
3. Select the "SHOW GENERATED QUERY" button.
4. The generated query will be displayed in the window shown in Figure 5.10.



```
THE GENERATED QUERY IS:
--------------------------
DISPLAY: VIDEO
SAVE:TEXT
PRINT:TEXT+IMAGE
FROM SITE(S): MONTREAL SITE , LONDON SITE
ON SUBJECT(S): POLITICS+WORLD
WHERE:
    NEWSGROUP           =           CNN
    NEWSPERSON          RELAXED =   SHAW

PREFERNCE ON:
    NEWSGROUP           =           CNN

ORDER BY:
    NEWSGROUP           -           CNN

--------------------------
```

Figure 5.10 The Display of Generated UIL Query

### 5.4.5.3 Procedure for Getting Other Details

1. Select the "PROCESS" option from the first window of Menu-based query generation.

2. "PROCESS" window pops out as shown in Figure 5.9.

3. Select the "NEWS DETAILS" button.

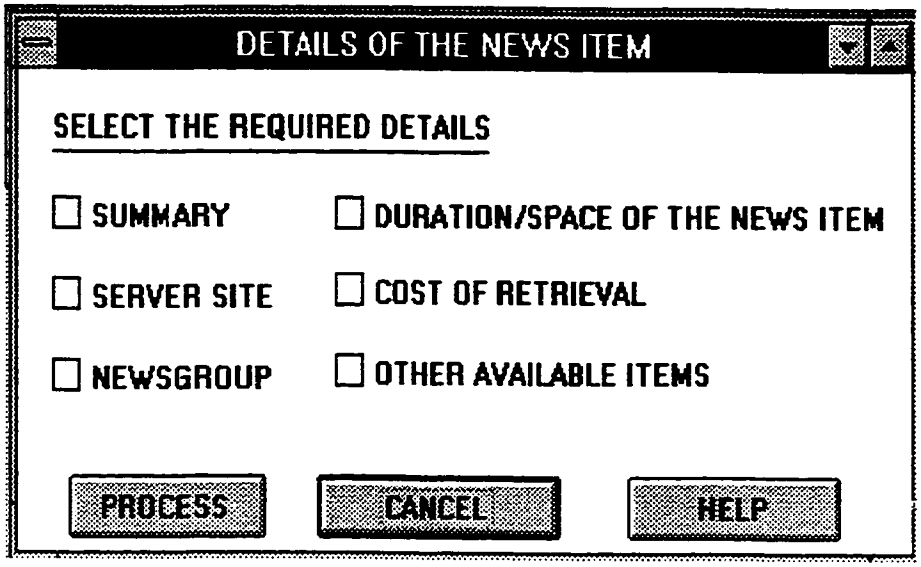
4. A window pops up with the name of the possible attributes (Figure 5.11).

5. Select the required details if they are not already selected.

6. Deselect the other details if they are already selected.

7. Select the "PROCESS" button.

8. The required details will be brought to the screen.



DETAILS OF THE NEWS ITEM

SELECT THE REQUIRED DETAILS

□ SUMMARY

□ SERVER SITE

□ NEWSGROUP

□ DURATION/SPACE OF THE NEWS ITEM

□ COST OF RETRIEVAL

□ OTHER AVAILABLE ITEMS

Figure 5.11 Window for Selecting the Required Details

## 5.4.6 Conventional News Media Presentation

### 5.4.6.1 Procedure for Selecting a TV Presentation

1. Select the " MEDIA NEWS RETRIEVAL" from the opening window (Figure 5.1).

2. A window pops out as shown in Figure 5.12.

3. Select the "TV".

4. A list of TV stations are displayed.

5. Select the required station from the list.

6. Select "PROCESS".

### 5.4.6.2 Procedure for Selecting a Radio Presentation

1. Select the "MEDIA NEWS RETRIEVAL" from the opening window (Figure 5.1).

2. A window pops out as shown in Figure 5.12.

3. Select the "RADIO".

4. A list of Radio stations are displayed.

5. Select the required station from the list.

6. Select "PROCESS".

### 5.4.6.3 Procedure for Selecting a News Paper Presentation

1. Select the "MEDIA NEWS RETRIEVAL" from the opening window (Figure 5.1).

2. A window pops out as shown in Figure 5.12.

3. Select the "NEWS PAPER".

4. A list of News Paper names are displayed.

5. Select the required name from the list.

6. Select "PROCESS".

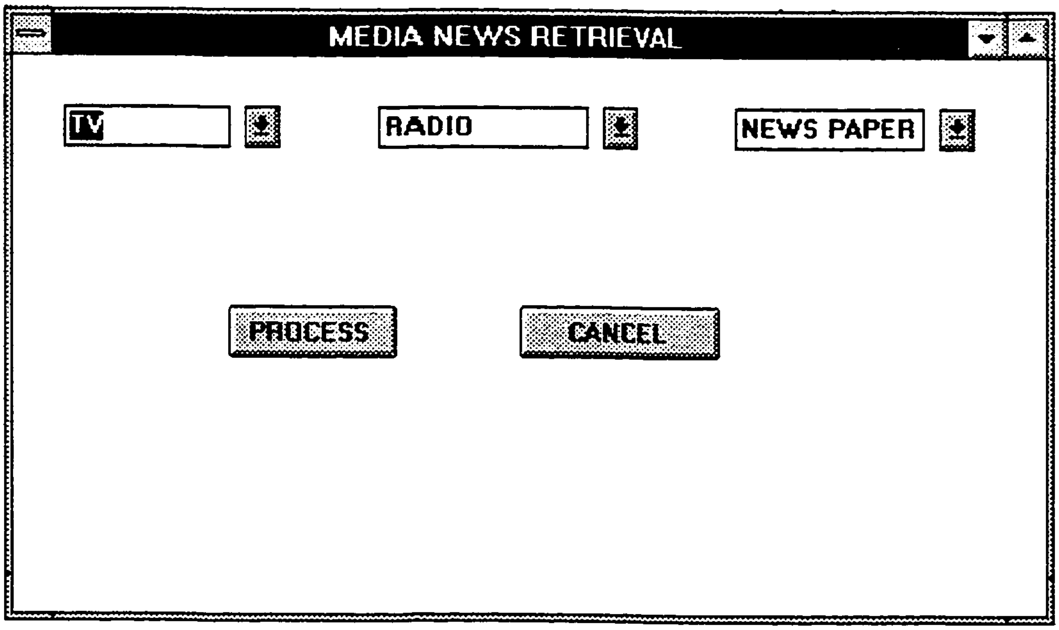
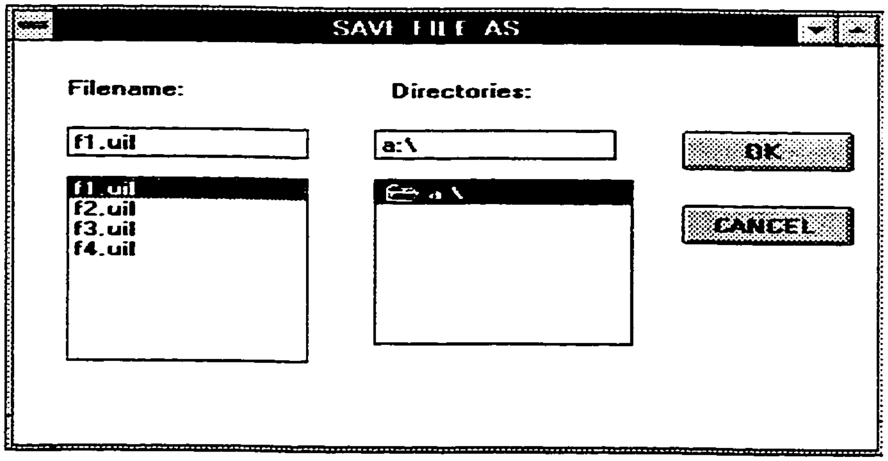


Figure 5.12 Window for Selecting the News through the Conventional Media

## 5.4.7 File Commands

### 5.4.7.1 Procedure for Saving a UIL Query into a File

1. Select the "FILE" option from the window of Menu-based query generation (Figure 5.3).
2. A pull-down menu appears.
3. Select "SAVE" from this menu.
4. "SAVE" window pops out as shown in Figure 5.13.
5. Select the Directory.
6. Select or type in the File name.
7. Press "OK" to save the query into the specified file.



Filename: f1.uil

f1.uil
f2.uil
f3.uil
f4.uil

Directories: a:\

a:\

Figure 5.13 Window for Specifying the Filename for Saving the Query

### 5.4.7.2 Procedure for Opening a UIL Query from a File

1. Select the "FILE" option from the window of Menu-based query generation (Figure 5.3).
2. A pull-down menu appears.
3. Select "OPEN" from this menu.

4. "OPEN" window pops out as shown in Figure 5.14.

5. Select the Directory.

6. Select or type in the File name.

7. Press "OK" to open the query from the specified file.

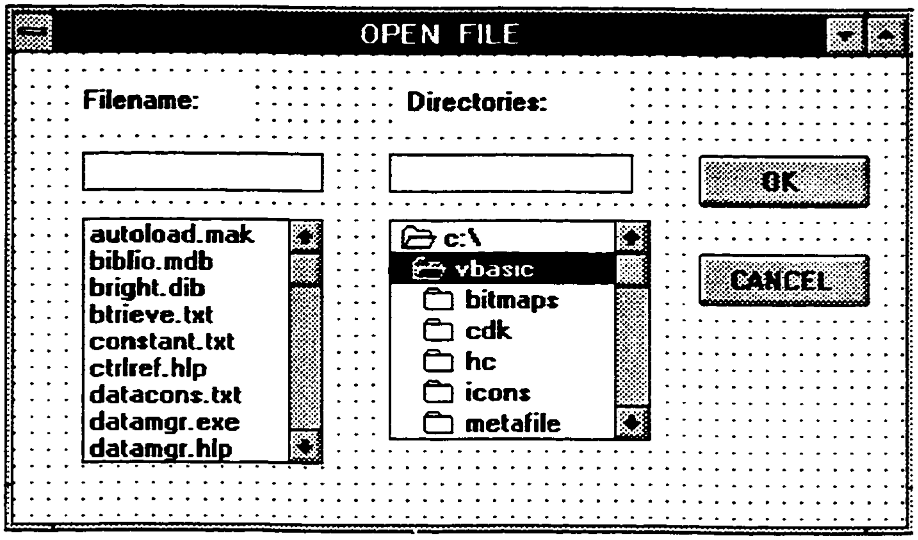


Figure 5.14 Window for Specifying the Filename for Opening the Query

## 5.4.8 UI for the Presentation Phase

In the Electronic News Delivery System, 4 types of media documents are presented: VIDEO, AUDIO, IMAGE, and TEXT. Along with the presentation of each media type, a set of presentation controls are activated and displayed. The use of each of these controls were described in Chapters 3 (Section 3.5) and 4 (Section 4.3). These display controls are shown in Figures 5.15 to 5.20. To invoke any displayed control, the corresponding command button has to be pressed. The processing of these controls are beyond the scope of this work. When two media documents are presented concurrently, the display controls for both these media types are active and displayed to the user.

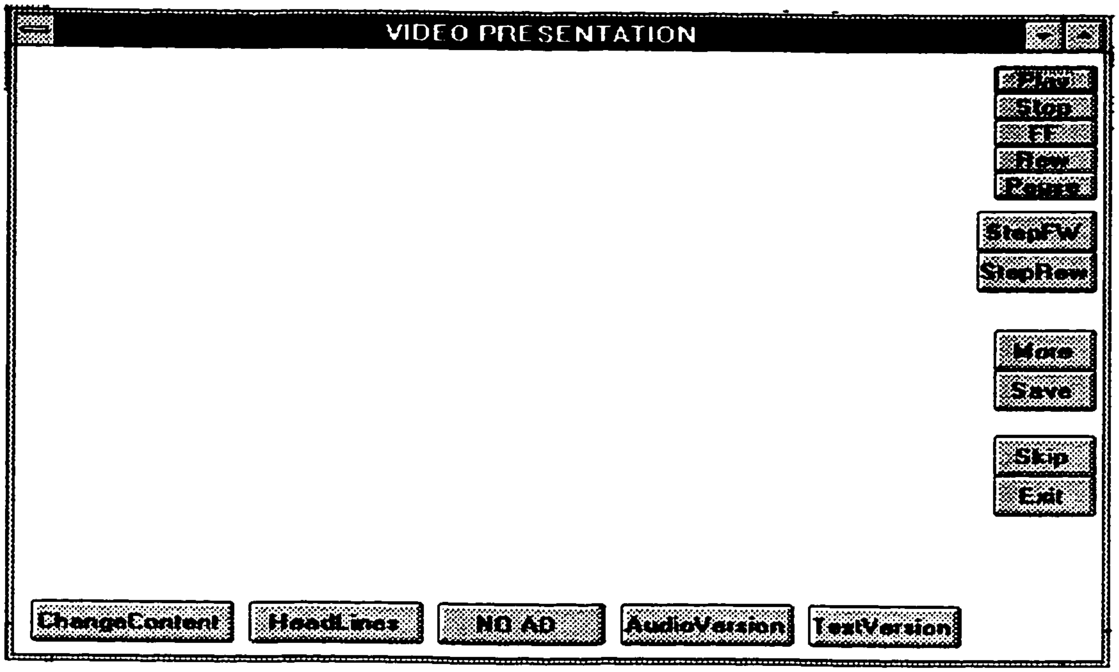


Figure 5.15 Display Controls during Video News Presentation (PND)

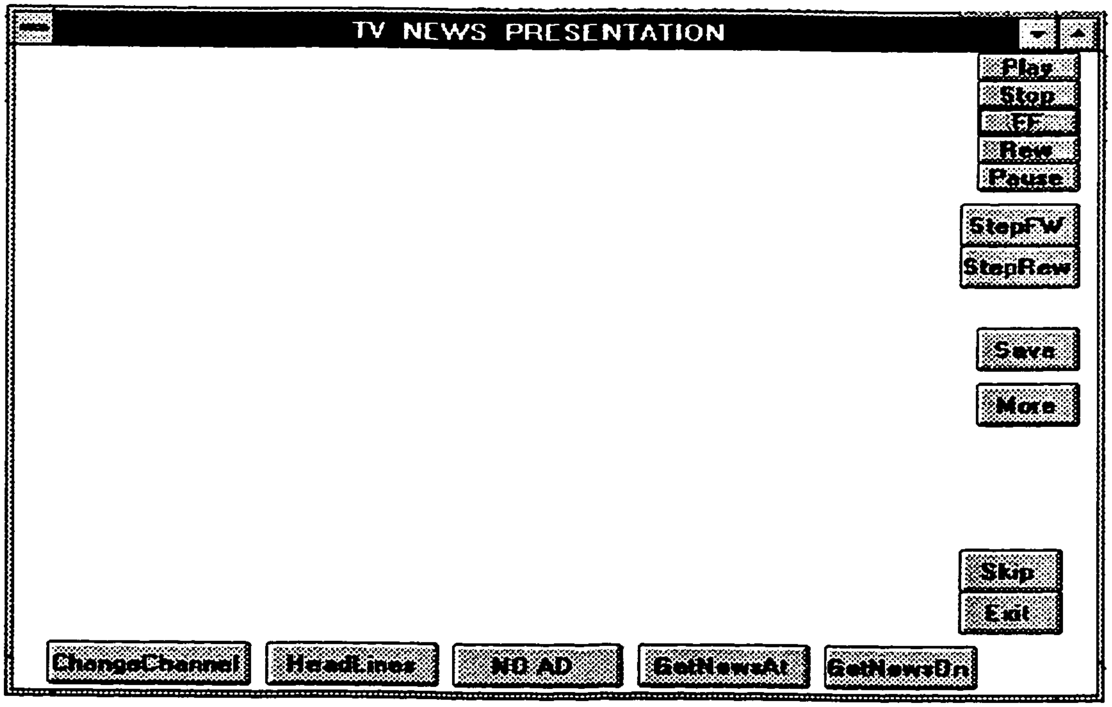


Figure 5.16 Display Controls during TV News Presentation (CND)

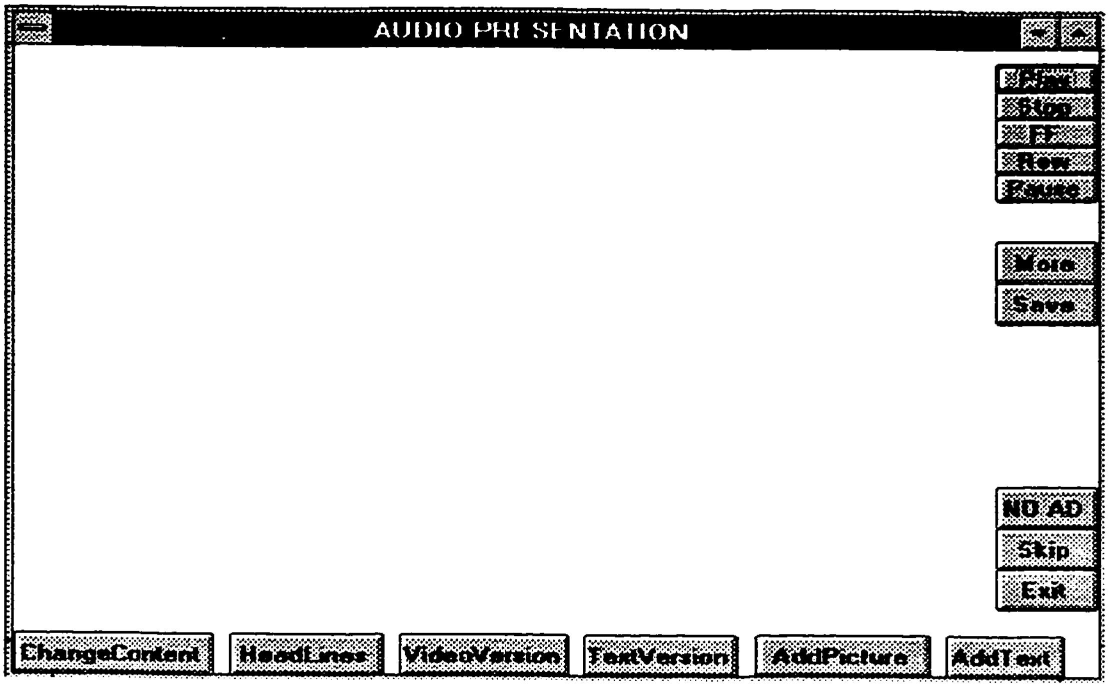


Figure 5.17 Display Controls during Audio News Presentation (PND)

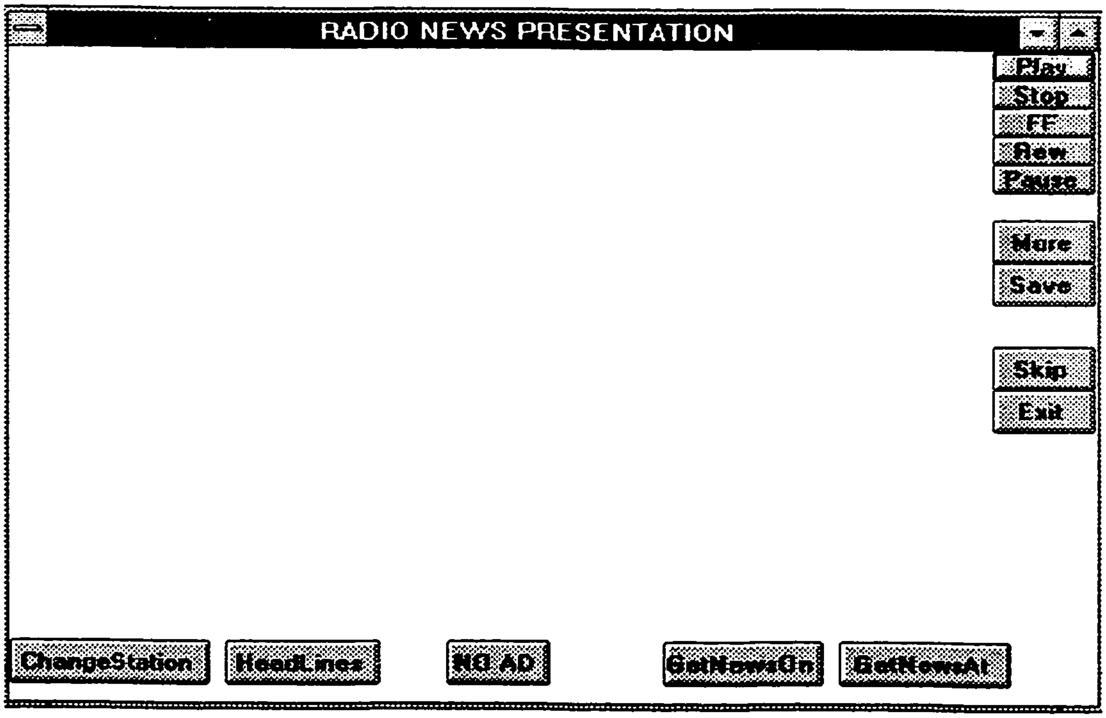


Figure 5.18 Display Controls during Radio News Presentation (CND)

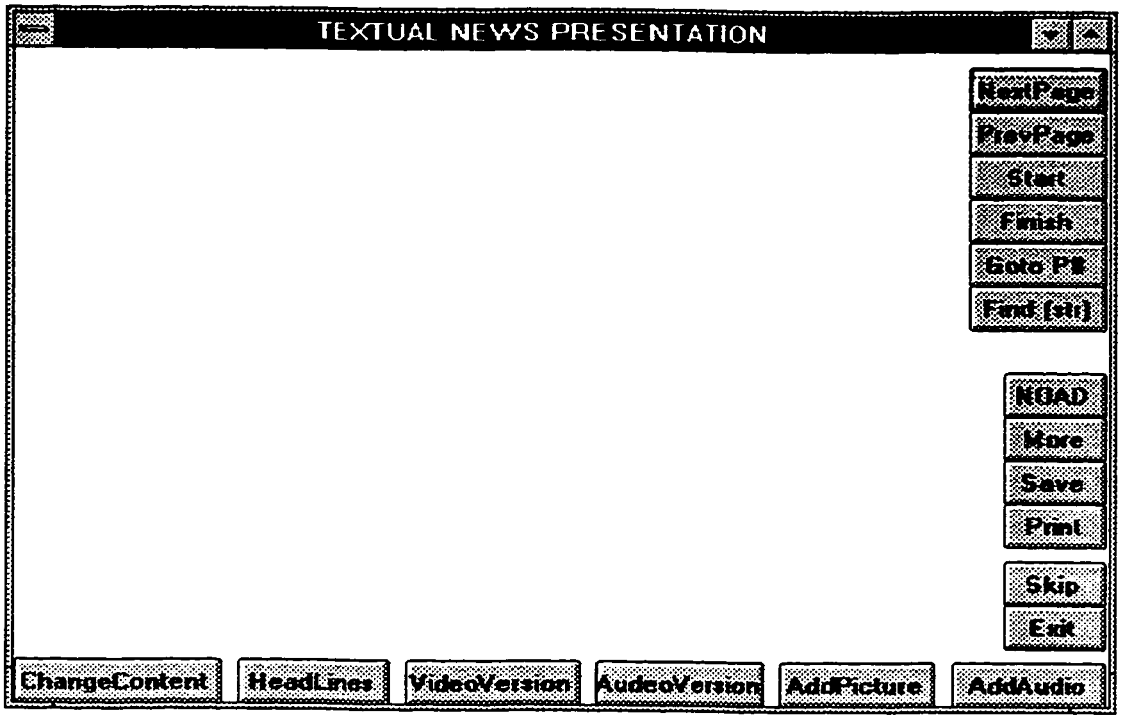


Figure 5.19 Display Controls during Textual News Presentation (PND)

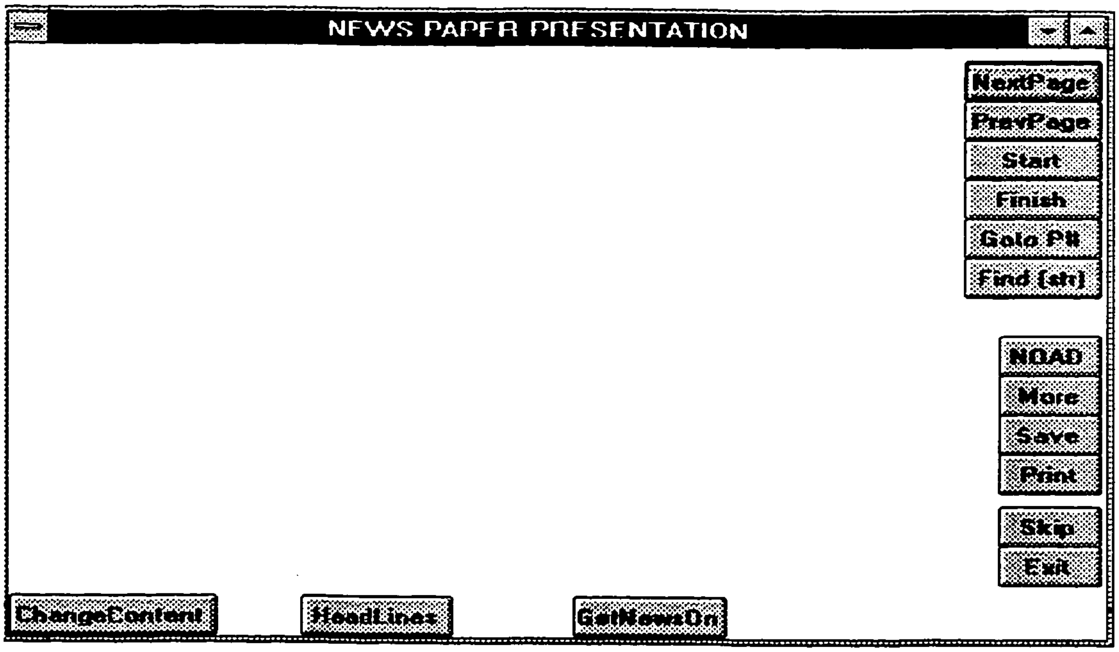


Figure 5.20 Display Controls during News Paper Presentation (CND)

## 5.5 Speech Input for UIL

Speech is the most natural communication modality of humans. In human-computer communication speech can offer a natural, fast, hands free, eyes free, and location free input medium. Learning to use this modality does not require any practice or special skill for the user. So most computer users would prefer speech over other inputs, if it is equally effective. The advantages and disadvantages of the speech input are explained in the following subsections.

### 5.5.1 Advantages of Speech Input

This section concentrates on two common claims of speech input in user-computer communication: (i) speech is a more efficient and readily available response channel, and (ii) speech provides an additional response channel, thereby extending the user's overall efficiency. Each of these advantages is elaborated in the following subsections.

#### 5.5.1.1 Additional Response Channel

Speech provides an additional response channel for the user. Adding one more channel, provides the user a chance to improve the ability to act better. People can do more than one thing at a time. People are generally more efficient if the multiple tasks are spread across different perception and response channels. One's ability to concentrate on more than one task at a time, seems to be expanded when the tasks are presented in separate perceptual channels and people respond to them across different response channels.

Another theory states that a person's information processing capacity is not a single reservoir allocated in graded amounts to simultaneously performed tasks. Rather processing capacity seems to be drawn from multiple resources [Mart89]. When different tasks tap different resources, as manual movements and speech are thought to do, then much of the processing can go on in parallel, not interfering with each other [Mart89]. When the tasks tap the same resource, interference between the tasks occurs and processing slows. Thus the user may be more efficient in perceiving, processing and responding to multiple channels. The additional response channel provided by the speech enhances the user's attentional capacity (ability to concentrate in more than one task) [Mart89].

### 5.5.1.2 Faster Input Medium

Comparison of spoken and written expression [Mart89] indicate that spoken expression tends to generate more material, such as words, phrases and ideas. So if we define the efficiency of the input by the amount of user responses, then the spoken responses are more efficient than typed responses.

This is beneficial for tasks involving frequent and short interactions between the user and the system. The benefits are less noticeable in tasks involving long transactions and long periods of thinking time between transaction. The main reason which makes speech a faster input medium is that users eyes and hands are free. With speech input, the users direction of looking can be changed usefully. Without speech input capability, the user's may spend more time looking at the keyboard to position their hands or locate keys. With speech, this time can be reduced drastically and users will be looking more of the time at informative areas such as display screen and reference material. Since specch provides an additional user response modality, it is more useful than other input media. Also, speech input lessens the user's workload in the sense that the users do not have to glance down at the keyboard as frequently, as otherwise would be needed.

The comparison of speech with the common input media are given below.

**Speech Vs. Mouse :** There are three reason for speech commands to be faster than mouse commands [Karl93]. The First reason is that when a user wishes to execute a menu command and her hand is not on the mouse, some time would be wasted to locate the mouse, put the hand on it, locate the mouse pointer on the screen, move the mouse to point to the desired menu and then traverse the menu list to reach the desired command. This takes considerably more time than merely speaking the desired command. Second reason, using the mouse to activate the menu commands required that the user remove her eyes from the work and find where had left off, after doing it. With the speech input the user had the advantage of allowing them to keep their eyes concentrated on their work. Third, using the hands to activate the mouse the user has to move her hand from the keyboard and then back to the keyboard. But with speech, this delay can be avoided.

**Speech input vs. Single keypress** : Speech is somewhat more efficient than single keypresses [Mart89]. In simple keypresses, time has to be spent to move the hand off the mouse, remember the correct abbreviation, locate the correct key on the keyboard and depress it. For speech input, this time is negligible.

**Speech input vs. Full word typed commands**: Speech is found to be considerably faster than typed, full-word input [Mart89]. In typed response, time is spent in two ways: for planning and command operations. Time for planning involves the time taken to move the hand off the mouse and locate the correct key on the keyboard. This planning time is less for speech input. The next component is the command operation. Time for command operation is the time taken between start and finish of the response. For typed command, this is much longer compared to the negligible time taken for the spoken command.

### 5.5.2 Disadvantages of Speech

Martin [Mart89] states in his research review that the research on comparing the efficiency of speech with other input modal is often contradictory and ambiguous. Even though both applied research and more basic psychological research support the claim that speech input will enhance user performance, some researches conducted in applied settings has not allowed verification of the claim that speech is more efficient. The efficiency of speech input often depends upon the quality of the speech recognition system and the environment. So in many cases speech does not yield faster or more efficient entry. Damper *et.al.* [Damp95] stated in a study that speech is enormously more error-prone than keying.

Speech is probably more efficient than other forms of input in tasks involving short transactions and high interaction with the computer, and less efficient for tasks that require thinking time or long transaction [Damp95].

### 5.5.3 Speech Commands in UIL

We assume that users will use Electronic News Delivery System very frequently to receive the news. The advantages and usefulness of UIL for news delivery are explained in the previous chapters. The interface for the news delivery system is implemented in Version 1-0, with mouse clicks and keypresses and the speech is intended for a later version. The main disadvantage of these input media is that user might have a feeling of "working" with computers while they receive their news. This feeling is worse for the users who just browse through the different documents (tasks involving frequent and short interaction) . They have to concentrate on their input actions very frequently and such an action may increase the cognitive load on the users. This difficulty can be reduced by using speech commands.

UIL provides presentation controls which can be activated when the news is being delivered. When the user is settled in reading or waiting for the news, she wants to forget about the computer and enjoy the news with her hands free and eyes concentrated on the news. If they want to use any of the control option during this time, speech can be the best input media for invoking these controls. With speech the user does not have to do any manual operation, other than uttering the word.

During the Selection Phase, the news is not being delivered and the user's eyes are free. For this phase of UIL, speech may not have much advantages, other than user's hands are free. To use the speech command, the user has to remember the correct vocabulary. The commands are to be uttered without ambiguity, which is not an easy task always. Moreover to be safe, she has to confirm that the system understood her clearly before bringing the news. Otherwise it can be waste of time and money. These tasks increase the cognitive load for the user. Speech alone cannot be a better medium for the selection phase of UIL. But Speech along with the other media can help the user to use the system more efficiently during this phase.

To conclude, we can say that speech alone may not be the best input media for the UIL. But along with other input modalities, speech can enhance the user-friendliness.

## 5.6 Summary

The User interface for the UIL in Electronic News Delivery system is implemented using the software tool "Visual Basic". The input for this interface is mouse click and key-presses. The interface converts these inputs into the appropriate UIL query. The processing of the generated UIL query is beyond the scope of this work and will be carried out in the future work.

The proposed GUI is user friendly for many reasons. The expert user as well as the naive user can "enjoy" this interface equally because of the two types of query creation (Menu-based and Manual). In both methods, the system would assists the user by reporting the committed errors and the required remedies, etc. At any stage the user can ask for help or see the Query status. By viewing the query status, the user will get the WYSIWYG effect. Also the naive user can very easily become an expert and transfer from the menu driven creation to manual creation of query. Since this UI provides the mechanism to switch between these two types at any stage, the users between the naive and expert status also can use this very efficiently by switching in these two modes whenever necessary. This UI provides options to save and open the files. So, the user can save the generated query and open it at a later stage and use it for different purpose.

# 6

# *Conclusion*

The work reported in this major report was necessitated by the limitations of the conventional news media like News Papers, TV, Radio, and Internet in delivering news to meet the personal requirements of an individual. In this study, an Electronic News Delivery System (ENDS), which delivers the multimedia news documents that are personalized to satisfy each customer, was proposed. The ENDS spreads over one client site and many server sites. In theory, the multiple server sites can be located all around the world. The system (ENDS) considers two type of news delivery: *Conventional News Delivery (CND)* and *Personalized News Delivery (PND)*. The latter is aimed at individual users whereas the former is intended for a "coherent" group of users. In the CND, users can receive the news in the same manner as they now receive in the conventional media like news paper for example, so that they can stick to their routine of news reception, but benefit from the electronic delivery. As opposed to this, the PND will customize the news delivery to the individual user.

In the ENDS, there are three parts which are geographically separated and they need to communicate between each other: User, Client, and Server. Since the system is highly interactive, the communication language plays an important role in its success. As the first step in finding a suitable communication language for the ENDS, the following major requirements were identified.

1. The system should be able to deliver any type of media individually or combined together.
2. The system should be able to deliver the news through one or more of the following modes: On-line Display, Downloading, or Hard-copying.
3. The user should be able to specify a future time in advance for the news delivery.
4. The system should be able to restrict the search to certain subject(s) or site(s).
5. There should be different types of matching available for qualifying the news documents.
6. The system should be able to accept the requirements with the vague specification of the attribute.
7. There should be some co-operation available between the system and the user to select the "neighbourhood" answer if the exact answer is not available.
8. There should be some way of further selecting the preferred documents for presentation from the retrieved set, if the number of the retrieved documents is too large.
9. The user should be able to specify the order of presentation from the retrieved set.
10. The system should provide presentation time control when the news is delivered.

The "new generation" database query languages were studied to see whether any of them met the requirements of the ENDS. But, none of them was found to be completely suitable for the news domain to provide the personalised delivery proposed in this work. Most of the languages were generic and did not have any specific features for any particular domain. There were very few domain specific languages in the literature, but they were not developed for news or any similar domain. Therefore the major objective of this work was to develop a suitable communication language for the proposed ENDS.

A new language called the User Interaction Language(UIL) was proposed in this work as the communication media in ENDS. This language can be used for the communication between the client and server and also with the user. UIL is a Domain Specific language which is designed to meet the requirements of news domain. There are three unique features which make UIL suitable for news domain.

1. UIL is highly interactive and allows the user interaction at any point of time even while the news presentation is going on. There are many user controls available during the presentation to adapt the presentation to the user's choice.
2. UIL applies concurrency during presentation, searching, and news reception with the use of a concurrent operator "+". Two media documents can be presented concurrently when relevant, and a document can be searched under two subjects to make sure that it is categorized under both of these subjects, and three modes of reception (on-line presentation, downloading, and printing) can be activated concurrently which can be useful for balancing the cost with quality of service. The sequential processing is demanded by the use of the sequential operator ",".
3. UIL allows different weights to be assigned to the qualifying conditions by specifying them as "relaxed" or "preferred". These different weights on the conditions will be useful when the system finds the number of documents is either too large or too small. In those cases, these conditions can be used to increase the Recall or the Precision.

The UIL operates in two phases: Selection Phase and Presentation Phase. During the Selection Phase, the initial query is specified. UIL query consists of four types of clauses: Target Clause, Range Clause, Qualification Clause, and Presentation Clause. Target clause deals with the type of media and reception mode. Range clause restricts the region to be searched in terms of subject or geographical site. The search criteria, in terms of boolean condition(s), are specified under the qualification clause. If there are any changes required in the normal presentation, it should be mentioned in the presentation clause. The second phase (Presentation Phase) is unique to UIL and it starts when the Selection Phase is over. It consists of the presentation of the news documents and the processing of a number of controls (interrupts) which are active with the presentation. By invoking these controls, the presentation can be altered to meet the user's choice.

As part of this report, the complete syntax and semantics of the UIL for both these phases were developed. All the UIL constructs have intuitive meaning in the News Domain so that, they are easy to learn and use. Unlike other Query Languages, UIL does not insist on the complete and clear query specifications. The incomplete queries are expressed through different ways: (i) Preference, (ii) Relaxation, (iii) Matching with Unknown Attribute, (iv) without Search Range, etc. These incompleteness are handled by the system in many ways: (a) Dialogue with the user, (b) Knowledge stored at the server sites, (c) Previous history of the news access, (d) Assistance of a personalised agent and (e) User profile stored at the client site. Also, the UIL constructs can be mapped to simple speech commands and the SLS can be used as the interface to help the users when they are working. For multimedia news access, UIL provides many different presentation time controls for the user. The media conversion is also possible during the presentation

The design of UIL met all the requirements which were listed for the ENDS. As a constructive proof of completeness of the UIL, a number of sample queries were written for CND and PND services for the ENDS. The queries were selected in such a way that each and every initial requirement was taken care in atleast one of the query. These set of queries verified that, UIL can be used to access any of the conventional services as well as content (subject, or keywords) based news. The ***relaxation*** and ***preference*** aspects of a user's query can be used to increase the Recall and/or the Precision.

A Graphical User Interface(GUI) (Version 1.0) was designed and developed to create the UIL query in the News Delivery System. The only inputs used in this interface are key presses and mouse clicks. The User Interface System interprets these inputs and develops the appropriate query in UIL. Since the ENDS expects the user from all social backgrounds, the user interface was developed for a wide range of user population. During the Selection Phase, user can select all the necessary requirements of the different query clauses by going through different menus, sub-windows, and command buttons. There are options for manual query creation for the expert user and menu based query creation for the naive user. The user can switch between these options whenever needed. This will help the users who are neither experts nor naive. The presentation controls of the UIL were designed in the User Interface using command buttons. These controls are different for each news media type and each conventional news media. All the required controls are activated and presented in the interface along with the delivery of the corresponding news.

As a future expansion, a number of tasks can be added to this work:

1. An extensive survey can be conducted to see the pattern and/or routine of people receiving the news. This can help to refine the requirements for UIL.
2. Using these added requirements, UIL design can be refined further.
3. The User Interface can be improved by adding the speech as the input. Speech can enhance the user performance , especially during the presentation phase.
4. Mobility of users can be taken into account in the news delivery.
5. Intelligent Agents based support can be added to the User interface in assisting the user.

# *Bibliography*

**[Bach93]:** Bach, J.R.,, Paul, S., and Jain, R. (1993), "A Visual Information Management System for the Interactive Retrieval of Faces", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 4, pp. 619-628.

**[Bert92]:** Bertino, E., Negri, M., Pelagatti, G., and Sbattella, L. (1992), "Object-Oriented Query Languages: The Notion and the Issues", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, No. 3, pp. 223-237.

**[Berz93]:** Berztiss, A. (1993), "The Query Language Vizla", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 5, pp. 813-825.

**[Card93]:** Cardenas, A.F., Ieong, I.T, Taira, R.K., Barker, R., and Breant, C.M. (1993), "The Knowledge-Based Object-Oriented PICQUERY+ Language", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 4, pp. 644-656.

**[Cha91]:** Cha, S.K (1991), "Kaleidoscope: A Cooperative Menu-Guided Query Interface (SQL Version)", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 3, No. 1, pp. 42-47.

**[Chan94]:** Chan, D.K.C., Trinder, P.W., and Welland, R.C. (1994), "Evaluating Object-Oriented Query Languages", *The Computer Journal*, Vol. 37, No.10, pp. 858-871.

**[Chu94]:** Chu, W.W., and Chen, Q. (1994), "A Structured Approach for Cooperative Query Answering", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 5, pp. 738-748.

**[Damp95]:** Damper, R.I., and Wood, D. (1995), "Speech Versus Keying in Command and Control Applications", *International Journal of Human-Computer Studies*, 42, pp. 289-305.

**[Foul97]:** Fouloustos, C., et.al. (1997), Selected Papers in the Area of "Multimedia Indexing". Available at the site *http://www.cs.umd.edu/~christos/selpapers.html.*

**[Feld93]:** Feldman, P., Jennings, R., Seymour, B., Eidson, B., Palmer, P., Gillmor, S., and Pesso, J. (1993), *Using Visual Basic 3*, Que Corporation.

**[Hoff91]:** Hoffert, E.M., and Gretsch, G. (1991), "The Digital News System at EDUCOM: A Convergence Of Interactive Computing, Newspapers, Television and High-Speed Networks", *Communications of the ACM*, Vol. 34, No. 4, pp. 13-19.

**[Jark85]:** Jarke, M., and Vassiliou, Y. (1985), "A Framework for Choosing a Database Query Language", *Computing Surveys*, Vol.17, No.3, pp. 313-340.

**[Karl93]:** Karl, L.R., Pettey, M., and Shneiderman, B. (1993), "Speech Versus Mouse Commands for Word Processing: An Emperical Evaluation", *International Journal of Man-Machine Studies*, 39, pp. 667-687.

**[Kier88]:** Kiers, D.E. (1988), *Handbook of Human-Computer Interaction*, Elsevier Science Publishers.

**[Kife92]:** Kifer, M., Kim, W., and Sagiv, Y. (1992), "Querying Object-Oriented Databases", *Proc. ACM SIGMOD*, pp. 393-402.

**[Kim89]:** Kim, W. (1989), "A Model of Queries for Object-Oriented Databases", *Proc. International Conference on Very Large Databases*, Amsterdam, August.

**[Kort91]:** Korth, H.F., and Silberschatz, A. (1991), *Database System Concepts*, McGraw-Hill, Inc.

**[Lacr87]:** Lacroix, M., and Lavency, P. (1987), "Preferences: Putting More Knowledge into Queries", *Proceedings of the 13th VLDB Conference*, Brighton, pp. 217-225.

**[Laks93]:** Lakshmanan, L.V.S., Sadri, F., and Subramanian, I.N. (1993), "On the Logical Foundations of Schema Integration and Evolution in Hetrogeneous Database Systems", *Proc. 3rd International Conference on Deductive and Object-Oriented Database*, Springer-Verlag, LNCS-760, December.

**[Laks96]:** Lakshmanan, L.V.S., Sadri, F., and Subramanian, I.N. (1996), "SchemaSQL: A Language for Interoperability in Multiple Relational Databases", *IEEE International Conference on VLD*, Bombay, India, September.

**[Laks97]:** Lakshmanan, L.V.S., Sadri, F., and Subramanian, I.N. (1997), "Logic and Algebraic Languages for Interoperability in Multidatabase Systems", To appear in the *Journal of Logic Programming*, October.

**[Macl91]:** Macleod, I.A. (1991), "A Query Language for Retrieving Information from Hierarchic Text Structures", *The Computer Journal*, Vol. 34, No.3, pp. 254-264.

**[Marc97]:** Marcus, s., and Subrahmanian, V.S. (1997), "Multimedia Database Systems", Available at the site *http://www.cs.umd.edu//projects/hermes/publications/abstracts/mds.html.*

**[Mart89]:** Martin, G.L. (1989), "The Utility of Speech Input in User-Computer Interfaces", *International Journal of Man-Machine Studies*, 30, pp. 355-375.

**[Mayh92]:** Meyhew, D.J. (1992), *Principles and Guidelines in Software User Interface Design*, Prentice Hall Inc.

**[Moha93]:** Mohan, L., and Kashyap, R.L. (1993), "A Visual Query Language for Graphical Interaction with Schema-Intensive Databases", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 5, pp. 843-857.

**[O'Doch91]:** O'Docherty, M.H., and Daskalakis, C.N. (1991), "Multimedia Information Systems - The Management and Semantic Retrieval of all Electronic Data Types", *The Computer Journal*, Vol. 34, No. 3, pp. 225-238.

**[Ozso93]:** Ozsoyoglu, G., and Wang, H. (1993), "Example-Based Graphical Database Query Languages", *IEEE Computer*, May, pp. 25-38.

**[Rous88]:** Roussopoulos, N., Faloutsos, C., and Sellis, T. (1988), "An Efficient Pictorial Database System for PSQL", *IEEE Transactions on Software Engineering*, Vol. 14, No. 5, pp. 639-650.

**[Rowe87]:** Rowe, L.A., and Stonebraker, M.R. (1987), "The POSTGRES Data Model", *"Proc. Of the 13th VLDB Conference*, Brighton, pp. 83-95.

**[Saxt90]:** Saxton, L.V., and Raghavan, V.V. (1990), "Design of an Integrated Information Retrieval/Database Management System", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 2, pp. 210-219.

**[Sutc91]:** Sutcliffe, A.G., and Wang, I. (1991), "Integrating Human Computer Interaction With Jackson System Development", *The Computer Journal*, Vol. 34, No.2, pp. 132-142.

**[Sutt95]:** Sutton, D.R., and King, P.J.H. (1995), "Incomplete Information and the Functional Data Model", *The Computer Journal*, Vol. 38, No.1, pp. 31-42.

**[Urba94]:** Urban, S.D., Lai, C.C., and Saxena, S. (1994), "The Design and Translation of ORL: An Object Retrieval Language", *J. Systems Software*, 24, pp. 187-206.

**[Urba95]:** Urban, S.D., and Wang, A.M. (1995), "The Design of a Constraint/Rule Language for an Object-Oriented Data Model", *J. Systems Software*, 28, pp. 203-224.

**[Yosh94]:** Yoshitaka, A., Kishida, S., Hirakawa, M., and Ichikawa, T. (1994), "Knowledge Assisted Content-based Retrieval for Multimedia Databases", *IEEE Multimedia*, Winter, pp. 12-20.

**[Wald90]:** Wald, J.A., and Sorenson, P.G. (1990), "Explaining Ambiguity in a Formal Query language", *ACM Transactions on Database Systems*, Vol. 15, No. 2, pp. 125-161.