A Graphical User Interface For Boolean Query Specification

STEVE JONES, SHONA MCINNES, MARK S STAVELEY

Department of Computer Science

University of Waikato, Hamilton, New Zealand

Telephone: +64 7 838 4021
Fax: +64 7 838 4155

Email: stevej@cs.waikato.ac.nz

Abstract

On-line information repositories commonly provide keyword search facilities through textual query languages based on Boolean logic. However, there is evidence to suggest that the syntactic demands of such languages can lead to user errors and adversely affect the time that it takes users to form queries. Users also face difficulties because of the conflict in semantics between AND and OR when used in Boolean logic and English language. Analysis of usage logs for the New Zealand Digital Library (NZDL) show that few Boolean queries contain more than three terms, use of the intersection operator dominates and that query refinement is common. We suggest that graphical query languages, in particular Venn-like diagrams, can alleviate the problems that users experience when forming Boolean expressions with textual languages. A study of the utility of Venn diagrams for query specification indicates that with little or no training users can interpret and form Venn-like diagrams in a consistent manner which accurately correspond to Boolean expressions. We describe VQuery, a Venn-diagram based user interface to the New Zealand Digital Library (NZDL). In a study which compared VQuery with a standard textual Boolean interface, users took significantly longer to form queries and produced more erroneous queries when using VQuery. We discuss the implications of these results and suggest directions for future work.

Keywords

searching, query interface, Boolean logic, query specification

1 Introduction

Digital libraries and other common on-line information repositories must provide effective access to their contents for a wide variety of users. Two modes of access which are commonly supported are browsing and searching. When browsing, users traverse information structures to identify required information. Their traversal may be supported by organisational schemas such as subject classifications, author lists and so on. When searching, users specify terms of interest, and information matching those terms is returned by an indexing and retrieval mechanism. World-Wide Web [1] based Internet search engines and some digital libraries (such as the New Zealand Digital Library, [17]) are examples of systems which provide both keyword searching and content browsing.

In this paper we focus on user interfaces for searching, particularly those for Boolean query specification. Textual query languages which combine user defined keywords and Boolean operands in a strictly defined syntax are commonly provided as a user interface to on-line information sources. For example, it is the default interaction technique provided by many Internet search engines and on-line public access catalogues (OPACs). However, it has been shown that difficulties in dealing with Boolean logic are common, particularly when a restricted syntax is used [2, 9, 5, 7]. The consequences are that significant numbers of erroneous queries are created. Beyond syntactic demands, the conflict between the meaning of operators in Boolean logic and English language poses problems. AND tends to be inclusive in English but is exclusive in Boolean logic; OR tends to be exclusive in English but is inclusive in Boolean logic. Alternative syntax is sometimes used to overcome this. The union operator is commonly represented with |, +, or ∪; intersection by & or ∩; negation by ! or –. The lack of consistency of use of these operators across systems and their lack of direct relationship to their meaning creates further difficulties for users.

We believe that diagrammatic techniques for query specification have two key advantages over textual query languages: they are less syntactically demanding and overcome the English language and Boolean operator ambiguity. We suggest that a particularly effective graphical technique to use is that of Venn diagrams which illustrate sets and their relationships.

In the following section we describe related work. It provides evidence to support the use of Venn diagrams in query interfaces. We go on to describe the context in which our work is being undertaken – the New Zealand Digital Library (NZDL) and report on analysis of

NZDL usage which can guide the design of a Venn diagram based query interface. In Section 4 we describe a user study that we carried out to identify how users might interpret and form Venn diagrams which correspond to Boolean query expressions. Our usage analysis and user study has directed the design of VQuery, a graphical user interface for query specification using Venn diagrams, and we describe and discuss this system in Section 5. In Section 6 we report on an evaluation of VQuery and discuss implications of the results. To conclude we detail how this work and VQuery in particular, may be refined and extended.

2 Related Work

The idea that Venn diagrams may be an effective medium to help users specify Boolean expressions is not new. Thomas [13] studied both interpretation and generation of Venn diagrams, partly with a view to guiding the design of query systems. In this study subjects used a wide variety of expressions to represent their interpretations and no sensible generalisation of the form of expressions could be made. However, there was commonality of meaning, most strongly with disjoint sets (set union) and set equivalence. The meaning of set intersection was much more varied across subjects. The study has some weaknesses, in that a small sample size was used, and the given diagrams had no clear indication of the particular subset that subjects were to identify. In a second study subjects were asked to draw Venn diagrams corresponding to given English language statements after brief pre-task training. Again, a wide range of individual differences was evident.

Michard [10] describes GQL (Graphical Query Language). In GQL users pointed and clicked on buttons representing a numeric keypad and relational operators to define an expression, which in turn was represented as a circle in a Venn diagram. Users could create a maximum of three circles and select areas of the resulting Venn diagram to indicate a Boolean expression. GQL was compared to TEST, a textual Boolean query language. In a comparative study subjects (who were given substantial training) made almost four times as many errors when using TEST than when using GQL. Just under half of the errors with TEST were due to the use of incorrect syntax, and a fifth due to the use of an incorrect Boolean operator. Syntactic errors were not possible in GQL, and half the number of incorrect operators were used than were with TEST. Complex Boolean expressions were more accurately represented in GQL than TEST. With respect to correct Boolean expressions, Michard found little variation in those formed with TEST, but frequent variation in those formed with GQL.

Katzeff [9] considered Venn diagrams in the context of users' mental models of information structures. Subjects in a study were given descriptions of a database and operations on its contents which adopted one of four models (no model, tables, shallow set explanation and deep set explanation). The set models were explained using Venn diagrams. The results show that the set models were more effective when subjects had to form complex queries requiring problem solving but did not differ from other modes for simple queries that matched those given in the descriptions.

Halpin [6] proposed the use of Venn diagrams as an alternative interface to SQL query specification. Although he did not carry out a usage study he convincingly argues that Venn diagrams are highly expressive, can represent a broad range of SQL queries, 'give a clear picture of the meaning of the query' and 'provide a simple means of clarifying what set comparisons mean'.

An important aspect of the use of Venn diagrams is reported by Willie and Bruza [16]. A usage study determined that *set assembly* and *set refinement* techniques were equally common when subjects were able to form their own diagrams (as opposed to indicating a portion of a given diagram). In set assembly, intersections between sets are created by overlapping circles. In set refinement circles are wholly contained within other circles to indicate intersections. In this context refinement does not imply refinement of the query, rather a particular way of arranging sets. Davies and Willie [3] carried out a comparative study of a Venn diagram user interface and a simple Query By Example [18] tool (QBE). They found that use of the Venn diagram query tool resulted in fewer errors, substantially faster specification of queries, and more positive user feedback. A key benefit was the relief from specification of queries using complex syntax as required by the QBE tool.

An aspect of a study by Hertzum and Frøkjær [7] was the comparison of Boolean retrieval using a textual query language and Venn diagrams. The Venn diagram interface was significantly faster and produced significantly fewer errors than the textual query language. Twice the number of operational errors were produced using the textual language than with the Venn interface.

These studies indicate that a Venn diagram based user interface shows promise in more effectively supporting query specification than the standard textual query languages that are currently used. In particular, the syntactical constraints of a textual language are alleviated, and a reduction in erroneous queries and time taken to form queries is likely. The use of Venn diagrams also appears to alleviate, but not remove, the conflict between the meaning of Boolean AND and OR operators and the meaning of these terms in English.

Insert Figure 1 near here

3 The New Zealand Digital Library

The New Zealand Digital Library (NZDL) is a freely-accessible system on the WWW[1] that provides full-text indexes to collections of documents. It is based in the Computer Science Department at the University of Waikato. Ten collections are publicly available and several more are available only to specific end-user organisations. The freely available collections include bibliographic databases, public domain literary works, information relating to indigenous peoples of the world and computer science technical reports (CSTR). The CSTR collection alone provides a full text index to approximately 46,000 technical reports gathered from over 300 sites world-wide, totalling 34 Gigabytes of source text.

The NZDL supports both ranked and Boolean querying, although it is the user interface to Boolean query specification that we address here. As with most other WWW based information sources users are required to form a combination of terms and operators in a particular syntax to represent their intended query. Intersection, union and negation operations are supported and are represented by &, | and ! respectively. Query terms or phrases are joined using these operators, and components of complex expressions may be grouped using parentheses. Queries may be further refined by the use of options. One option

[1] http://www.nzdl.org

is granularity of the search. For example, multiple terms may be required to appear within the same report, same page or same paragraph. Other options control case-folding, stemming and the maximum number of matching documents to return. It is also possible to weight the terms within an expression, so that they have stronger or weaker effects on the documents that are returned. Sample query and result screens are shown in Figure 1.

3.1 NZDL Usage Profile

Details of NZDL usage are automatically logged. Users remain anonymous in the usage logs. Through analysis of the logs we can develop an idea of how users form queries, which in turn can help us to provide more effective interface support. We will discuss usage of the CSTR collection, a detailed analysis of which is provided in Jones et al [1]. The target audience for this collection are Computer Science researchers and so our expectation is that its users are familiar with Boolean logic and search strategies using Boolean queries. A study of this particular population can give insight into characteristics of what we might expect to be the best user performance with Boolean queries over the library as a whole.

Insert Table 1 near here

The logs were considered over two consecutive periods. In the first, Boolean querying was provided by default. In the second, ranked querying was the default query mechanism. In each case, users had to explicitly select the alternative. Table 1 summarises the distribution of ranked and Boolean query use[2]. We see that two thirds of users use the default query method regardless of what it is, and this holds true over further periods for which the logs have been analysed. Nevertheless, even with ranked querying as the default, a substantial proportion of queries (a third) follows from explicit selection of Boolean querying by users. It appears that in many cases the preferred mode of searching is via Boolean queries.

Insert Table 2 near here

---

[2] All figures exclude use from within the Department of Computer Science at the University of Waikato so that testing and evaluation use (which would be highly uncharacteristic) is not considered.

We can also determine the distribution of the number of query terms for all queries. This data is summarised in Table 2 (n=32796 because six queries used 17 terms or more and were discounted). Just under 80% of queries used one, two or three terms. This is marginally fewer than reported by Clarke et al [4]. Two terms are most commonly used and make up a third of all queries. The mean number of terms in across all queries is almost 2.5, with a slight variation between Boolean and ranked queries (means of 2.23 and 2.57 terms respectively). A further 16.6% use between four and seven terms. These figures raise the question of whether users can adequately express most of their information searching needs with one, two or three terms, or whether the methods for specifying more complex queries prove too much of a challenge.

Insert Table 3 near here

From the usage logs we can also identify the frequency of use of Boolean operators within multiple term Boolean queries. This data is summarised in Table 3. The intersection operator (&) is by far the most common, appearing in a quarter of all Boolean queries. The mean number of intersection operators in queries where they were used was approximately two. The union operator is used 10 times less frequently than intersection, and negation is used in only 1% of queries. Parentheses are used in only a small percentage of queries (4.6%) to produce compound expressions. 18% of queries use four or more terms; therefore it is perhaps the case that users do not effectively use parentheses to manage query complexity. These observations indicate that although Boolean querying is explicitly selected for a third of the queries, the queries tend to be simple in the logic that they express, mainly using only the intersection operator.

It is also interesting to consider the level of query refinement that users carry out. The information required to determine this has only been logged more recently than the data sets used above. Here we consider a further set of 23550 queries within the CSTR.

Insert Table 4 near here

Consecutive pairs of queries submitted by individual users were analysed. The frequency with which terms were repeated from a user's preceding query is summarised in Table 4.

Approximately one half of the queries were completely distinct from the previous one, sharing no terms. The most frequent number of common terms is 2 for Boolean queries (for 17.82% of queries) and 1 for ranked queries (for 18.76% of queries). The mean number of repeated terms is 1.22 for Boolean queries and 0.96 for ranked queries. It appears that query refinement with respect to the search terms used is a common user activity, occurring slightly more frequently for Boolean queries than ranked. In particular, users often build on previous queries to form a new query, most commonly reusing one or two terms.

In most cases Boolean queries are short and simple, but expansion of queries with further terms is common. It may be the case that such strategies enable users to effectively find useful information, and that no further support is necessary. However, this view is confounded by the observation that for two thirds of Boolean queries none of the returned documents are accessed by the user. Therefore investigation of alternative user interface techniques to support Boolean query specification seems necessary. In the following sections of the paper we describe how graphical representations of Boolean queries might be a suitable approach.

4 A Study of Venn Diagram Usage

We carried out a paper based study to determine how users form and interpret graphical Venn-like diagrams which correspond to Boolean queries. The study was similar in nature and intention to that of Thomas [13]. It was carried out in two parts. In the first part of the study users formed natural language expressions representing their interpretations of given Venn-like diagrams. This enabled us to investigate if our envisioned mapping of diagram to expression matched those commonly identified by users. In the second part users formed Venn-like diagrams to represent given natural language Boolean expressions. This allowed us to consider several questions. First we wanted to determine how effectively users can manipulate Venn diagrams to represent Boolean expressions. Second we were interested in whether users create strict Venn diagram representations, and if not whether common variations can be identified. Third we wanted to determine if the use of both set assembly and set refinement approaches as identified by Willie [15] could be confirmed.

4.1 Subjects

A sample of 18 students enrolled in first year undergraduate computer science courses was randomly selected. At the time of study the subjects had received no more than a few weeks of tuition, and the majority were taking computer skills courses rather than Computer Science courses. 12 subjects were enrolled in a variety of arts and science majors across a broad range of disciplines. Only 6 intended to major in Computer Science. None could be said to have a Computer Science background or expertise, although all had some prior experience of using computers. 16 of the subjects had some experience of using the Internet and had all used one or more WWW based search engines. 10 subjects had prior experience of Venn diagrams during high school study, six experience elsewhere and two had no prior experience. 14 had English as a first language. 12 subjects were aged between 18 and 20 and the remainder were aged between 21 and 29. There were eleven male and seven female subjects.

Subjects were given no training in how to draw or interpret Venn diagrams.

4.2 Method

Each subject completed two tasks:

Task A: subjects formed English language expressions to represent their interpretation of given Venn-like diagrams.

Task B: subjects formed Venn-like diagrams to represent given English language Boolean expressions.

Insert Figure 2 near here

Insert Table 5 near here

Subjects were alternately assigned to two groups. Presentation order of the tasks for each group is shown in Table 5.

In Task A 11 Venn-like diagrams were presented for interpretation. Subjects were told that they represented Internet search engine queries. The form of diagrams is shown in Figure

2(V1-V11). They become progressively more complex and use both set assembly and set refinement. The subjects were asked to write down a description of the shaded areas. For queries involving more than 3 terms we diverge from strict Venn representation because of the recognised difficulties in displaying 4 or more overlapping sets. We allow non-overlapping sets, and were interested to determine if subjects interpreted them as intersection or union relationships.

Insert Table 6 near here

In Task B nine English language representations of Boolean expressions were presented. The expression structures are shown in Table 6. Again subjects were told that they represented Internet search engine queries. The expression structures correspond directly to the Boolean expressions represented diagrammatically in Task A. There are two fewer because there is no set assembly/refinement distinction. Subjects were asked to draw, using any diagramming method, representations of the expressions. A suggestion that '...one approach may be to represent each query term with a circle and to arrange the circles so that you can shade an area to represent the query' was made.

The descriptions, diagrams and time taken by subjects to complete each task were recorded. On completion of each task subjects were asked to rate their confidence in the correctness of their solution to each question. Responses were given on a seven point scale, 7 being very confident, 1 no confidence. Subjects were also asked if they had any general comments about the task. The first task was removed before the second was administered.

4.3 Results and Discussion

Insert Table 7 near here

Each subject's responses in Task A were trimmed to their most concise expression, removing superfluous text, and then grouped and counted. Table 7 summarises subject responses, labelling terms A, B, C and so on to clarify the Boolean expression structures.

Overall, we see that across all subjects there is a consistent and accurate interpretation of simple set intersections, and that intersection is represented with the word AND (V1, V3, V5 and V6). This indicates that in Venn diagram interpretation users can overcome the difference between the exclusive AND of Boolean logic and the inclusive AND of English language. When groups A and B are compared there is no discernible difference in success in interpretation of intersections. Simple negation (V7) was successfully interpreted across all but one of the subjects, expressed with the word NOT.

Simple union operations (with two or three terms) produce a variety of responses (V2 and V4). A number of subjects responded to these questions by using the word AND. There are two possible explanations. First, subjects may have interpreted the diagrams as intersections and appropriately used the word AND to match their interpretation. Second, the conflict between inclusive OR of Boolean logic and exclusive OR of English language may have proven difficult for subjects. In this case they may have expressed inclusivity with the inclusive English word AND. It is interesting to note that a number of subjects provided redundant detail in interpretation of union expressions. In both V1 and V4 some subjects included the intersections between sets which are already included in their union, being exhaustive in their description of shaded areas of the diagrams. This implies that for a diagram such as in V2 subjects identified three components: A not B, B not A, A and B. It may be that these subjects think of set A as implicitly being A *only* using an implicit Boolean negation operator.

Set refinement representation (V3 and V6) was consistently interpreted accurately by the majority of subjects. However, there was more variation displayed than in the intersections represented by set assembly.

In some cases (V2, V3, V6, V11) a number of subjects incorrectly combined two query terms into a single term. This is likely because the terms involved were “waikato” and “university”, which naturally combine to “waikato university” for students of the institution.

Subjects had no difficulty in identifying multiple set collections (V9, V10 and V11) and representing them distinctly in their responses. Less than half of the subjects identified the non-overlap of sets within these diagrams as a union operation expressed with OR. The majority expressed non-overlapping set relationships with AND. It is unclear whether their intention was to represent inclusion or exclusion. Interestingly, the redundancy of V4 was not repeated for V10.

There is some indication of a learning effect with respect to V9, V10 and V11. For multiple set collections more subjects in group A interpreted non-overlapping sets as intersections expressed by AND than unions expressed by OR. For group B (who interpreted diagrams as the second task) more subjects interpreted non-overlapping sets as unions expressed by OR than intersections expressed by AND.

Insert Table 8 near here

Table 8 shows subjects' confidence in the correctness of responses; how well they thought they represented the given diagram. Responses were given on a seven point scale, 7 being very confident, 1 no confidence.

Insert Table 9 near here

All subjects were positively confident that their English interpretation correctly represented the meaning of the diagrams. Of the simple queries (V1 to V7) there is lower confidence for the first instance of set refinement (V3). This is not the case for the second instance (V6). Group A exhibits a decrease in confidence for more complex diagrams (V8 to V11), whereas for group B there is a small increase for V8 and V9 and little change for V10 and V11. The mean confidence across all questions is higher for group B than group A. However, there is no significant difference (at the 5% level) between the confidence of the two groups. This may be due to a true lack of effect of task order on confidence. For Task A there is no significant difference (at the 5% level) between the time taken by Groups A and B. Table 9 summarises the observations.

Insert Table 10 near here

Table 10 summarises the responses for Task B in which subjects formed Venn-like diagrams to represent given Boolean English expressions. Group B formed the diagrams prior to the interpretation task, and Group A formed diagrams after the interpretation task. As with Task A, we see that across all subjects there is consistent and accurate representation of set intersection and negation for simple queries (E1, E4, E5). This is also evident for more complex queries using intersection and negation (E8 and E9). This indicates that the phrasing used to represent intersection and negation was interpreted as intended without apparent conflict between the English language and Boolean logic meanings. Both set refinement and assembly approaches were used for intersection of three sets, but only set assembly for intersection of two sets. Again, as with Task A, subjects seemed to experience more difficulty in dealing with set unions. This is consistent with the findings of Michard [10]. The responses to E2 and E3 show that some subjects interpret expressions such as A or B as 'A only or B only', and exclude the set intersections from their diagrams.

Overall the most common responses directly matched the representation that we presented in the interpretation task. Group A subjects display a strong match even though the diagrams were removed when Task A had been completed. With little prior experience they could remember and reproduce the given diagrammatic notation. Group B responses were more varied as we might expect given that they had not seen any example diagrams. However, for E1 to E6 the most common response was still the representation that we presented in the interpretation task There is much more variation within Group B for the multiple set collection queries (E7, E8 and E9). It seems that this type of query is prone to idiosyncratic representations without prior experience of an appropriate diagram form. No subjects bounded their set combinations with a representation of the universal set, which seems to be implicit in their diagrams.

All subjects were positively confident that their diagrams correctly represented the given English language Boolean expressions. The mean confidence across all questions is higher for Group A than Group B. However, there is no significant difference (at the 5% level). This indicates that subjects' confidence was not increased significantly by experiencing a diagrammatic notation prior to forming their own diagrams. Again, this may be due to a true lack of effect of task order on confidence, or to the small sample sizes of each group.

For Task B there is a significant difference (at the 5% level) between the time taken by Groups A and B. Table 9 summarises the observations. Group A formed diagrams significantly faster than Group B. This is most likely attributable to their prior experience of a diagrammatic representation in Task A.

The incidence of revisions of diagrams across all subjects was low. Only 18 of 162 diagrams (11%) were altered and this was evenly spread between the two groups. This is consistent with the confidence in the responses that subjects reported.

4.4 Implications For a Venn diagram Based User Interface

The study is encouraging, in that it reveals a reasonably high level of consistency across subjects in interpretation and formulation of Venn-like diagrams which represent Boolean expressions. Although incorporation of multiple variations of Venn representation into a user interface might support a range of users, it introduces complexity and potential ambiguity. Given the commonality of subject responses, and particularly the way in which Group A were able to reuse the given representation style with minimal exposure to it, we believe that a single representation will be effective. Use with minimal or no prior experience is an important aspect of a user interface for a WWW based information source.

Both the set assembly and refinement approaches were used and should be provided in a Venn based user interface. Set refinement seems to have utility in expressing more complex intersections.

Intersection and negation operations were interpreted and represented with little variation and the representations used in Task A seem to be appropriate.

A noticeable issue was in how subjects dealt with union of sets. Particularly, some users considered the union of two sets to exclude their intersection. This and other minor variations across subjects may be dealt with by provision of appropriate feedback on the semantics of diagrams which they create.

The English expression of Boolean queries produced little variation in the Boolean logic represented by subjects in their diagrams. Variations in the diagrams were due to idiosyncratic representations of the same logic. We therefore assume that the forms of English expression which we used were not ambiguous and could be used in a user interface to provide feedback to users. Although only 10% of diagrams were revised we believe it is important to provide tools which allow diagrams to be amended. This is particularly important for query refinement.

Insert Figure 3 near here

Our study dealt with the overlap of a maximum of three sets in any one set collection, but considered how effective a non-overlapping representation is for representing four or more sets. In Task A half of the subjects interpreted non-overlapping collections of terms (V9-V11) to represent intersection. However, in Task B slightly more than half created diagrams in which non-overlapping collections represented union. Noticeably, most of these subjects had seen this representation in the first task that they did. Therefore, it seems that with little training this representation is useful for forming more complex queries.

As we described in Section 3.1 approximately 19% of queries were composed of more than three terms. The organisation of four or more circles in a diagram to enable selection of intersections is difficult. A method for elegantly showing four overlapping sets is shown in Figure 3 but is unlikely to be easily created by users. A second approach may be used for queries containing four or more overlapping sets. Some abstraction technique could be used to allow multiple query components to be 'collapsed' into a single graphical entity. This was suggested by Michard [10] but was responsible for the introduction of errors in query formulation, and so its design must be carefully considered.

5 VQuery

Figure 4 shows the VQuery interface. It has two main components: the query window and the result preview window. The query window is divided into three main areas. To the top left is the query workspace. This is a window in which the user can organise query terms to create Boolean queries. To the bottom left is the natural language query view—a text panel in which an English interpretation of the current query is displayed. To the right is the control panel.

5.1 A Workspace for Query Manipulation.

The query workspace is a large scrollable window, which when fully expanded provides a full screen workspace. It displays circles containing query terms which may be phrases or single words. Each circle may be thought of as the set of documents containing the term within the circle, and below each term is a number indicating how many documents are in that set. A new term is created by typing a word or phrase into the term entry box within the control panel and the number of terms that can be added to the workspace is unlimited. The circles can be selected (as 'Tarzan' currently is), moved and resized using standard pointing and dragging with a device such as a mouse. Multiple terms can be selected, moved and resized using a keyboard modifier. Selected terms can be removed using the delete key. Set assembly and refinement representations are both supported – terms can overlap (such as 'lion' and 'tiger' in Figure 4) and be fully contained within other terms (as 'cheetah' is within 'Jane' in Figure 4)..

The workspace is divided into two areas; an active query area within the rectangle labelled 'Active query' and a non-active query area outside of it. The active rectangle contains the terms that the user is combining to form a query and the non-active area acts as a storage area for terms which have been used in the past or may be used later, but do not contribute to the current query. Terms can be dragged in and out of the active area to amend the active query. The active query area can be moved or resized by dragging its bounding rectangle, providing a second method for inclusion or exclusion of terms.

Through the arrangement of term circles and positioning of the active query area users can create complex queries containing multiple terms and using combinations of the AND, OR and NOT operators. There is a distinction between selection of term circles for moving and resizing operations and for what we term *query-selection*. Query-selection occurs when the user points at or clicks within a term circle, to indicate that the term is to form part of the query. When this happens a coloured dot appears in the term circle to indicate that it is selected.

An AND operation on two or more terms is represented by query-selection of the intersection of the circles of those terms. An OR operation on two or more terms is represented by query-selection of the circles of those terms but not in the area of their intersection. Any term circles within the active area which are not query-selected are combined with other terms using the NOT operator. The query shown is Figure 4 is ((lion AND tiger) OR jungle) NOT Tarzan.

Shneiderman [12] has argued for dynamic query interfaces which reflect changes in query results as the query itself is refined. The query selection mechanism provides such immediate feedback. Query-selection of terms can be either transient or fixed. Transient selection is used when immediate previews of results of queries involving single terms or the AND operator are required. In this case, when the user moves the mouse pointer into a circle a dot (which is red in the interface) will appear at its centre indicating that it is to be used within the query being formed ('jungle' in Figure 4 is transiently selected). The results preview window will be updated to contain the titles of the documents which contain the term within the circle. When the mouse pointer leaves the circle, the dot is removed and the document titles associated with the term are removed from the results preview window.

If the mouse pointer is moved into the intersection of two or more circles a dot appears at the centre of their intersection, and the circles are joined by lines, to represent their association in the AND operation. Again the results preview window is updated, and the interface reverts to its prior state when the mouse pointer leaves the intersection.

When the user is satisfied with a component of the query its query-selection can be fixed by clicking within its circle or the intersection of multiple circles. In this case the query-selection is not removed when the mouse pointer leaves the selected region. This is useful for experimentation with addition of other components to a base query, and is necessary to allow specification of the OR operator using disjoint term circles. Fixed query-selections can be removed by a second click within the region of the selection.

5.2 Result Previews

The result preview window contains a label which indicates how many documents match the currently selected query and a list of titles of the documents which will be returned by the currently selected query. The current query may be refined through transient selection, fixed selection, term deletion or manipulation of the active query area, and in each of these instances the result preview is immediately updated to reflect the change. Single or multiple selections of items within the title list can be made using the mouse and a keyboard modifier. When the user double-clicks on selected items, the full text of those items is retrieved from the NZDL and displayed to the user.

5.3 Query Reinforcement

The natural language query view is situated below the query workspace. It presents an English language interpretation of the active query, which for new users reinforces the semantics of the visual notation, and provides feedback as to the meaning of a selected query. It is immediately updated whenever the active query is amended. We use a simplistic transformation, attempting to reduce ambiguity in the English expression. English versions of Boolean expressions are inserted into boilerplate text which gives user instructions and indicates which collection is to be queried.

5.3 Storage and Retrieval of Queries

Users are provided with utilities to store and retrieve VQuery state both within and between query sessions. To save the current state of a query workspace the "Bookmark current state..." command is used, accessible through the Bookmarks menu. When this is selected the user is prompted to supply a textual label for the bookmark. This label is then added to the Bookmarks menu. There is no limit on the number of bookmarks which can be created in a VQuery session. When a bookmark is selected from the menu the current workspace state is replaced by the state at the time that the bookmark was created. The replication of the state is exact, including size, positioning and selection of all components of the query workspace. Each bookmark is stored in a concise ASCII file format.

Persistent storage of a querying session is also supported. Commands to save and load sessions are accessible through the File menu and users are prompted with standard file selection dialogs to name and select sessions. Both session and bookmark files can be communicated to other VQuery users so that queries and results may be shared between people.

6 Evaluation of VQuery

We administered a user study to determine the efficacy of the VQuery design. The study had two aspects. The first was to determine if user performance with VQuery interface is better than with a standard textual interface for Boolean query specification tasks. The performance measures that we were particularly interested in were the speed with which tasks were completed and the correctness of the Boolean queries that were formed. The second aspect of the study was a qualitative analysis based on user perceptions of the two interfaces. The textual query language against which VQuery was compared was that of the NZDL. In the NZDL intersection is represented by &, union by | and negation by !. Precedence can be controlled by parenthesising portions of the query.

6.1 Subjects

12 subjects were recruited for the study. All subjects were university students and were enrolled in degree programmes in a range of subject areas. All subjects had prior experience of forming Boolean queries in textual query languages and were aware of Venn diagram representations of Boolean expressions.

6.2 Method

Each subject completed two tasks:

Task A: translation of given English language representations of Boolean expressions into NZDL notation;

Task B: translation of given English language representations of Boolean expressions into VQuery notation.

Insert Table 11 near here

Subjects were alternately assigned to two groups. Presentation order of the tasks for each group is shown in Table 11.

Insert Table 12 near here

Insert Table 13 near here

For both tasks subjects were given a total of 9 English language queries to translate. Example queries are shown in Table 12, and the underlying structure and presentation order of the queries is shown in Table 13. The 9 queries required subjects to use each Boolean operator both individually and in combination. The queries extend beyond the characteristics of usage of the standard Boolean query interface of the NZDL. This allowed us to investigate if standard behaviour was better supported by VQuery, and how support for more complex query construction compared to the standard textual interface. Queries were presented to subjects one at a time.

The queries in this study had a restricted set of 9 terms, 5 of which were shapes (circle, triangle, square, pentagon, hexagon), and 4 of which were colours (red, green, blue, black). These terms were randomly inserted into the query structures shown in Table 13.

Both tasks were administered in a semi-automated way. For Task A a set of World Wide Web pages was constructed. Each page presented a query to the subject and provided an area into which their response could be typed, along with a brief explanation about the Boolean query syntax of the NZDL. On completion of the query the subject clicked a 'submit' button to move to the next query. The amount of time a subject spent on a query page, and their submitted Boolean query were automatically logged by the system.

For Task B VQuery was amended in the following way. A persistent dialogue window was introduced which contained the English language representation of each of the 9 queries to be created. A single document collection was made available for querying (described below), and a button was introduced which allowed the user to submit their completed query and move to the next task. Again, the time taken to form the query and the final query representation was automatically logged.

Subjects' comments and behaviour were recorded by the experimenter in both tasks.

In order to analyse the accuracy of queries formed by subjects we created a restricted set of 60 simple documents for use in the study. Each document contained only a combination of a subset of the 9 shape/colour query terms. For each query the sets of documents matching both a correctly formed query and a subject's submitted query were determined.

6.3 Results

Insert Figure 5 near here

Insert Table 14 near here

Table 14 shows the mean completion times for each query using the two styles of interface. Using the textual interface the mean time taken to complete all tasks was a little over 7 minutes. For VQuery it was a little under 19 minutes. There is a clearly significant difference in the mean times taken using the two styles of interface. Overall, subjects took almost three times longer to complete the query tasks with VQuery than they did with the textual interface. There was a degree of variation between performances of individual subjects—the shortest time taken for the textual task was 4 minutes and 22 seconds, the longest was 11 minutes and 39 seconds. For the textual interface the standard deviation in completion times is just under two minutes. Using VQuery the shortest time taken was 8 minutes and 18 seconds, the longest was 29 minutes 52 seconds, and the standard deviation of the times is approximately 7 minutes.

When we consider the individual queries we see that those that contained three or more terms took longer to complete than those that contained two. Across all subjects the mean completion time for queries 4 to 9 was highly similar with the textual interface, although there is more substantial variation for those queries between groups A and B. There is a greater degree of variation for queries 4 to 9 over all subjects using VQuery. Queries 7, 8 and 9 in particular took longer to form than the others.

When we consider whether task ordering resulted in a difference in completion times we find a significant difference at the 5% level (one-tailed t-test) between groups A and B in the textual task. Subjects who performed the VQuery task first then completed the textual task significantly faster than the group who performed the textual task first. When we consider the VQuery task we find that task ordering had no significant effect.

Figure 5 shows the cumulative completion times for the tasks. We see that times taken in the two conditions begin to diverge immediately. For the textual task there is a steady, linear increase in time taken. For the VQuery task there are two clear points at which the time taken on queries increases. The first is at the transition from two-term to four-term queries requiring the union operator. The second is at the transition to four term queries requiring the intersection operator.

Insert Table 15 near here

Table 15 shows the accuracy achieved by the subjects. Over all subjects 10 queries (of 108) were formed incorrectly using the textual interface, giving an accuracy rate of 90.7%. This is significantly different to the accuracy achieved using VQuery—69% over all subjects. In respect to the textual task, the subjects who performed it second produced slightly fewer erroneous queries than those who performed it first. Using VQuery, the subjects who had performed the textual task first produced slightly more errors than those that used VQuery first.

For both systems the majority of errors occurred in the final three queries (7-9). With the textual interface seven out of ten errors occurred in these queries. The impact is more pronounced for VQuery with 28 of 33 errors occurring in queries 7, 8 and 9. For queries 1 to 6 the numbers of errors are similar for the two systems—3 errors with the textual interface and 5 errors using VQuery.

6.4 Discussion

There are a number of reasons as to why subjects took more time when using VQuery. First, all subjects had prior experience of forming textual Boolean queries (although not necessarily in the form required in the study). They were given brief instruction on how to use VQuery, but their experience with VQuery was less than they had with textual approach to query specification. We did not provide training in the use of VQuery because its intended target user group is casual users of query systems. A comparison of performance without training is an important consideration given that we a proposing to introduce a query technique which substantially differs from those experienced by users in common query environments such as digital libraries and Web search engines. User acceptance of the system is likely to be poor if it appears to support their querying task less efficiently than the interfaces that they currently use.

The second reason for the time differences is less related to the subjects' expertise with VQuery. Query specification in VQuery is a three stage process: users specify the required query terms, organise the query term circles into an appropriate spatial layout, and then select the areas of the layout which represent their required Boolean expression. An overhead is incurred in the physical act of moving and resizing the circles, and we observed that users spent a noticeable amount of time rearranging circles into a layout that they were happy with. This overhead is not present in the textual language. Additionally, whenever subjects made typing errors while entering query terms, they would have to reenter the term correctly to create a new term circle for the correctly typed term. In the textual interface existing text could be easily edited. An expert user of VQuery created the queries required in the tasks given to the subjects, taking just over 10 minutes to complete all 9 queries. This is closer to, but still more than, the mean time taken by subjects in the textual task Therefore even with substantial training, the overhead of manipulating the graphical representation renders VQuery slower than a purely textual interface.

A third reason for the disparity in time taken between the systems is the difficulty experienced by subjects in query-selecting portions of the diagram. In the case of intersections of terms the overlap between circles must be sufficiently large for the presence of the mouse cursor to be recognised and a selection recorded. Often users would click on a small overlapping area but the system would fail to recognise this as a selection of an intersection. Users would then have to deselect the selected area and rearrange the diagram before trying again. This effect is particularly noticeable for queries 7-9 where subjects had to select intersections of multiple overlapping circles.

It is clear from these observations that there are several areas for redesign of VQuery. First it is necessary to allow the text of query terms to be edited within the query workspace to alleviate the overhead of creating new terms circles when typographical errors have occurred. Second, to reduce selection errors the sensitivity with which selection of circle intersections is recognised must be improved, or overlapping areas could be constrained to a minimum size. Most importantly the overhead in managing the arrangement of term circles must be alleviated. One way to achieve this is through the provision of query templates for common Boolean expressions. Users might select from a list a template which would appear in the workspace and then allow query terms to be associated with each term circle. This would extend the restrictive three term template provided by GQL [10] and TeSS [7]. Templates might be provided in both textual and graphical forms. Templates could be automatically organised within the workspace to ensure that the full range of selections was possible. Additionally, graphical items in the template could include both set assembly and set refinement forms of intersections between terms.

Group B (subjects who completed the textual task after the VQuery task) completed the task significantly faster than Group A. It seems that the use of VQuery had a positive impact on the speed with which subjects could then form textual queries. This might simply be due to increased familiarity with the task and the queries to be formed. However from observations of, and comments from, the subjects it appears that the textual task was commonly treated as a translation task from the English language query to the Boolean syntax. With VQuery the translation was not so direct. Subjects thought more deeply about the semantics of the query rather than merely its representation—a behaviour which then supported more rapid formulation of textual queries. We believe that integration of diagrammatic Venn representations of queries into user support materials (such as on-line help) may help users to more rapidly form Boolean queries in standard textual interfaces.

The mean completion time for one query (query 7) was almost twice that of any other when using VQuery. This is because subjects had to expand the underlying Boolean expression (A OR B) AND (C OR D) into (A AND C) OR (A AND D) OR (B AND C) OR (B AND D) in order to be able to represent the query. Even if they produced the correct expansion arrangement of term circles would be difficult. Again, this problem may be alleviated by the user of query templates.

The increase in task completion time resulting from the use of VQuery might be acceptable to users if it reduced the number of erroneous queries created. Unfortunately far more erroneous queries were produced with VQuery, although for queries 1 to 6 the number is comparable to the textual interface. The major disparity occurs in queries 7 to 9. These queries differ from the others in that they require expansion of the expression, or intersections of three or more overlapping term circles to be selected for the query to be accurately formed. As described above the selection mechanism did not effectively support precise selections of overlaps unless the diagram had been organised appropriately. This is predominantly a user interface issue which can be resolved by refinement of the selection mechanism.

An observational analysis of the subjects provides additional insight into the timing and accuracy measures. Notes were taken regarding subject behaviour during the tasks, and comments that they made were recorded. A common positive comment was that the natural language feedback regarding the currently selected query was useful, enabling incorrect queries to be rapidly identified and refined to produce the required Boolean expression. Although subjects produced many erroneous queries with VQuery *they knew* that they were erroneous and often commented to this effect—they simply could not form the appropriate graphical representation. This kind of immediate feedback is useful, and could be integrated into a textual query interface. In fact several subjects found it frustrating that the textual interface did not provide this feedback.

Subjects used differing strategies to form the graphical queries. One strategy was to first create term circles for all terms within a multi-term query, and then to organise them to enable selection of the appropriate query. Another strategy was to incrementally form the query, adding to and organising the graphical representation term by term. The first strategy had not been envisaged and was poorly supported in VQuery—all term circles initially appeared in the same position in the workspace. This problem can be resolved by offsetting the position of new term circles from previous ones. Additionally four subjects used set refinement rather than set assembly techniques to form intersections—for some users the ability to do so is a supportive aspect of VQuery.

Despite the slower and less accurate performance with VQuery several subjects viewed its use as a positive experience commenting that its was more fun and easier to use than the textual interface.

For the purposes of directly comparing VQuery against the NZDL query interface we used a restricted set of Boolean query expressions. There are some Boolean expressions that can not be represented using VQuery. Disjoint term circles (or groups of term circles) that contain a query selection are viewed as unions, and those that do not contain a selection are viewed as negations. Consequently an expression such as A NOT (B AND C) can not be represented. Further work is required to investigate how the current representation can be extended to include such expressions.

7 Conclusions

The first study presented in this paper indicates that there is a high level of consistency in the way that users both interpret and form Venn-like representations of Boolean expressions even though they may not conform to strict Venn diagram notation. Drawing from these results we have developed VQuery, a system for interactive graphical specification of Boolean queries, which uses the most common notations revealed in the study. A second study comparatively evaluated VQuery and a standard textual Boolean query language. The results show that subjects took significantly longer, and produced more erroneous queries when using VQuery than they did with the textual interface. These results are at odds with previously reported speed and accuracy increases resulting from the use of Venn diagram notations for Boolean query specification.

The key difference between VQuery and other Venn diagram oriented query systems which have produced more positive results is the level of flexibility provided. Other systems, such as GQL [10] and TeSS [7] provide restricted Venn diagram templates in which three overlapping set circles are constantly displayed. Terms or expressions entered by users are abstractly matched to circles in the templates by intermediate labels. In contrast VQuery allows users to form queries containing more than three terms, and gives them the freedom to manage the spatial layout of those terms as they wish. This supports more complex queries and both set assembly and set refinement strategies for creating expressions involving intersection. However, it is clear that for many users there is a substantial overhead involved in the manipulation and organisation of term circles leading to increased query specification times. We suggest that standard query templates (that extend those provided in GQL and TeSS) may alleviate this problem. Such templates would provide a starting point for users, and then support interactive rearrangement as users refine the base query. In more exploratory tasks (where the goal is not a specific target expression) the dynamic querying aspect of VQuery will allows users to exploit such templates to rapidly investigate a range of expressions without the refinement overhead incurred with a textual interface.

The accuracy with which users can create queries in VQuery is clearly also an issue. This is mainly attributable to the sensitivity of the query selection mechanism within the interface. Refinement is evidently required to reduce the frustration and errors that users experienced as a result.

Overall, although the performance measurements for VQuery were disappointing, we are encouraged by the positive responses to the system by subjects despite the frustrations that many experienced. The insights gained from the study will support redesign and evaluation VQuery to address the observed problems.

8 Future Work

The manner in which Boolean expressions are represented in VQuery is one aspect of its support for query formulation. A further aspect is the application of dynamic querying techniques which enable users to rapidly explore a range of relationships between query terms without reformulating the query as would be required in a standard textual query language. Users can get an idea of how many and what sort of documents will match their query as they refine it. Once we have resolved the interface design issues relating to query specification we will investigate how well dynamic querying supports users' document retrieval tasks.

The collaborative nature of information browsing and retrieval has been highlighted by Twidale and Nichols [14]. In the NZDL we would like to support collaboration between users, but also between users and the NZDL librarian who deals with users requests, offers advice on its use, and suggests how queries might be formed and refined. We have developed a prototype system Collaborative VQuery, implemented in GroupKit [11], which allows geographically distributed users to collaborate on an information retrieval task. They can create and manipulate queries on a shared workspace and communicate through textual message windows. We will investigate the efficacy of this system in support of common collaborative tasks within the New Zealand Digital Library.

Acknowledgements

Thank you to the anonymous referees for a range of useful comments on an earlier draft of this paper.

References

1. Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H.F., Secret, A.: The World-Wide Web. Communications of the ACM, 37(8):76-82, 1994.
2. Borgman, C.L.: The User's Mental Model of an Information Retrieval System: an Experiment on a Prototype Online Catalog. International Journal of Man-Machine Studies, 24:47-64, 1986.
3. Davies, T., Willie, S.: The Efficacy of a Venn-based Query Interface: an Evaluation. In; Balbo, S. (ed.): Proceedings of QCHI95 Symposium, Bond University, Queensland, Australia, August 1995, pp 41-50.
4. Clarke, C.L.A., Cormack, G.V., Tudhope, E.A.: Relevance Ranking for One to Three Term Queries. In: Proceedings of RIAO95: Computer Assisted Information Searching on the Internet, McGill University, Montreal, Canada, June 1997, pp 388-400.
5. Greene, S.L., Devlin, S.J., Cannata, P.E., Gomez, L.M.: No IFs, ANDs or ORs: a Study of Database Querying. International Journal of Man-Machine Studies, 32:303-326, 1990.
6. Halpin, T.A.: Venn Diagrams and SQL Queries. The Australian Computer Journal, 21(1):27-32, 1989.
7. Hertzum, M., Frøkjær, E.: Browsing and Querying in Online Documentation: a Study of User Interfaces and the Interaction Process. ACM Transactions on Computer-Human Interaction, 3(2):136-161, 1996.
8. Jones, S., Cunningham, S.J. and Rodger McNab. An Analysis Of Usage Of A Digital Library. In: Research and Advanced Technology for Digital Libraries: Proc ECDL'98 Second European Conference, Heraklion, Crete, 1998. Lecture Notes in Computer Science no. 1513, p 261—277 (Springer).
9. Katzeff, C.: The Effect of Different Conceptual Models Upon Reasoning in a Database Query Writing Task. International Journal of Man-Machine Studies, 29:37-62, 1988.
10. Michard, A.: Graphical Presentation of Boolean Expressions in a Database Query Language: Design Notes and an Ergonomic Evaluation. Behaviour and Information Technology, 1(3):279-288, 1982.
11. Roseman, M., Greenberg, S.: Building Real Time Groupware with Groupkit, a Groupware Toolkit. ACM Transactions on Computer-Human Interaction, 3(1):1-37, 1996.

12. Shneiderman, B.: Dynamic Queries for Visual Information Seeking. IEEE Software, 11(6):70-77, 1994.
13. Thomas, J.C.: Quantifiers and Question Asking. IBM Research Report 7021976, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 1976.
14. Twidale, M.B., Nichols, D.M.: Collaborative Browsing and Visualization of the Search Process. Aslib Proceedings, 48(7/8):177-182, 1996.
15. Willie, S.: Query Context: Would a Graphical Interface Help? In: Howard, S., Leung, Y. (eds.):Proceedings of OZCHI 94, Fourth Australian Conference on Computer-Human Interaction, Melbourne, Australia, November 1994, pp 141-146.
16. Willie, S., Bruza, P.: Users' Models of the Information Space. In: Fox, E.A., Ingwersen, P., Fidel, R. (eds.): Proceedings SIGIR95, the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, July 1995, pp 205-210.
17. Witten, I.H., Cunningham S.J. , Apperley M.D.: The New Zealand Digital Library Project. New Zealand Libraries 48(8):146-152, 1996.
18. Zloof, M.M.: Query By Example. AFIPS Conference Proceedings, 44:431-437, 1975.

Figure 1: The standard NZDL user interface, showing a query screen (a) and a results screen (b).

Figure 2 (V1-V11): Venn diagrams presented for interpretation in Task A. The generic term labels A, B, C… were replaced by particular terms in the task.

Figure 3: A graphical arrangement for four overlapping sets.

Figure 4: The VQuery user interface. The query workspace is shown to the left, and the result window to the right.

Figure 5: Cumulative task completion times for textual and graphical query formulation tasks.

Table 1. Usage of Boolean and ranked queries for the CSTR collection of the NZDL.

	Boolean as default 46 week period		Ranked as default 15 week period		Total 61 week period	
Number of queries	24687		8115		32802	
Boolean queries	16333	(66.2%)	2693	(33.2%)	19026	(58%)
Ranked queries	8354	(33.8%)	5420	(66.8%)	13774	(42%)

Table 2. Distribution of the number of terms in queries for the CSTR collection of the NZDL.

Number of terms in query	Frequency (total=32796)	Percentage
0	492	1.50
1	8788	26.79
2	11095	33.83
3	6505	19.83
4	2926	8.92
5	1477	4.50
6	692	2.11
7	342	1.04
8	209	0.64
9	130	0.40
10	53	0.16
11	31	0.09
12	27	0.08
13	15	0.05
14	5	0.02
15	7	0.02
16	2	0.01

Table 3. Frequency of Boolean operators in multiple term queries for the CSTR collection of the NZDL

	Boolean as default 46 week period		Boolean as non-default 15 week period		Total 61 week period	
Number of Boolean queries containing						
&	3731	(22.8%)	1178	(43.7%)	4909	(25.8%)
|	345	(2.1%)	122	(4.5%)	467	(2.5%)
!	181	(1.1%)	35	(1.3%)	215	(1.1%)
parentheses	682	(4.2%)	187	(6.9%)	869	(4.6%)
Total	16333		2693		19026	

Table 4. Number of terms repeated from a user‘s preceeding query for the CSTR collection of the NZDL.

Number of repeated terms	% of Boolean queries	% of ranked queries
0	50.31	54.38
1	12.83	18.76
2	17.82	15.11
3	10.35	6.67
4	4.48	2.48
5	2.17	1.17
6	1.04	0.49
7	0.54	0.44
8	0.29	0.22
9	0.10	0.14
10	0.05	0.05

Table 5. Task ordering for the two groups in the paper based study.

	First Task	Second Task
Group A	A (Venn to English)	B (English to Venn)
Group B	B (English to Venn)	A (Venn to English)

Table 6. Summary of the Boolean expressions presented in Task B of the paper based study. The generic term labels A, B, C... were replaced by particular terms in the task.

Expression	find all documents containing
E1	both A and B
E2	either A or B
E3	A, B, or C
E4	A, B and C
E5	A but not B
E6	both A and B and may contain C
E7	both A and B or alternatively any or all of C, D and E
E8	all of A, B and C or alternatively both of D and E
E9	A but not B or alternatively all of the terms C, D and E

Table 7. Summary of the subjects interpretations of the diagrams presented in Task A of the paper based study. The generic term labels A, B, C... have been substituted for the particular terms used in the task.

Diagram	Responses	Frequency		
		Group A	Group B	Total
V1	A and B	9	9	18
V2	A or B	2	1	3
	(A and B) and AB	3	0	3
	A and B	3	1	4
	(A or B) or AB	0	1	1
	A or B or (A and B)	0	3	3
	Other or no answer			4
V3	A and B	6	7	13
	AB	2	0	2
	Other or no answer			3
V4	A or B or C	3	3	6
	(A or B or C) or (A and B) or (A and C) or (B and C) or (A and B and C)	4	1	5
	A and B and C	1	2	3
	(A or B or C) or (A and B and C)	0	3	3
	Other or no answer			1
V5	A and B and C	9	9	18
V6	A and B and C	6	6	12
	A and BC	3	2	5
	Other or no answer			1
V7	A not B	9	8	17
	A and B	0	1	1
V8	(A and B and C) or (A and B)	2	5	7
	A or B	1	0	1
	A and B and C	2	2	4
	A and B	2	0	2
	(A and B and C) and (A and B)	1	1	2
	A or B or C	0	1	1
	Other or no answer			1
V9	(A and B and C) or (D and E)	2	5	7
	(A and B and C) and (D and E)	6	3	9
	Other or no answer			2
V10	(A and B) or (C or D or E)	2	6	8
	(A and B) and (C or D or E)	5	0	5
	(A and B) and (C and D and E)	1	2	3
	(A or B) and (C or D or E)	0	1	1
	Other or no answer			1
V11	(A not B) or (C and D and E)	1	4	5

(A not B) and (C and D and E)	5	3	8
(not AB) and (C and D and E)	1	2	3
Other or no answer			2

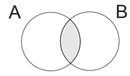
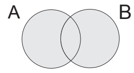
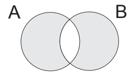
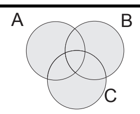
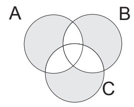
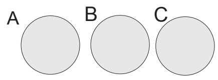
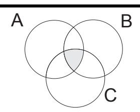
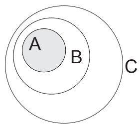
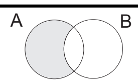
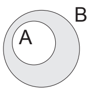
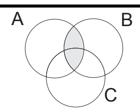
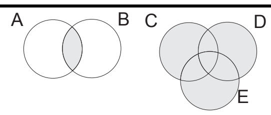
Table 8. The subjects' confidence in the correctness of their responses in Tasks A and B for the paper based study.

Task A				Task B			
Query	Group A mean	Group B mean	All subjects mean	Query	Group A mean	Group B mean	All subjects mean
V1	5.89	5.39	5.64	E1	6.11	4.72	5.42
V2	5.67	5.67	5.67	E2	6.33	5.67	6.00
V3	4.78	4.89	4.84				
V4	5.33	5.78	5.56	E3	4.78	5.33	5.06
V5	5.89	5.00	5.45	E4	6.00	5.67	5.84
V6	5.22	5.78	5.50				
V7	6.22	5.00	5.61	E5	5.11	6.11	5.61
V8	4.33	5.56	4.95	E6	5.00	4.00	4.50
V9	4.11	6.00	5.06	E7	5.22	4.72	4.97
V10	4.22	5.22	4.72	E8	5.11	4.89	5.00
V11	4.78	5.22	5.00	E9	5.33	4.56	4.95
Mean	5.13	5.41	5.27	Mean	5.44	5.07	5.26

Table 9. Subject response times in Tasks A and B of the paper based study.

	Time Taken (in minutes)			
	Task A		Task B	
	Group A	Group B	Group A	Group B
Mean	12.78	12.00	8.78	14.56
Variance	13.69	17.50	3.44	9.28
SD	3.70	4.18	1.86	3.05

Table 10. Summary of diagrams formed by subjects to match the Boolean expressions presented in Task B of the paper based study. The generic term labels A, B, C... have been substituted for the particular terms used in the task.

Expression	Responses	Frequency		
		Group A	Group B	Total
E1		8	6	14
	Other	1	3	4
E2		6	4	10
		2	1	3
	Other	1	4	5
E3		5	4	9
		3	0	3
		1	1	2
	Other	0	4	4
E4		4	4	8
		5	0	5
	Other	0	5	5
E5		9	4	13
		0	3	3
	Other	0	2	2
E6		6	3	9
	Other	3	6	9
E7		8	3	11
	Other	1	6	7

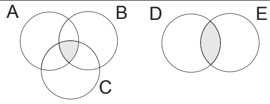
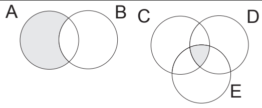
E8		8	3	11
	Other	1	6	7
E9		9	2	11
	Other	0	7	7

Table 11. Task ordering for the two groups in the interface evaluation study.

	First Task	Second Task
Group A	A (English to textual)	B (English to VQuery)
Group B	B (English to VQuery)	A (English to textual)

Table 12. Sample tasks presented to subjects in the interface evaluation study.

find the documents that are about circles but not about squares
find the documents that are about either circles or squares
find the documents that are about both circles and squares
find the documents that are about either blue circles or red squares
find the documents that are about blue shapes, red shapes, squares or triangles
find the documents that are about hexagons, pentagons or shapes that are both red and black

Table 13. Ordering and structure of tasks presented to subjects in the interface evaluation study.

Task	Boolean structure to be represented
1	A NOT B
2	A OR B
3	A AND B
4	(A AND B) OR (C AND D)
5	A OR B OR C OR D
6	A OR B OR (C AND D)
7	(A OR B) AND (C OR D)
8	A AND B AND C AND D
9	A AND B AND (C OR D)

Table 14. Mean completion times (in minutes and secons) for tasks in the interface evaluation study.

	Textual Interface			VQuery		
Query	All subjects	Group A	Group B	All subjects	Group A	Group B
1	1:20	1:27	1:13	1:37	1:20	1:55
2	0:25	0:28	0:21	1:08	0:57	1:20
3	0:26	0:34	0:18	0:52	1:00	0:44
4	0:46	0:52	0:40	2:23	2:17	2:30
5	0:54	1:08	0:41	1:51	1:40	2:03
6	0:49	0:57	0:40	1:36	1:37	1:35
7	0:55	0:57	0:54	4:23	6:04	2:43
8	0:50	1:02	0:38	2:18	2:47	1:48
9	0:47	0:52	0:42	2:40	2:54	2:27
Mean total time	7:12	8:18	6:06	18:49	20:35	17:04

Table 15. Accuracy of subject responses (number of subjects who correctly formed queries) in the interface evaluation study.

	Textual Interface			VQuery		
Query	All subjects	Group A	Group B	All subjects	Group A	Group B
1	12	6	6	11	5	6
2	11	5	6	11	5	6
3	12	6	6	12	6	6
4	11	6	5	11	5	6
5	12	6	6	12	6	6
6	11	6	5	10	6	4
7	8	3	5	3	1	2
8	10	5	5	3	2	1
9	11	5	6	2	1	1
% of all queries that were correctly formed	90.7%	89%	92.3%	69%	68.5%	70.4%