Working Paper Series
ISSN 1170-487X

A Graphical User Interface for Boolean Query Specification

by Steve Jones and Shona McInnes

Working Paper 97/31
December 1997

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

A graphical user interface for boolean query specification

Steve Jones **Shona McInnes**

Department of Computer Science
University of Waikato
Private Bag 3105, Hamilton
New Zealand

Telephone : +64 7 838 4490
Facsimile : +64 7 838 4155

Email : stevej@cs.waikato.ac.nz

Abstract

On-line information repositories commonly provide keyword search facilities via textual query languages based on Boolean logic. However, there is evidence to suggest that the syntactical 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. 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. We describe VQuery, a Venn-diagram based user interface to the New Zealand Digital Library (NZDL). The design of VQuery has been partly motivated by analysis of NZDL usage. We found that few queries contain more than three terms, use of the intersection operator dominates and that query refinement is common. 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 which accurately correspond to Boolean expressions. The utility of VQuery is considered and directions for future work are proposed.

Keywords: searching, user interface, Boolean logic

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 (Berners-Lee, 1994) based Internet search engines and some digital libraries (such as the New Zealand Digital Library; Witten et al, 1996) 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 virtually all Internet search engines. However, it has been shown that difficulties in dealing with Boolean logic are common, particularly when a restricted syntax is used (Borgman, 1986; Katzeff, 1988; Greene et al, 1990; Hertzum & Frøkjær, 1996). 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. To conclude the paper 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 (1976) 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 (1982) 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 maximim 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 (1988) 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 (1989) 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 (1995). 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 bewteen 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 (1995) carried out a comparative study of a Venn diagram user interface and a simple Query By Example (Zloof, 1975) 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.

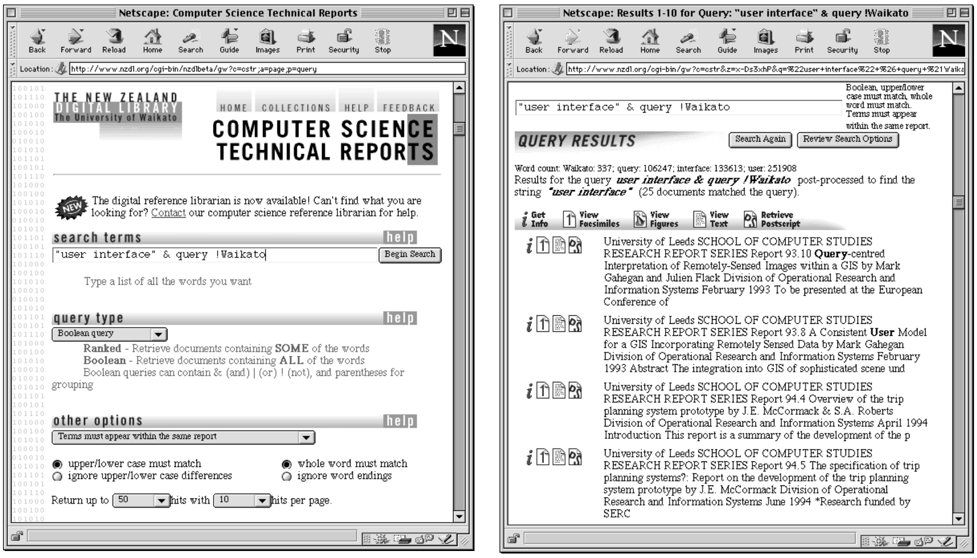


Fig. 1. The current NZDL user interface, showing a query screen (left) and a results screen (right).

An aspect of a study by Hertzum and Frøkjær (1996) 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. Almost ten times the number of syntactically incorrect queries were produced using the textual language than with the Venn interface. Interestingly, browsing of the information structure was significantly faster and produced significantly less errors then either Venn diagrams, the textual query language or a combination of the two.

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.

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 publically available and several more are available only to specific end-user organisations. The freely available collections include an independently published weekly on-line newsletter, 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 40,000 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 complement 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 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.

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

	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 1. Usage of Boolean and ranked 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 2. Distribution of the number of terms in queries for the CSTR collection of the NZDL.

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. The CSTR collection has the most usage data available and so we shall use it as the example collection. The NZDL user interface can be considered over two periods. In the first, Boolean querying was provided by default. In the second ranked querying was the default provision. In each case, users had to explicitly select the alternative. Table 1 summarises the distribution of ranked and Boolean query use. All figures exclude users from within the University of Waikato. We see that two thirds of users use the default query method. Ranked queries are simple to specify as no complex syntax must be used, but it is difficult for users to understand why a particular ranking has been suggested (Clarke et al, 1997). Boolean queries are harder to form, but the justifcation for the provided results is more clear.

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 (1997). Two terms are most commonly used and make up a third of all queries. The average number of queries is 2.5. 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.

From the usage data 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. Noticeably, when users had to explicitly select Boolean querying, about 44% of queries used intersection. The mean number of intersection

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

Number of common terms	Frequency (total = 13650)	%
0	4573	33.5
1	3085	22.6
2	3153	23.1
3	1542	11.3
4	642	4.7
5	300	2.2
6	164	1.2
7	109	0.8
8	41	0.3
9	27	0.2

Table 4. Number of common terms in pairs of consecutive queries for the CSTR collection of the NZDL.

operators in queries where they were used was approximately two. The union operator is used 10 times less frequently than intersection, and complement is used in only 1% of queries. Further usage studies will investigate why users so strongly favour the use of intersection. 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.

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 13650 queries within the CSTR with ranked querying as the default.

The number of common terms in pairs of consecutive queries submitted by individual users is summarised in Table 4. Approximately one third of the queries were completely distinct from the previous one. However, the remaining two thirds had at least one query term in common. The most frequent number of common terms is 2; almost a quarter of all queries. This implies that query refinement is a very common user activity.

4 A Study of Venn Diagram Usage

We carried out a 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 (1976). 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 (1994) could be confirmed.

	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 5. Summary of task ordering for the two groups in the usage study.

4.1 Subjects

A sample of 18 students enrolled in first year undergraduate computer science courses was randomly selected. Subjects were paid a nominal sum when their participation in the study was complete. There were eleven male and seven female subjects. They were initially individually asked some background questions to gain more insight into the experience of the sample. Six of the subjects were enrolled in programmes in the School of Computing and Mathematical Sciences, the remaining 12 being distributed across five other Schools. 16 of the subjects had some experience of using the Internet and had all used one or more WWW based search engine. 14 had English as a first language. 10 subjects had prior experience of Venn diagrams during high school study, six experience elsewhere and two had no prior experience. 12 subjects were aged between 18 and 20 and the remainder were aged between 21 and 29.

We gave no training in how to draw or interpret Venn diagrams to the subjects.

4.2 Method

Each subject completed two tasks which we shall label Task A and Task B. In Task A subjects formed English language expressions to represent their interpretation of given Venn-like diagrams. In Task B subjects formed Venn-like diagrams to represent given English language Boolean expressions.

Subjects were alternately assigned to two groups, based on order of presentation. Group A carried out Task A then Task B. Group B carried out Task B then Task A. This is summarised in Table 5.

This enabled learning effects influenced by order of task presentation to be considered.

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.

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 time taken by subjects to complete each task was 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

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 the response data, labelling terms A, B, C and so on to clarify the Boolean expression structures.

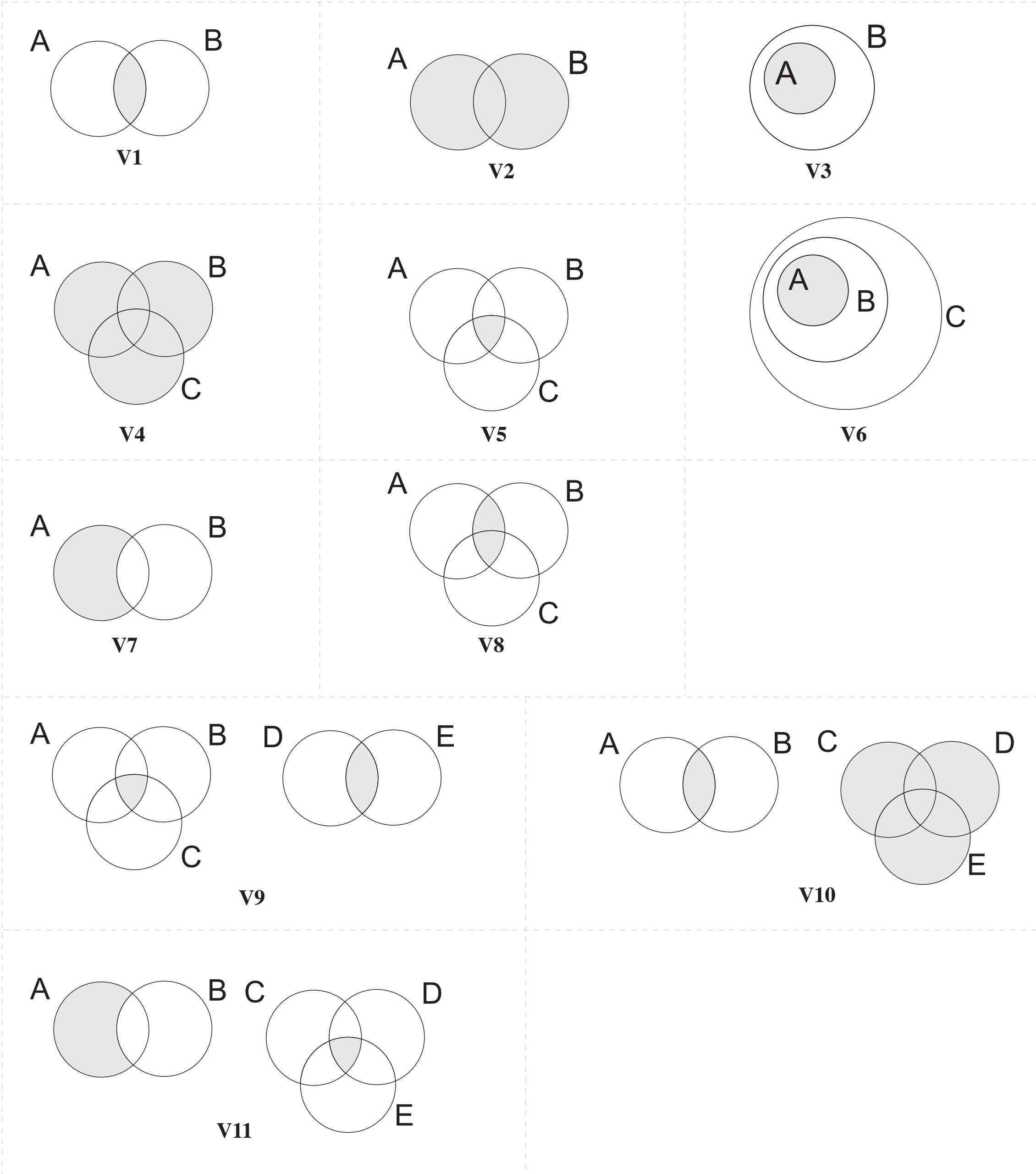


Fig. 2(V1-V11). Venn diagrams presented for interpretation in Task A. The generic term labels A, B, C... were replaced by particular terms in the task.

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 6. Summary of the Boolean expressions presented in Task B. The generic term labels A, B, C... were replaced by particular terms 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 7. Summary of the subjects interpretations of the diagrams presented in Task A. The generic term labels A, B, C... have been substituted for the particular terms used in the task.

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 discernable difference in success in interpretation of intersections. Simple complement (V7) was successfully interpreted across all but one subject, 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

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 8. The subjects' confidence in the correctness of their responses in Tasks A and B.

	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 9. Subject response times in Tasks A and B.

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 complement 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 interepreted non-overlapping sets as unions expressed by OR than intersections expressed by AND.

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.

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

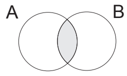
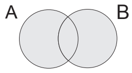
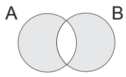
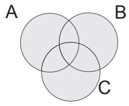
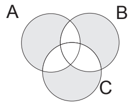
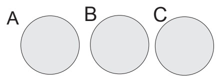
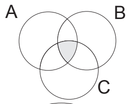
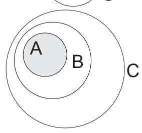
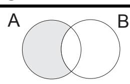
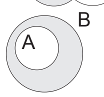
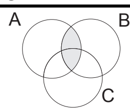
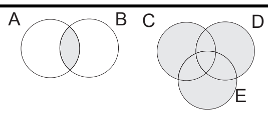
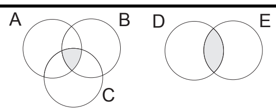
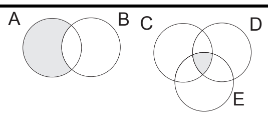
Expression	Responses	Frequency		
		Group A	Group B	Total
E1		8	6	14
	Other	1	3	4
E2		6	4	10
		2	1	3
	Other	1	4	5
E3		5	4	9
		3	0	3
		1	1	2
	Other	0	4	4
E4		4	4	8
		5	0	5
	Other	0	5	5
E5		9	4	13
		0	3	3
	Other	0	2	2
E6		6	3	9
	Other	3	6	9
E7		8	3	11
	Other	1	6	7
E8		8	3	11
	Other	1	6	7
E9		9	2	11
	Other	0	7	7

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

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 (p=0.05) between the confidence of the two groups. This may be due to a true lack of effect of task order on confidence, or to the small sample sizes of each group. We can make no assertions about how forming diagrams prior to interpreting them affects subjects' confidence.

For Task A there is no significant difference (p=0.05) between the time taken by Groups A and B. Table 9 summarises the observations. Again, task order may not have had an effect on subject performance, or the sample size is insufficiently large.

Table 10 summarises the responses for Task B in which subjects formed Venn-like diagrams to respresent 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 complement for simple queries (E1, E4, E5). This is also evident for more complex queries using intersection and complement (E8 and E9). This indicates that the phrasing used to represent intersection and complement was interepreted 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 (1982). 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 (p=0.05). 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. We can make no assertions about how experiencing a diagrammatic notation prior to forming them affects subjects' confidence.

For Task B there is a significant difference (p=0.05) 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. Given the small sample size we must be cautious about this assertion.

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.

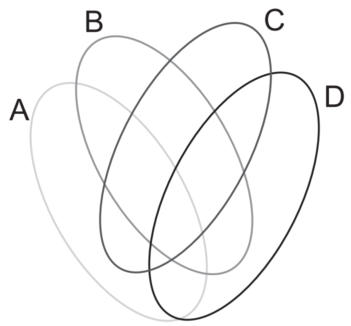


Fig. 3. A graphical arrangement for four overlapping sets.

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 complement 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.

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. Although more subjects interpreted non-overlapping sets as having AND rather than OR relationships there was no clear distinction. The majority of subjects formed non-overlapping sets to represent union operations, particularly when they had already seen examples of their use. Consequently non-overlapping sets may be effective in a user interface when four or more sets are used.

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 (1982) but was responsible for the introduction of errors in query formulation, and so its design must be carefully considered.

5 VQuery

We have developed a graphical query interface application based on Venn diagrams. This system, called VQuery, is an alternative to the textual query interface currently used by the NZDL. Although it is used in conjuction with the NZDL at the moment, it is easily portable to other digital library or search engine applications which operate over the WWW.

VQuery is implemented as a Java application which supports graphical creation and amendment of Boolean queries. It does not deal with presentation of query results – that is left to the particular information source with which it interfaces. Figure 4 shows the VQuery interface window and an

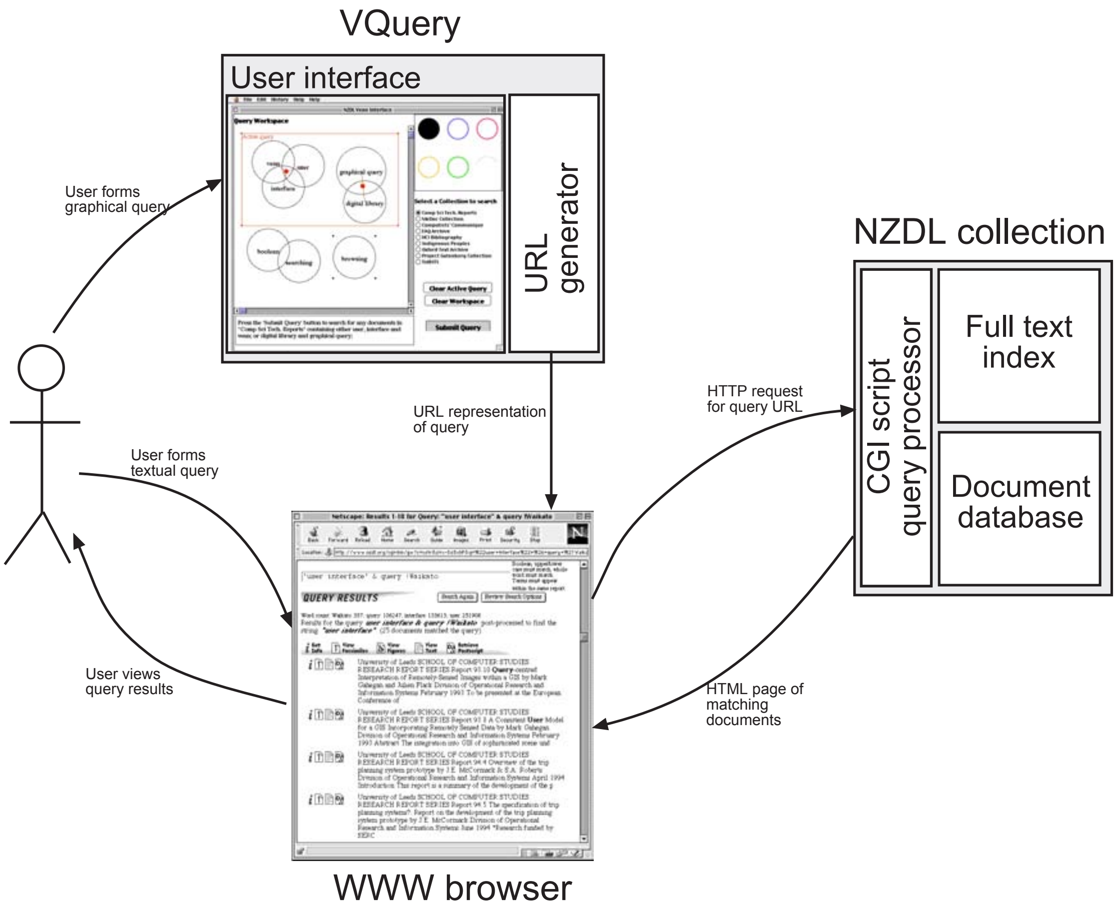


Fig. 4. The integration of VQuery, a WWW browser and the NZDL.

NZDL query result window. Users will form their query in the VQuery window, submit it to the NZDL and the results will appear in the browser window.

Using the conventional WWW interface to the NZDL, the user creates a textual query via HTML forms and submits it. The query terms and options are used to form a URL which invokes a CGI script on the NZDL server with the appropriate parameters. When using VQuery, the user forms a graphical representation of a Boolean query, and submits this to the NZDL server. In fact, at this point a VQuery translator function creates a query URL which is passed to a WWW browser (in this case Netscape). The created URL corresponds directly to that which would be created via the conventional WWW interface to the NZDL server and again contains details of a CGI script to execute and its required parameters. In both cases the NZDL server dynamically creates an HTML representation of the query results which is returned to the WWW browser.

This architecture allows the user to exploit one or both of the available query interfaces and to interleave their use.

5.1 The VQuery User Interface

Figure 5 shows the VQuery interface in detail. It has three main components. To the top left is the *query workspace*. This is a scrollable window in which the user can organise individual query terms to create more complex 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*.

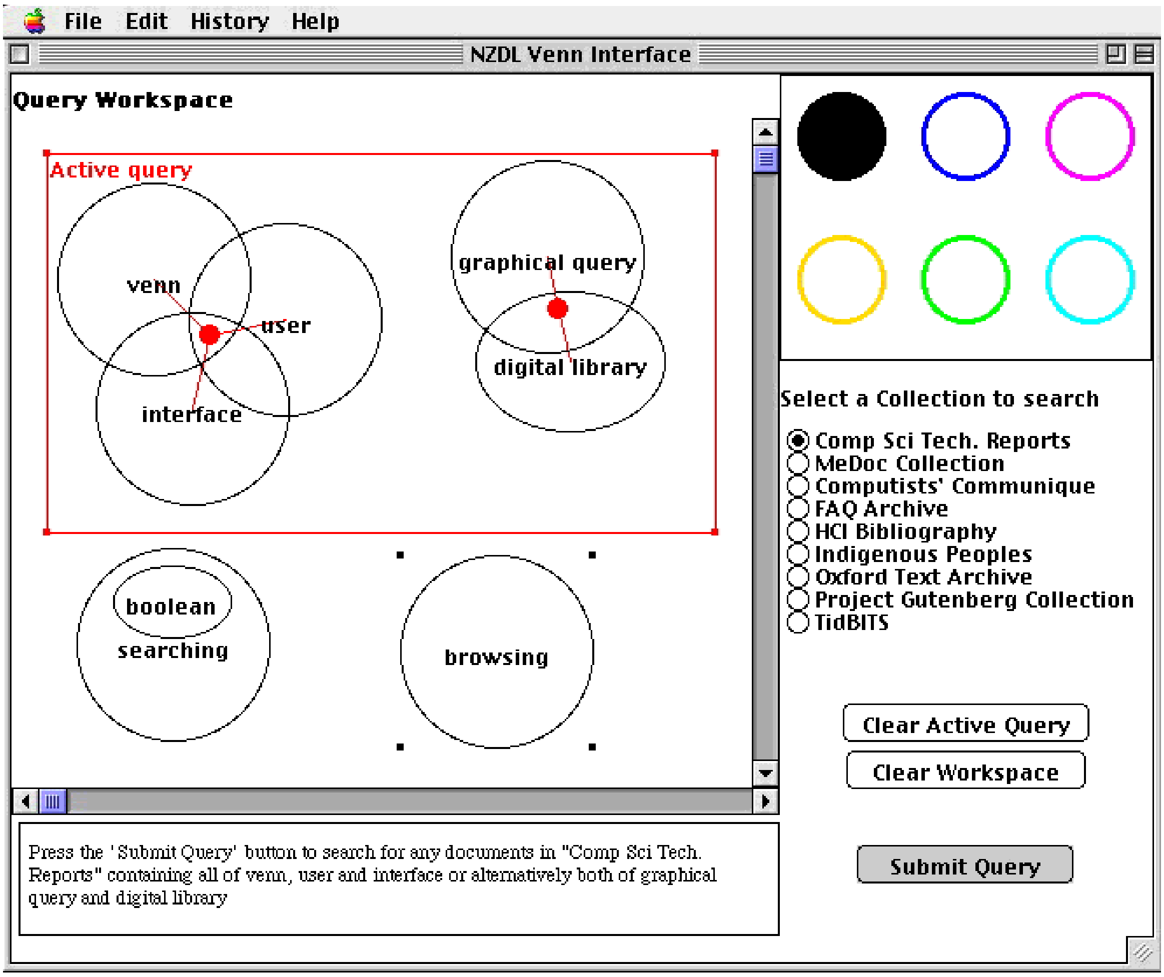


Fig. 5. The VQuery user interface, showing the query workspace, natural language view and control panel.

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. A new term is created by clicking on the background of the workspace, outside of any other terms. The dialogue displayed in Figure 6 allows the user to enter the term or phrase to be represented. The circles can be selected (as 'browsing' 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 and be fully contained within other terms. The number of terms displayed on the workspace at any one time is unlimited. The workspace is divided into two areas; an active query area within the labelled rectangle and a non-active query area outside of it. The active area contains the terms that the user is combining to form a query for submission 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 dimensions of the active query area can be changed by dragging its bounding rectangle, providing a second method for inclusion or exclusion of terms.

The Boolean relationship between terms is described by clicking on the circles and their intersections within the active query area. The way that terms are combined is represented graphically within the workspace using red dots and connecting lines. A click on a red dot will remove it. AND, OR and NOT operations, and complex combinations of them can be graphically represented. Most simply, a single term can be selected. Terms can be combined with a Boolean AND by selecting the intersection of their circles. The OR operation is specified by selection of terms, but not within their area of intersection. Terms that are within the active query area but are not selected are combined with other terms using the NOT operator. Finally, queries with arbitrary numbers of terms using some or all of the operators can be represented. The union operator is used on non-overlapping terms.

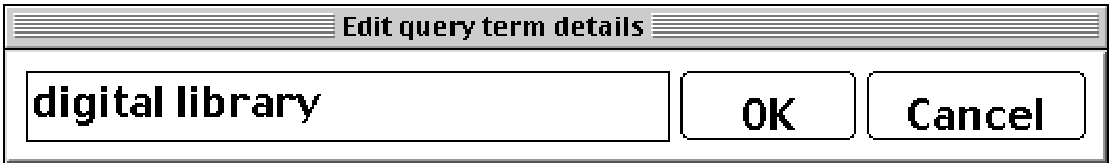


Fig. 6. The VQuery search term input dialogue window.

```
date <Mon Dec  1 16:38:53 NZDT 1997>
collection <HCI Bibliography>
colour <red>
active-query <20, 20> <300> <150>  #origin, width and height
term-count <3>
term <user> <30, 30> <30> <black> <selected>  #term, centre, radius, colour, selection status
term <interface> <270, 140> <50> <green> <unselected>
term <graphical> <100, 180> <30> <green> <unselected>
```

Fig. 7. ASCII file format for saved queries.

The natural language query view presents an English language interpretation of the active query. It is updated immediately whenver 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. Comma separated lists of terms are used for AND and OR operations on multiple terms. Compound queries containing more than one multiple term operation are represented as semi-colon separated lists.

The control panel supports a miscellany of functions. At the top right users can select the default display colour for newly created terms. Both the circle and the text of new terms are displayed in the selected colour. Below the colour selection panel, users can select which information source will be queried. A variety of document collections from the NZDL are currently displayed. When a new information source is selected the natural language view is immediately updated to reflect the change. Below the information sources are two buttons, which act as single action shortcuts for common but tedious functions. The first deletes all terms from the active query area and the second deletes all terms from the entire workspace. The final button in the control panel submits the active query to the NZDL via the user's WWW browser.

The menu bar entries shown at the top of Figure 5 contain functions corresponding to those decribed above for new term creation and deletion, colour selection, collection selection, shortcut actions and query submission. Standard file operations are also provided. The application state can be saved to persistent storage, including workspace content and current collection and colour selections. Previously saved states can be reloaded. The application state is represented in a concise ASCII file as exemplified in Figure 7.

5.2 VQuery implementation and architecture

VQuery has been implemented in Java Version 1.0.2 using Microsoft Visual J++, on an IBM compatible personal computer. It has also been run without any necessary user interface amendments on an Apple Macintosh and in a UNIX environment. A minor reconfiguration must be carried out to indicate the location of the WWW browser with which it communicates when it is transferred to a new environment.

The VQuery software architecture is flexible, adopting a Model–View–Controller approach (Burbeck, 1987). This approach is shown in Figure 8. An underlying data structure *models* the list of terms created by the user, and the semantics of the application. This is necessarily abstract so that it is sufficiently independent of particular user interface designs. The *View* component presents the current state of the model to the user on the display. The *Controller* component deals with user input. The View and Controller components are closely linked in a direct manipulation interface such as VQuery. In fact, VQuery uses three View–Controller pairs which integrate with the single data model. The first is the query workspace which displays created query terms and indicates term selection. This also supports user input. Circles are moved, resized, selected and so on. The second pairing is the natural language query output. Essentially this is only a view, with an inactive

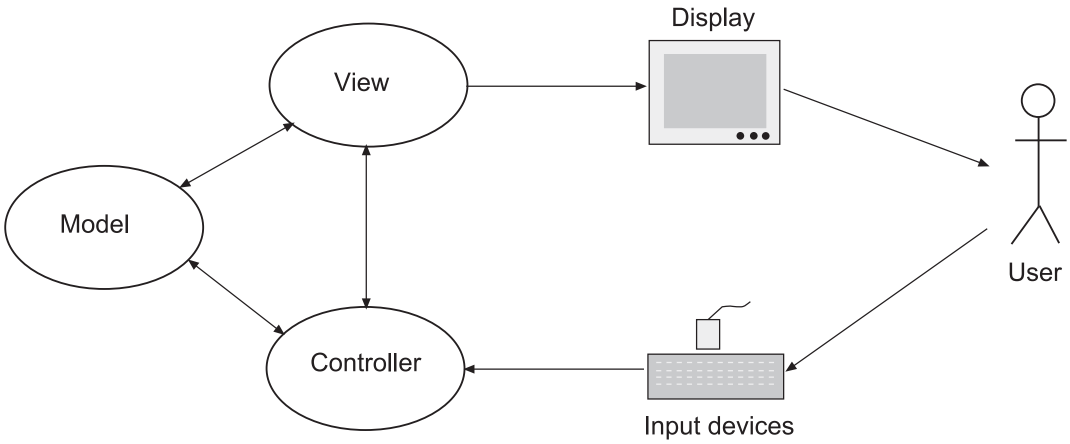


Fig. 8. The Model–View–Controller architecture used in VQuery.

controller component. Third is the URL corresponding to the active query which will be sent to the NZDL server. Again this has an inactive controller component, and is, in fact, hidden from the user.

This architecture supports provision and investigation of extended or alternative interfaces. Other View–Controller pairings may be integrated with the model.

5.3 Discussion

We have taken the graphical representation and English interpretations that we found to be effective and embedded them as closely as possible in VQuery.

An evident distinction is the method used for indicating the query within the diagram. Ideally we would like to provide shading to indicate the selected expression. However, this has proven troublesome in the version of Java that we are using.

There are some Boolean expressions that can not be represented using VQuery, mainly because it does not support selection of the universal set. Consequently some complement expressions such as those involving the complement of single sets, or the complement of compound expressions are unavailable to users. Examples of these are shown in Figure 9. This means that users are unable to request ‘documents that do not contain A’, or ‘documents that do not contain both of A and B’. Although less than 2% of Boolean queries in the NZDL use the complement operator it is important that its provision in VQuery is complete and consistent. We must consider how best to support this in the VQuery interface.

6 Future work

In the next stage of this work we will refine VQuery to support more effective representation of selected areas of the Venn diagrams, and extend it so that the universal set may be selected.

We will then carry out a comparative study of VQuery and the current NZDL textual interface. We are interested in the comparative accuracy and speed of the two interfaces in allowing users to form queries which match their information seeking needs. Suitable sample queries will be identified to reflect the patterns of usage that are evident from analysis of NZDL usage. This will enable comparision with previous studies and allows us to consider the utility of non-overlapping set notation, flexible diagram layout, the workspace for storage and reuse of query terms and natural language feedback. These are attributes which have not been present in previous systems.

In the longer term we plan to extend the interface in several ways. First we will allow users to specify additional NZDL search options such as case-folding, stemming, term weighting and search granularity. Term weighting and search granularity in particular may lend themselves to graphical representation.

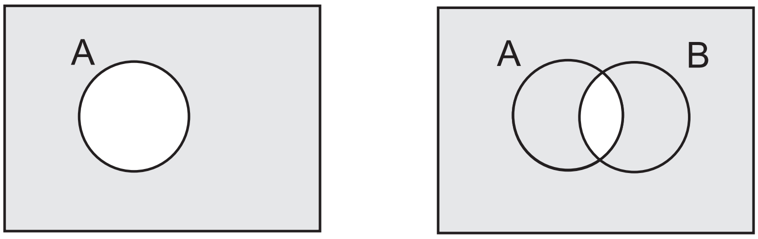


Fig. 9. Set relationships involving the universal set which can not be expressed in VQuery.

Shneiderman (1994) has argued for dynamic query interfaces which reflect changes in query results as the query itself is refined. We have implemented a prototype extension of VQuery which shows a display of document titles matching the selected query which is updated when the query changes. Users can get an idea of how many and what sort of documents will match their query as they refine it, without they delay associated with submitting a query to the NZDL. This capability will be fully integrated with the revised version of VQuery.

The collaborative nature of information browsing and retrieval has been highlighted by Twidale and Nichols (1996). In the NZDL we would like to support collaboration between users, but also between users and the NZDL librarian who deals with users requests and offers advice on its use. We have developed a prototype system Collaborative VQuery 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 chat windows. This system is currently implemented in GroupKit, a groupware development toolkit (Roseman and Greenberg, 1996). We will implement this system in Java for integration with the single user VQuery and carry out user evaluation.

References

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.

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.

Burbeck, S.: A*pplications Programming in Smalltalk-80: How to Use Model-View-Controller (MVC).* Softsmarts Inc., 1987.

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.

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.

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.

Halpin, T.A.: Venn Diagrams and SQL Queries. *The Australian Computer Journal*, 21(1):27-32, 1989.

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.

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.

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.

Roseman, M., Greenberg, S.: Building Real Time Groupware with Groupkit, a Groupware Toolkit. *ACM Transactions on Computer-Human Interaction*, 3(1):1-37, 1996.

Shneiderman, B.: Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6):70-77, 1994.

Thomas, J.C.: Quantifiers and Question Asking. IBM Research Report 7021976, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 1976.

Twidale, M.B., Nichols, D.M.: Collaborative Browsing and Visualization of the Search Process. *Aslib Proceedings*, 48(7/8):177-182, 1996.

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.

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.

Witten, I.H., Cunningham S.J. , Apperley M.D.: The New Zealand Digital Library Project. *New Zealand Libraries* 48(8):146-152, 1996.

Zloof, M.M.: Query By Example. *AFIPS Conference Proceedings*, 44:431-437, 1975.