

Browsing and Querying in Online Documentation: A Study of User Interfaces and the Interaction Process

Morten Hertzum and Erik Frøkjær

Department of Computing Science, University of Copenhagen

A user interface study concerning the usage effectiveness of selected retrieval modes was conducted using an experimental text retrieval system, TeSS, giving access to online documentation of certain programming tools. Four modes of TeSS were compared: (1) browsing, (2) conventional boolean retrieval, (3) boolean retrieval based on Venn diagrams, and (4) these three combined. Further, the modes of TeSS were compared to the use of printed manuals. The subjects observed were 87 computer science students who solved a number of information retrieval tasks in an area of computing new to them. In the experiment the use of printed manuals is faster and provides answers of higher quality than any of the electronic modes. Therefore, claims about the effectiveness of computer-based text retrieval have to be wary in situations where printed manuals are manageable to the users. Among the modes of TeSS, browsing is the fastest and the one causing fewest operational errors. On the same two variables, time and operational errors, the Venn diagram mode performs better than conventional boolean retrieval. The combined mode scores worst on the objective performance measures; nonetheless nearly all subjects prefer this mode.

Concerning the interaction process, the subjects tend to manage the complexities of the information retrieval tasks by issuing series of simple commands and exploiting the interactive capabilities of TeSS. To characterise the dynamics of the interaction process two concepts are introduced: threads and sequences of tactics. Threads in a query sequence describe the continuity during retrieval. Sequences of tactics concern the combined mode and describe how different retrieval modes succeed each other as the retrieval process evolves.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Query Formulation and Search Process; H.5.2 [**User Interfaces**]: Evaluation and Interaction Styles

General terms: Experimentation, Human Factors, Performance

Additional Keywords and Phrases: Usage effectiveness, interaction process, online manuals, documentation

1. Introduction

Users of text retrieval systems often experience difficulties when trying to phrase their information need in the retrieval language of the system. One reason for this is that it is inherently difficult to give a precise and differentiating description of something one lacks knowledge about [Belkin et al. 1982; Blair & Maron 1985; Bates 1986a]. Another reason is that most text retrieval systems offer the user little support in initiating and focusing his search. Taking the latter problem as the point of departure we have developed an experimental text retrieval system, TeSS, which provides interactive access to documentation of a certain set of programming tools through four different

retrieval modes. They include browsing, conventional boolean retrieval, querying based on Venn diagrams, and a combined mode including both browsing and querying. Retrieval by use of hard copy manuals of the programming tools is included as a fifth retrieval mode. The study comprises an experimental evaluation of the usage effectiveness of the retrieval modes and emphasises the interactive aspects by closely studying how the retrieval processes evolve during the subjects' solution of their tasks.

Three working hypotheses guided the design of this experiment. The first is that some of the essential problems encountered with boolean retrieval can be dealt with in the user interface. They might not be unavoidable properties originating from the need for exact match, though this is the typical point of view, see for example Belkin & Croft [1987] and Radecki [1988]. Our first hypothesis is studied by comparing conventional boolean retrieval with querying based on Venn diagrams. The second working hypothesis is that browsing and querying supplement, rather than duplicate, each other. This assumption, also expressed by for instance Marchionini & Shneiderman [1988] and Jerke et al. [1990], is studied by examining whether the combination of browsing and querying is superior to each one in isolation. The third working hypothesis originates from our reaction to the general belief in the superiority of electronically stored information over 'old-fashioned' hard copies. Consequently, the modes of TeSS are not only compared to each other but also to the use of printed manuals. This is also intended to provide a baseline for the experiment. Thus, the purpose of the experiment is threefold:

- To compare a conventional interface to boolean retrieval with one based on Venn diagrams.
- To compare browsing and querying, when used separately and in combination.
- To compare computer-supported text retrieval with the use of hard copy manuals.

The experiment concerns a relatively small body of related texts compared to some of the very large, heterogeneous corpora important to the field of information retrieval. However, people and organisations make use of quite a number of small bodies of texts, making this an important subarea in its own right.

The next section is a description of TeSS. The third section is about the method and design of the experiment. In the fourth section the results are presented and discussed.

2. The text retrieval system TeSS

TeSS is a text retrieval system developed specifically for this experiment and currently gives access to online documentation relevant when developing graphical user interfaces in the X Window System. It runs on a network of HP9000/300 workstations and is implemented using the object-oriented programming language Beta, Athena Widgets, C, and Ingres. The text database comprises

3 Mb of text and consists of the C language interface manuals for Xlib, X Toolkit Intrinsics, and Athena Widget Set.

2.1 Browsing

Browsing is a retrieval process where the users navigate through the text database by following links from one piece of text to the next aiming to utilise two human capabilities [Bates 1986b]: (1) The greater ability to recognise what is wanted over being able to describe it; and (2) the ability to skim or perceive at a glance. This allows users to evaluate rapidly rather large amounts of text and determine what is useful.

Among the various approaches to browsing, TeSS includes one which emphasises the structure of the documentation. In TeSS, browsing is limited to the table of contents while the text itself is presented in separate windows. By this separation the structure is continuously apparent to the user, reducing the well-known risk of ‘getting lost in hyperspace’. Browsing takes two forms: direct access through *find string*, and hierarchy expansion through *more detail* and *less detail*, see figure 2.1. *Find string* highlights the next occurrence of a string and can thus be used to skip through the table of contents, by jumping from one occurrence of the string to the next. *More detail* unfolds an entry in the table of contents by adding the next, more detailed, level of subheadings. *Less detail* is the inverse of *more detail*.

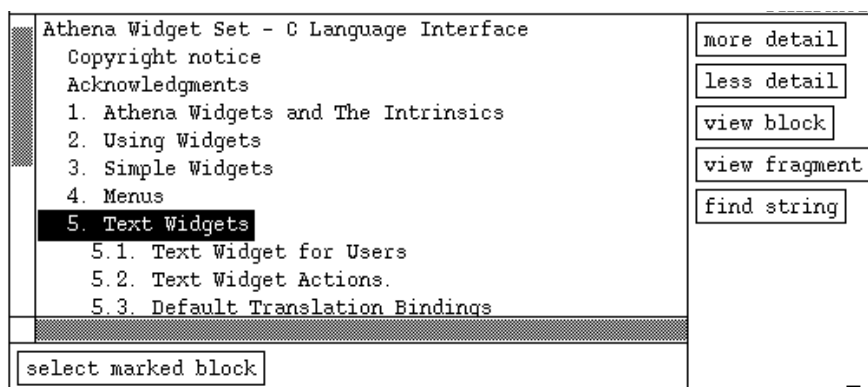


Figure 2.1. Browsing subwindow.

The actual text is accessed by opening so-called Text Viewers. A Text Viewer can be opened at any level in the text hierarchy by either *view block* or *view fragment*. *View block* opens a Text Viewer with the whole subhierarchy of text belonging to the chosen heading. In this way the user can retrieve, for example, an entire chapter for closer inspection. *View fragment*, on the other hand, opens a Text Viewer which presents the text belonging to a heading with exclusion of text belonging to its subheadings. *View fragment* is useful when inspecting hits from a query because the texts presented are reduced to the smallest parts that match the query.

2.2 Querying

Querying is a retrieval process in which the user's initial query leads to a series of reformulations of it. With each new formulation, the user may identify useful information and, thus, his information need is satisfied by a series of references and texts presented by the system at each state of the process. The user's information need is not static, it evolves as he encounters new pieces of information and thereby gets new ideas and directions to follow [Bates 1989].

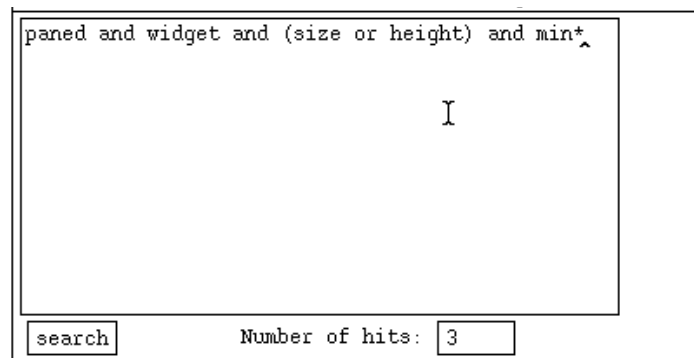
TeSS includes two techniques for querying. The first is conventional boolean retrieval where queries are logical expressions built of query terms, ANDs, ORs, NOTs, parentheses, and wildcards, see figure 2.2 (a). The second is also boolean retrieval but represented graphically by a Venn diagram.

This idea is not new. It has for example been suggested by Smith [1976] as a paper-and-pencil tool in client-searcher communication and implemented by Michard [1982] to avoid the explicit use of boolean queries. Spoerri [1994] has extended the idea and created an integrated visual query language and information presentation tool where the user can explore relationships among any number of concepts in a flexible, interactive way.

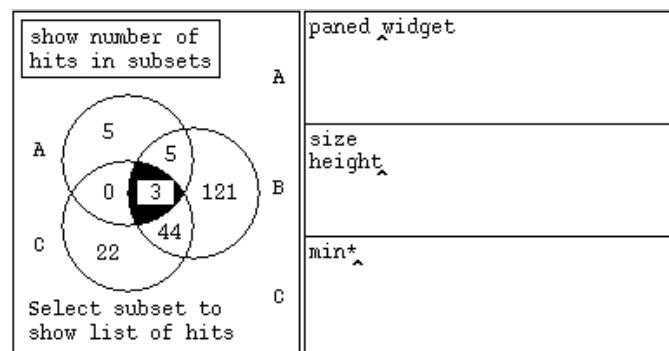
In TeSS the Venn diagram consists of three overlapping circles each with an accompanying input field, see figure 2.2 (b). Searching is performed by: (1) entering the query terms in the input fields; (2) clicking on *show number of hits in subsets*, which causes the number of hits in each of the seven subsets of the Venn diagram to appear; and (3) clicking in one of the subsets to see the accompanying hit list with references to the texts that match the query (cf. subwindow 4 in figure 2.3). The user may bypass step 2, causing it to be done automatically when he executes step 3. A query term is either a single word or a phrase, and it occupies one line in an input field. Referring to figure 2.2 (b) where the user wants to know how, if at all, a paned widget can be defined to have a certain minimum height, the query terms are 'size', 'height', 'min*', and the phrase 'paned widget'. A query in the Venn diagram interface corresponds to a logical query expression generated by placing:

- AND between words on the same line in an input field, i.e. between words in a phrase.
Example: paned AND widget.
- OR between the lines in an input field, i.e. between terms.
Example: size OR height.
- AND between the input fields that are part of the selected subset.
Example: paned AND widget AND (size OR height) AND min*.
- AND NOT in front of the input fields that are not selected.
In the example no input fields remain, so the negation is not used.

Note, that phrases are actually quasi-phrases implemented by ordinary ANDs and, thus, often retrieve more texts than those in which the words make up the indicated phrase.



(a)



(b)

Figure 2.2. Querying subwindows. (a) Querying by logical expressions. (b) Querying by Venn diagrams. The queries in (a) and (b) are identical. But if the user, in (b), had clicked in the intersection of A and C only (the subset returning zero hits), then the corresponding logical query expression would have been 'paned and widget and not (size or height) and min*'.

The Venn diagram has several seemingly attractive qualities:

- It replaces boolean operators and parentheses with a, supposedly, more immediately understandable graphical image of intersecting sets.
- It offers a prestructuring of the query that forces the user to phrase his information need as the intersection of one, two, or three facets, each defined by one or more terms. The idea is to invite the user to address the facets one by one and develop the query incrementally.

- It gives detailed feedback about the distribution of the hits. It is, for instance, directly visible when one facet returns very few or very many hits. This feedback can direct the reformulation of the query by pointing to the part that needs adjustment.

These qualities support the user in bridging the gap between his vision about the information needed and his formulation of a proper query. Further, the Venn diagram interface provides a kind of term weighting and hit ranking, obtainable for example by first examining the hits from the intersection of all three facets and then the hits from the intersection of the two presumably most important facets. To some extent, this meets two of the major objections raised against boolean retrieval, namely the lack of term weighting and hit ranking, see for example Belkin & Croft [1987] or Radecki [1988].

2.3 The Control Window

The user interface of TeSS was designed to be used by programmers with a need for consulting documentation while coding. Thus, TeSS leaves part of the screen available for program editing. The user interface includes the Control Window and any number of Text Viewers. The Control Window consists of four subwindows, see figure 2.3, and is designed to display, at all times, all functions available to the user and all information defining the current state of the search. The top subwindow is the table of contents giving an overview of the texts available; the second subwindow shows the texts currently selected for querying; the third subwindow allows querying by either logical expressions or Venn diagrams; and the fourth subwindow is the hit list. To allow browsing at any stage during the search, the browsing facilities are replicated in the table of contents, the selection list, and the hit list.

During the design of TeSS, several ways of combining some of the subwindows were discussed, see Broløs et al. [1993]. However, the possibilities considered either introduced complicated codings of the entries in the table of contents or overlaid important information with other information. The chosen design was tested through a couple of informal think-aloud experiments. After an initial overwhelming impression of the interface, no one seemed to have problems understanding and using it.

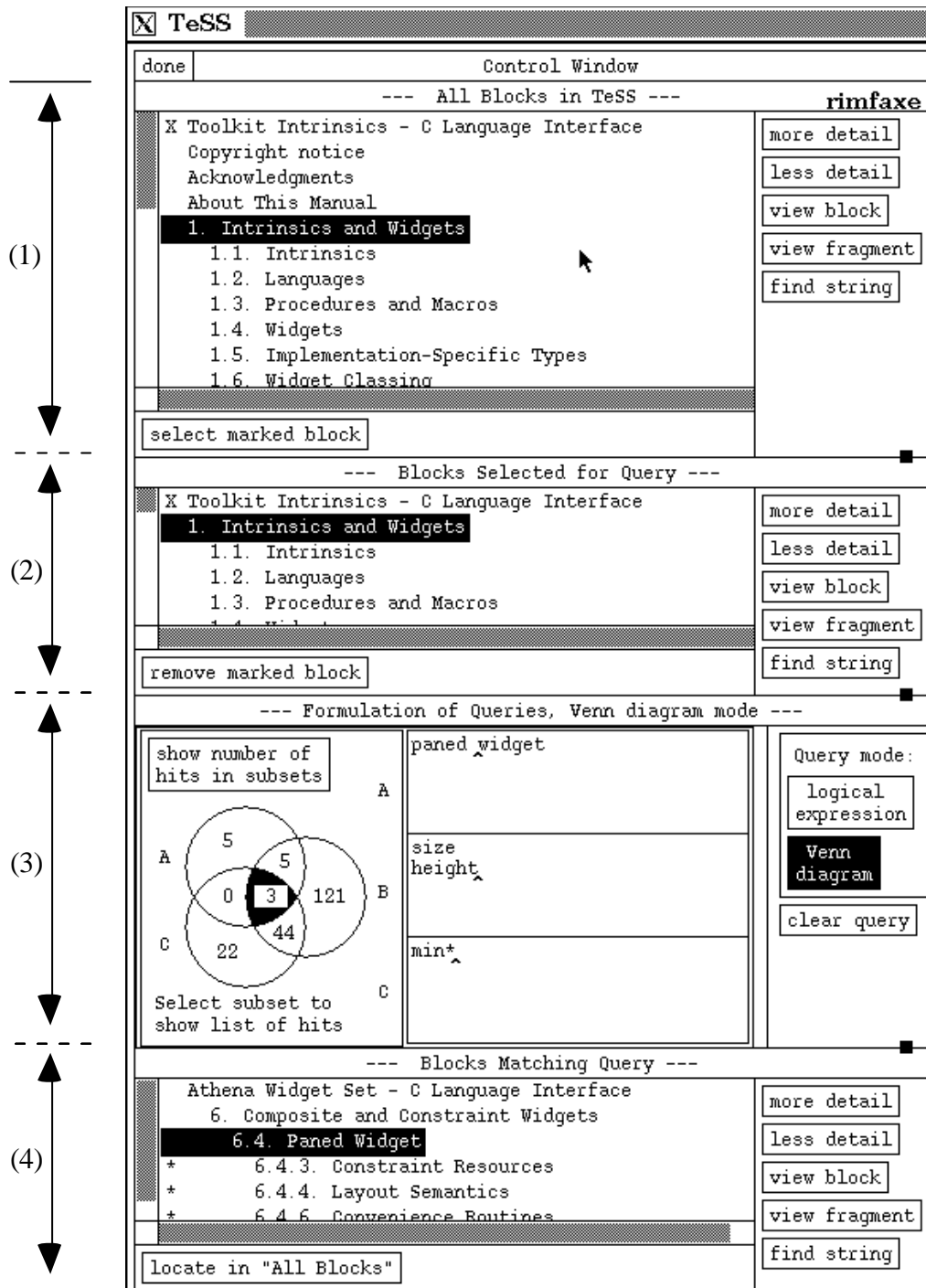


Figure 2.3. TeSS Control Window consisting of four subwindows: (1) the table of contents, (2) the selection list, (3) the query subwindow (currently in Venn diagram mode), and (4) the hit list. The relative size of the subwindows can be adjusted at any time.

2.4 The five retrieval modes

The five retrieval modes in the experiment are:

- **BROWSE.** A mode of TeSS offering browsing only. The table of contents subwindow is active while the three other subwindows are inactive.
- **LOGICAL.** A mode of TeSS offering querying by logical expressions only. The table of contents is frozen at chapter level, and chapters can be added to and removed from the selection list. The browsing facilities are disabled in all subwindows.
- **VENN.** A mode of TeSS offering querying by Venn diagrams only. This mode is identical to **LOGICAL** except for the way queries are formulated.
- **ALL.** The whole of TeSS offering the combination of **BROWSE**, **LOGICAL**, and **VENN**.
- **PAPER.** In this mode searching is done in printed manuals, i.e. independently of TeSS.

2.5 Penalties

Three aspects of the retrieval modes are important to the experiment because they penalise one or the other mode:

- The capabilities for text presentation are limited in the current version of TeSS. The opening of a Text Viewer is a rather slow operation taking 5-15 seconds. Further, the number of Text Viewers a user can have open simultaneously is limited to 3-5 when TeSS is running on a heavily used machine. It is assumed that this penalises browsing since users who are browsing, probably, work more with the text itself.
- Text presented in Text Viewers maintains the original formatting, but is 'dead', i.e. there is no highlighting of query terms and no means for manipulating the text other than scrolling. This can make it difficult to find the potentially relevant parts if the text is large, but normally hits from queries are rather small, from ten lines to two pages.
- Inadvertently, the hard copy of the Athena Widget manual contained no index. Obviously, this has impeded the use of the manual. The only structured access path was the table of contents.

3. The method of the experiment

The experiment was conducted at the University of Copenhagen, the subjects being computer science students solving a three week, grade-giving project in user interface design. The tasks to be solved during the experiment were also part of their projects, so what the experiment added was the data collection and a strict procedure. While the project was a mandatory part of the students' education, participation in the experiment by allowing the data collection to take place was voluntary and anonymous.

3.1 Experimental Subjects

The subjects were 87 students in their third year of a bachelor degree in computer science. The subjects' mean age was 23.9 years (80% were between 20 and 26 years of age), and nine out of

ten were male. All subjects were practised computer users, and most of them had some information retrieval experience from online help systems in general programming languages and from public libraries. In particular, some subjects were familiar with the online documentation available on Unix, the ‘man pages’. On the other hand, the subjects had no prior knowledge of the programming tools on which the tasks were based and very limited knowledge of the object-oriented architecture of the tools and programming environment. Thus, the subjects were first-time users of TeSS and of the programming tools, but had a solid background for quickly getting acquainted with both.

3.2 Experimental tasks and their evaluation

The tasks concerned the development of graphical user interfaces, and the subjects were asked whether and how certain interface properties could be achieved. Some tasks were formulated in the context of the X Window System in general; others took the user interface of TeSS as their point of departure. To answer the tasks the subjects had to identify and combine the relevant user interface objects, e.g. widgets, methods, and resources. As the subjects were unfamiliar with the X Window System, the tasks involved a substantial element of learning in addition to the need for retrieving specific facts.

The experiment included three sets of tasks: (1) Two training tasks intended to make the subjects familiar with TeSS. (2) Twenty information retrieval tasks which were used to study the subjects’ performance in terms of a number of objective performance measures. During this set of tasks the experimental design determined the retrieval mode available to the subject at any time. (3) Three implementation tasks which included the implementation of three minor modifications of the user interface of TeSS. To retrieve the information needed for implementing these modifications, the subjects were free to choose among the four modes of TeSS, making it possible to identify which modes were preferred.

The tasks were only partly formulated using the terminology of the X Window System and had to be interpreted before the search process could be focused. A further difficulty stemmed from the tasks being stated in Danish, the subjects’ native language, while the language of the X Window System manuals is English. Appendix A includes an English translation of the 25 tasks. Here, two examples are given:

Task 5. Radio buttons are used in situations where exactly one option must be chosen from a group of options. Which widget class is used to implement radio buttons?

Task 6. TeSS is divided into four subwindows, so-called panes, separated by thin horizontal lines with square handles. To distinguish these lines more clearly from other horizontal lines in the interface, they could be made thicker. Find out how, if at all, this can be done.

The subjects’ answers were evaluated in terms of five performance measures: (1) the time used, (2) the quality of the answer, (3) the number of commands executed, (4) the number of Text

Viewers opened, and (5) the number of operational errors made, an error being a user activity violating the syntax of the user interface. The quality of the answers was assessed by one of the researchers and expressed by a grade on a five point scale, see figure 3.1. As an example, a medium and a high quality answer to task 5 must include an identification of toggle widgets as the relevant widget class. A brilliant answer also explains the use of radio groups to cluster the toggle widgets.

Grade	Mnemonic	Description
1	very low	Failure, a completely wrong answer
2	low	Inadequate or partially wrong answer
3	medium	Reasonable but incomplete answer
4	high	Good and adequate answer
5	very high	Brilliant answer

Figure 3.1. The five point scale used to grade the tasks.

3.3 Experimental Procedure

The experiment was explained at a lecture, after which each student drew a userid and a password for a special TeSS account. However, a student’s TeSS account was not opened until he had decided whether or not he would participate in the experiment. After this introduction, the subjects had ten days to complete the tasks. Questions and the like could be reported through email or brought up at the ordinary, weekly classes, but a precondition in the design of the experiment was that the subjects could proceed on their own.

The subjects received a manual of TeSS and a two-page walk-up-and-use introduction. The system itself was available on the terminals allocated to third year students. These terminals are located in one room to which students have access 24 hours a day. The library where the manual searching took place was open three hours a day; thus, some planning was required to reach these tasks during opening hours. An undisturbed part of the library was reserved for the subjects, and one of the researchers was constantly present to hand out tasks and receive solutions. On entering the library, the subjects received hard copies of the three manuals, a sheet with the proper task, and a log sheet. The log sheets which had fields for starting time, finishing time, and solution were filled in by the subjects.

All subjects solved the tasks in the same sequence and all subjects performed using all retrieval modes, that is the experiment employed a within-groups design. To avoid order effects, the subjects were exposed to the retrieval modes in a systematically varied order, see figure 3.2. The 20 information retrieval tasks were clustered into five blocks. The first block was solved with one of the five retrieval modes, the second block with one of the remaining four retrieval modes. Thus the permutations of the modes on the two first blocks divided the subjects into 20 groups. The

number of subjects did not allow all 5! sequences of the five modes to be included, and the 20 groups were not divided further. Rather, for each group the order of the three remaining modes was decided by picking one of the six possible permutations.

Tasks (in time sequence)		T ₁ ... T ₄		T ₅ ... T ₈		T ₉ ... T ₁₂		T ₁₃ ... T ₁₆		T ₁₇ ..T ₂₀	
		Block ₁		Block ₂		Block ₃		Block ₄		Block ₅	
Subjects	Student ₁	Group ₁	Mode ₁	Mode ₂	Mode _{p(3)}	Mode _{p(4)}	Mode _{p(5)}				
	⋮		⋮	⋮	⋮	⋮	⋮				
	⋮		⋮	⋮	⋮	⋮	⋮				
	⋮		⋮	⋮	⋮	⋮	⋮				
	⋮		⋮	⋮	⋮	⋮	⋮				
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	⋮	⋮	⋮	⋮	⋮	⋮					
	Student ₈₇	Group ₂₀	Mode ₅	Mode ₄	Mode _{p(1)}	Mode _{p(2)}	Mode _{p(3)}				

Figure 3.2. The experimental design.

3.4 Data collection and the Task Handling Module

The data collection includes a detailed log of the subjects' interaction with TeSS, log sheets from the manual searching, and the subjects' response to a questionnaire about their age, sex, prior knowledge, and subjective assessment of TeSS. The interaction log gives a time-stamped account of the commands executed by each subject. It also includes task demarcation and solutions reached, both obtained from a separate module governing the access to TeSS, the Task Handling Module.

The Task Handling Module makes it possible to let the subjects work unsupervised while at the same time enforcing the experimental procedure. This module handles login, presents the tasks to the subject one at a time, gives access to the retrieval mode to be used by that subject when solving that particular task, and records his solutions. The subjects can quit the Task Handling Module between tasks or in the middle of a task. In the latter case the solution of the task is broken into sessions.

3.5 Data analysis

The data analysis should ensure that the objective performance measures reflect differences between the retrieval modes only. Due to the within-groups design, differences between subjects have to be accounted for; otherwise the dispersion of the subjects' mean performance, for example between consistently fast and consistently slow subjects, might blur differences pertaining to the retrieval modes. Following Dix et al. [1993, p. 384] this is avoided by aligning each subject's

performance with the overall mean performance. Differences between the tasks must also be taken into account because not all tasks have been solved exactly equally many times with each retrieval mode. For each performance measure, v , subject $_i$'s performance on task $_j$ is adjusted as follows:

$$v'_{ij} = v_{ij} - (ms_i - m) - (mt_j - m)$$

where v'_{ij} is the adjusted performance, v_{ij} is the measured performance, $(ms_i - m)$ is the difference between the subject mean and the overall mean, and $(mt_j - m)$ is the difference between the task mean and the overall mean.

Some answers to individual tasks were left out from the data analysis because they did not meet the experimental requirements. This screening of the answers was highly important because the subjects had worked unsupervised. The 87 subjects received 20 information retrieval tasks each, giving a potential total of 1740 answers. However, 113 answers were nonexistent; 19 were excluded because they contained a more than one hour long period with no logged user activity; 17 were excluded due to technical problems with TeSS; 14 were excluded because it was impossible to judge the quality of the answer; and 2 were excluded because they were solved poorly in less than two minutes, i.e. without any attempt to reach a solution. Finally, 4 subjects were excluded because they clearly did not take the experiment seriously. Thus, 11% of the answers were nonexistent or excluded. The analysis is based on the remaining 1555 answers comprising 648 hours of work performed by 83 subjects.

One performance measure was nearly normally distributed and analysed with the standard method for analysis of variance, ANOVA; subsequently t-tests were used to make pairwise comparisons of the retrieval modes. The other performance measures were not nearly normally distributed and were, therefore, analysed with robust rank-sum tests: Kruskal-Wallis test (a rank sum version of ANOVA) and Wilcoxon test for the pairwise comparisons.

4. Results and discussion

This section first presents the results of the statistical analysis of the performance measures. The subsequent discussion centres around the working hypotheses: Printed manuals and online documentation are compared; the two kinds of querying are compared; browsing is analysed; and the combination of browsing and querying is analysed.

4.1 Performance of the computer-based retrieval modes

The mean performance with the four modes of TeSS is summarised in figure 4.1. The null-hypothesis that there is no difference between the retrieval modes is rejected for time at the $p < .0001$ level ($c^2=21.16$, $df=3$), for commands at the $p < .0003$ level ($c^2=18.56$, $df=3$), and for operational errors at any computable p-level ($c^2=70.83$, $df=3$). There are no significant differences

for grade ($p=.31$ for the null-hypothesis, $F=1.19$, $df=3$) and for viewers ($p=.099$ for the null-hypothesis, $c^2=6.28$, $df=3$).

	BROWSE	LOGICAL	VENN	ALL
time	23.08	30.02	25.53	30.77
grade	2.73	2.89	2.76	2.83
commands	33.81	35.21	35.11	43.95
viewers	6.09	5.37	6.31	5.05
operational errors	0.10	0.97	0.48	0.39

Figure 4.1. Mean performance to solve a task, in terms of time used (minutes), grade for quality of solution (lowest: 1, highest: 5), number of commands executed, number of Text Viewers opened, and number of operational errors made.

For the performance measures capturing significant differences it follows from pairwise comparisons of the retrieval modes that ('<' indicates 'less than', '<<' indicates 'significantly less than'):

time: BROWSE << VENN << LOGICAL < ALL
 commands: BROWSE < VENN < LOGICAL << ALL
 operational errors: BROWSE << ALL < VENN << LOGICAL

BROWSE is significantly faster than VENN (null-hypothesis rejected at the $p<.02$ level, $Z=-2.07$), and VENN is significantly faster than LOGICAL (null-hypothesis rejected at the $p<.01$ level, $Z=-2.28$). By implication, VENN is significantly faster than ALL, BROWSE is significantly faster than LOGICAL, etc. Despite these significant differences the relative variation between the modes is moderate, i.e. the slowest retrieval mode is 33% slower than the fastest. ALL causes the execution of significantly more commands than LOGICAL (null-hypothesis rejected at the $p<.0004$ level, $Z=-3.36$) while there are no significant differences between BROWSE, VENN, and LOGICAL. Again the relative variation is moderate, i.e. 30%. As for operational errors, BROWSE causes significantly fewer operational errors than ALL (null-hypothesis rejected at the $p<.001$ level, $Z=-3.02$) and VENN causes significantly fewer operational errors than LOGICAL (null-hypothesis rejected at the $p<.0002$ level, $Z=-3.61$). Here, the differences are not only statistically significant, but also large in magnitude. LOGICAL causes nearly ten times as many operational errors as BROWSE, and twice as many as VENN, the second most error prone retrieval mode.

Naïvely, one could suppose that the distribution of grades cluster around the mean, but as figure 4.2 shows 28-32% of the solutions are failures, i.e. receive the lowest grade. There are large individual differences in the quality of the results, but all subjects make at least one failure. It has not

been possible to identify a common cause for the failures. Borgman [1986a] also finds a failure ratio of more than 25% in her experiment and is concerned about the consequences of this finding. However, a failure ratio of this magnitude might simply be what one must expect from controlled experiments where the answers reached are not applied and tested in practice by the subjects.

The performance measures can be split into search result measures, i.e. grade, and search process measures, i.e. all but grade. Significant differences between the four modes of TeSS are found only in the measures concerning the search process. It seems that the subjects compensate for the strengths and weaknesses of each retrieval mode by adjusting their own effort.

	BROWSE	LOGICAL	VENN	ALL
failure ratio	30%	30%	32%	28%

Figure 4.2. Failures, i.e. completely wrong answers (grade 1).

Influence of penalties

The actual influence of the penalties, see section 2.5, was as follows:

- The accumulated response time of the Text Viewers is approximately one minute per task for any mode of TeSS. This is considerably less than the average time for solving a task which varies from 23 to 31 minutes. Therefore, we do not expect this circumstance of the experiment to have influenced the results substantially. The same argument goes for the query response time which was 4 to 5 seconds.
- The circumstance that the text is ‘dead’ might have influenced our results. The two querying modes have probably not been much affected, since the selected texts could normally be presented in one or possibly two screen pages, where query terms are quickly recognised. On the other hand, the browsing mode might have suffered somewhat more since larger texts were sometimes selected for inspection.
- The missing index in one of the hard copy manuals has only impeded the use of the most effective retrieval mode, the manual searching, see the next section.

4.2 Computer-based versus manual searching

The two performance measures applying to PAPER, time used to solve a task and grade for quality of solution, are presented in figure 4.3. Comparing with figure 4.1, it is apparent that the printed manuals outperform computer-based searching. PAPER is significantly faster than the fastest mode of TeSS (null-hypothesis rejected at any computable p-level, $Z=-4.35$), and causes significantly higher grades than the mode of TeSS achieving the highest grades (null-hypothesis rejected at the

$p < .006$ level, $t = -2.51$, $df = 623.8$). PAPER is found 31% faster than the fastest mode of TeSS, 48% faster than the slowest mode of TeSS, and causes 29%-38% fewer failures.

PAPER	
time	15.85
grade	3.14
failure ratio	20%

Figure 4.3. Mean time (minutes), grade (lowest: 1, highest: 5), and failure ratio for manual searching.

An advantage of online documentation is that it is available to any number of users at a time. Furthermore, the retrieval facilities are much more powerful. However, one subject gave up solving a task using TeSS with the comment that he lacked general knowledge of the tools involved—to proceed he needed a manual. TeSS provided him with exactly that manual, but somehow its form was inappropriate to the task. One explanation for this could be that it is easier to read longer pieces of text on paper than from a computer screen [Mills & Weldon 1987]. Another reason might be one of habits and expectations: You consult online documentation when you have a rather specific problem; otherwise you turn to a printed manual or a textbook.

In a comparison of print and electronic encyclopaedias, Marchionini [1989] finds that subjects took almost twice as long to conduct the electronic searches. Nevertheless, half of the subjects thought the electronic encyclopaedia faster, and one third considered it easier to use. Egan et al. [1989] report two studies of Superbook, before and after improvements. In the first, there was no significant difference between the number of tasks completed using Superbook and printed texts, but tasks were solved significantly faster with the printed texts. In the second study, a significantly greater number of tasks were completed using Superbook than using the printed texts, and completion times were almost identical. These examples from the literature illustrate, in accordance with the TeSS study, that it takes more to develop competitive computer-based text retrieval systems than just representing the text electronically and introducing a selection of more or less sophisticated retrieval facilities.

4.3 Logical expressions versus Venn diagrams

The performance comparison of LOGICAL and VENN is in favour of VENN. Generally, the subjects also prefer VENN to LOGICAL, see figure 4.8 later in this section.

Query correctness and expressive power

LOGICAL suffers from two problems with the logical query expressions. First, 7% of the queries (186 out of 2697) are syntactically incorrect, compared to 0.8% in VENN (22 out of 2733).

Second, in LOGICAL 26% of the queries containing both an AND and an OR (23 out of 90) probably misrepresent the subjects' intent. These queries are all of the form 'x AND y OR z' where y and z belong to the same category and x belongs to another; for example, 'callback AND procedure OR function'. Here it seems as if the subjects, incorrectly, expect OR to bind the terms more strongly than AND. In a large empirical study Avrahami & Kareev [1993] find the same misconception of this class of queries. They also find that otherwise AND is rightly considered to bind more strongly than OR. Being computer science students, the subjects are well-acquainted with boolean logic. The problems are not caused by lack of knowledge, but by failure to apply possessed knowledge. It seems as if the formal rules governing logical expressions are simply inconvenient to humans when they are intellectually occupied with other tasks. In VENN, the misconception is prevented by the faceted structure of the queries.

The Venn diagram interface was built in an effort to increase the intuitiveness of the user interface and is based upon a prestructuring of the query. As Meadow et al. [1989] point out this prestructuring is achieved at the cost of a certain loss of expressive power compared to conventional boolean retrieval. To investigate the consequences of these limitations, all queries executed in LOGICAL were tested to determine whether they could be expressed in VENN. It was found that 97% of them (2612 out of 2697) could be transformed, so the limitations, for instance that at most three facets can be visually combined in VENN, are of merely marginal importance.

Query complexity

Previous studies show that queries formulated by end-users tend to be very simple, involving only the most basic facilities [Borgman 1986b]. Figure 4.4 summarises the query complexity employed by the subjects in the TeSS experiment by giving the average number of the various constituents of the queries. For VENN, the reported number of ANDs, ORs, and NOTs refer to the logical expression equivalent to the Venn diagram (see section 2.2).

	LOGICAL constituents given by user	VENN constituents equivalent to Venn diagram given by user
words	1.97	2.29
ANDs	0.84	1.36
ORs	0.07	0.04
NOTs	0.008	0.59
asterisk wildcards	0.43	0.51

Figure 4.4. Average number of constituents of the queries.

Queries are kept very simple; in fact, the percentage of queries consisting of just one word is 34% in LOGICAL and 21% in VENN. Furthermore, neither LOGICAL nor VENN has made the subjects include many alternative terms in their queries. The limited use of the asterisk wildcard and, especially, of OR indicate that the subjects prefer to provide the necessary richness of alternative wordings by a sequence of queries rather than by one all-inclusive query. This reflects an interactive search strategy enabling the subjects to exploit information obtained earlier in the query sequence.

On the average, queries in VENN include more words and more ANDs. This extra complexity is almost exclusively caused by phrases, i.e. the conjunction of several words into one query term. In LOGICAL, the same phrases can be constructed by explicit use of ANDs (for an example see figure 2.2), but this is not done to the same extent. Forcing the user to type ANDs between the words of a phrase prevents the unreflected use of it as one unit. For some reason, this has the effect that phrases are used less.

In LOGICAL explicit NOTs can be included at will, but they are hardly used. NOTs are much more common in VENN, where they appear implicitly every time a subject retrieves the hit list of any other subset than the intersection of all facets in the query. It remains an open question whether the subjects are aware of these NOTs and use VENN accordingly. Though the selected subset is shown by highlighting the relevant part of the Venn diagram, the potential misconception arises from the fact that no selectable subset corresponds to the whole of, for instance, set A. Clicking in the subset labelled A corresponds to the query including A, but excluding B and C.

Number and variability of queries

When a subject starts on a task, the first query expresses his initial, incomplete attempt to reach a solution. If this query does not provide the subject with the information needed, another must be formulated. At this point the user is subject to what psychologists call anchoring, i.e. the tendency to make insufficient adjustments to initial values when judging under uncertainty [Tversky & Kahneman 1974; Blair 1980]. Thus, the subjects may tend to refrain from abandoning the initial query terms or from adjusting them very far, making the subsequent queries biased towards the initial one.

To measure the adjustments made to the queries in the TeSS experiment the concept of a thread is defined: Two queries are threaded if they have at least one term in common. If two successive queries are not threaded, i.e. have no terms in common, the second query marks the start of a new thread. Figure 4.5 shows the average number of queries and threads used to complete a task. For VENN the number of queries is the accumulated number of hit lists retrieved and of hit distributions computed without subsequent retrieval of a hit list.

	LOGICAL	VENN
queries	7.42	9.37
threads	2.50	1.52

Figure 4.5. Average number of queries and threads per task.

It takes more queries to solve a task using VENN than using LOGICAL, but it takes fewer threads, i.e. the number of queries in each thread is considerably larger in VENN. In VENN, each query usually builds upon the previous at the same time as the retrieving of hit lists for different subsets causes profound adjustments of the queries. VENN invites retrieval of hit lists for different subsets since a single mouse-click is enough to specify the adjustment of the query, whereas in LOGICAL it requires a keyboard-based change of the query itself. It can be noted that 45% of the hit lists retrieved in VENN belong to other subsets than the intersection of all facets in the query. Thus, rather than restricting the scope of the queries the continuity reflected in the long threads are achieved in combination with considerable diversity. In this way the subjects have exploited the possibility of developing their queries incrementally, a result that coheres with Hearst [1994] and Spoerri [1994] who expect the Venn diagram to support the users in exploring their information need.

4.4 Browsing

BROWSE is the mode of TeSS obtaining the best mean performance, but it does not provide stable, good performance. On three of the 20 tasks the quality of the solutions is very low, much lower than with the other modes of TeSS. Thus, browsing seems well-suited to most tasks but unsuited to some tasks, a circumstance which suggests that it should not be the only available retrieval technique.

In BROWSE two facilities are available: direct access through *find string*, and hierarchy expansion through *more detail* and *less detail*. Direct access browsing can be described as a very simple kind of keyword-based retrieval, the keywords being all words appearing in headings. The keywords are available for inspection by scrolling the table of contents and always appear in context, i.e. as part of a heading which, in turn, is presented with indication of its position in the entire manual. Hierarchy expansion allows the user to focus his search on seemingly relevant parts of the table of contents, to backtrack if they turn out to be dead ends, or to leave them open while investigating other possibilities.

Though direct access browsing and hierarchy expansion can be mixed freely, 41% of the tasks solved with BROWSE (127 out of 310) are solved with either one or the other, in both cases with above average performance. Restricting oneself to retrieval by direct access browsing means using a retrieval system with just three simple commands, namely *find string*, *view block*, and

view fragment. Still, 44 tasks are solved this way, and four subjects solve three or all four of their BROWSE tasks using direct access browsing only. Regarding hierarchy expansion, 83 tasks are solved with this facility only, and ten subjects solve three or all four BROWSE tasks this way.

Girill & Luk [1992] discuss a number of problems connected with hierarchical path walking, which is well-known from menu selection systems and could be considered a restricted form of the hierarchy expansion in TeSS. One of the problems is that the most discriminating headings are near the bottom of the hierarchy, not near the top where the user starts his search. To overcome the problems Girill and Luk suggest a combination of hierarchical path walking and hypertext facilities providing direct access. In accordance with this suggestion the majority of the browse tasks are solved using both browsing by hierarchy expansion and by direct access. However, in four cases out of ten the subjects use just one of them and obtain above average performance. Once more the message 'keep it simple' is worth serious consideration before accepting more complexity in the pursuit of properties such as completeness and expressive power.

4.5 Browsing and querying combined

Having both browsing and querying available, the user may either choose the one he finds the more appropriate to the task or combine the two techniques in his efforts to satisfy his information need. Thus, ALL offers much freedom and flexibility. On the other hand, it provides no guidance on how best to mix and utilise the facilities available. There are some indications that the subjects have suffered from this lack of guidance, see below, but specific causes for the low average performance have not been identified.

Performance versus preference

During the third set of tasks, the implementation tasks, the subjects were free to use the mode of TeSS they preferred. Figure 4.6 shows for each mode the number of sessions carried through. In 98% of the cases the subjects chose ALL, though they performed better with the other modes of TeSS. Thus, the subjects do not prefer the mode with which they perform best. This result is an exception conferred to the general finding that users prefer the objectively best system [Nielsen & Levi 1994] and must be recognised as an indication that commonly used performance measures such as time, grade, and operational errors can be missing something of essential importance to the subjects.

As choosing any other mode of TeSS than ALL means cutting off possibilities, it seems probable that the subjects have chosen ALL to avoid to choose. However, choosing ALL only postpones the choice, and leaves the subjects with multiple available retrieval techniques, i.e. in a more open, but also more complex, situation. In this case the increased possibilities have resulted in decreased performance. This illustrates that designers shall be wary of providing several means to reach the same objective.

	BROWSE	LOGICAL	VENN	ALL
sessions	2	1	6	572

Figure 4.6. Subject preference in terms of retrieval mode used.

Sequences of tactics

To get a picture of how browsing and querying were combined, the commands available in ALL were divided into three groups: browsing, querying with Venn diagrams, and querying with logical expressions. In the following, an uninterrupted sequence of commands from one of the groups is called a tactic. A browsing tactic can include exploration of the manuals with *find string*, *more detail*, or *less detail*, and opening of Text Viewers. The two querying tactics include the actual querying commands as well as the preceding selection of the texts to be searched and any subsequent viewing of one or more hits, as long as it involves no browsing commands.

A task is solved through a sequence of tactics, for example this sequence of three tactics: browsing, Venn diagram querying, browsing. Figure 4.7 shows the sequences grouped according to the number of tactics used. The figure seems to indicate that the more trouble a subject experiences solving a task, the more often he will change tactic; and the more often he changes tactic, the worse he performs. One explanation for this could be that the subject does not commit himself to the tactic chosen. Having trouble finding a solution he quickly resorts to questioning the choice of tactic and, thus, the search proceeds as a series of half-done attempts.

number of tactics in sequence	proportion of the tasks	average time	average grade
1	31%	16.77	3.15
2-3	30%	21.60	2.79
4-5	18%	29.13	2.68
6-35	21%	65.97	2.65

Figure 4.7. Tasks solved with ALL grouped by the number of tactics.

The sequences of tactics tell something about how the subjects have combined browsing and querying. The frequencies with which the three tactics are used are summarised in figure 4.8 which covers the first five choices of tactic. 31% of the tasks (97 out of 314) are solved without combining the available retrieval techniques; in these cases the subjects have chosen the technique they preferred for the task and held on to it. The remaining 69% of the tasks involve a combination of two or all three retrieval techniques; here the subjects have found different retrieval techniques appropriate to different parts of the search.

Browsing is the most popular retrieval technique. It is used as the first tactic for more than half of the tasks, and, at any stage throughout the search, browsing is used for at least one third of the tasks. Among the two ways of querying, Venn diagram querying is used approximately twice as often as querying by logical expressions. Figure 4.8 covers all tasks solved with ALL; isolating the good solutions (those graded 4 or 5) reveals a very similar picture, except for a slightly stronger tendency to use browsing as the first tactic.

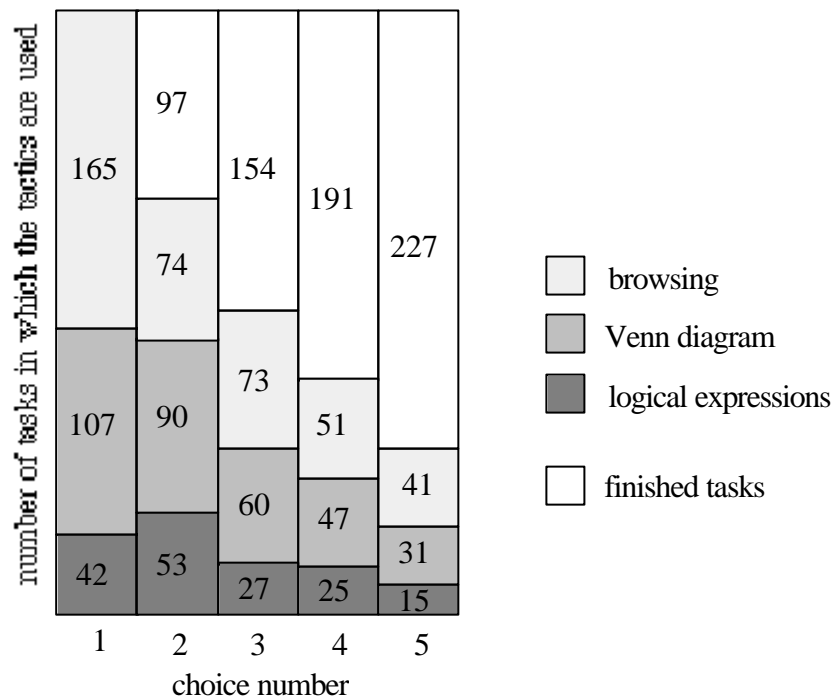


Figure 4.8. The frequencies of the three tactics calculated for the first five choices of tactic.

5. Conclusion

Four modes of the computer-based text retrieval system TeSS, which gives access to online documentation, were compared to each other and to printed manuals. The subjects in the experiment were computer science students being first-time users of TeSS without prior knowledge of the programming tools on which the tasks were based.

All significant differences between the four modes of TeSS are differences in the interaction processes, not in the quality of the solutions. It seems as if the subjects compensate for the strengths and weaknesses of each retrieval mode by adjusting their own effort. This is possible within certain limits, as shown by the fact that the use of printed manuals is significantly more effective than any mode of TeSS, a result which accords with other comparisons of computer-based and manual searching. Therefore, claims about the effectiveness of computer-based text retrieval must be wary in situations where a hard copy of the body of texts is manageable to the users.

Among the modes of TeSS, browsing has the best average performance, but it is found unsuited to some tasks. This demonstrates that browsing is a very valuable, but incomplete, retrieval technique—a strong argument in support of the widespread recommendation of combining several retrieval techniques. In line with this, the subjects almost unanimously prefer the combination of browsing and querying, where they stay in an open situation allowing them to choose among the retrieval techniques. But, contrary to our working hypothesis, this mode is less effective than the others. It seems as if the open situation of the combined mode is more complex and leads to searches which proceed as series of half-done attempts. The discrepancy between performance and preference also indicates that the subjects' perception of usage effectiveness is not covered by the objective performance measures commonly used.

In the two querying modes, the subjects restrict themselves to many, simple queries. Thus, the complexities of the information retrieval tasks are managed by exploiting the interactive capabilities of TeSS, rather than by building few and complex queries. It seems as if the formal rules governing logical expressions are inconvenient to humans when they are intellectually occupied with other tasks. The Venn diagram mode, which offers a prestructuring of the query into facets and simplifies the logical formulation of the query, performs better than conventional boolean retrieval.

To characterise the dynamics of the interaction processes two concepts have been introduced: threads and sequences of tactics. Threads in a query sequence describe continuity during information retrieval. The Venn diagram mode takes more queries to solve a task than conventional boolean retrieval, but uses fewer threads. This continuity is achieved along with considerable diversity obtained by exploiting the subsets of the Venn diagram. Sequences of tactics are useful to describe how different retrieval modes succeed each other as the retrieval process evolves from beginning to conclusion. In this study 31% of the tasks are solved using one tactic only. Browsing is the most frequently used tactic. Among the querying tactics the Venn diagram is used approximately twice as often as conventional boolean retrieval.

Acknowledgments

The design and implementation of TeSS as well as the design and execution of the experiment was a joint effort by the TeSS group in which we collaborated with the computer science students Jette Holm Broløs, Marta Kristín Lárusdóttir, Kristian Bang Pilgaard, and Flemming Steen Sørensen. Their expertise and strong commitment not only made it possible to fulfill our ambitions with the project; it also turned the project period into one of the intensive periods where we really learned. We are indebted to all the students who chose to participate in the experiment as subjects; and to Mjølner Informatics, Aarhus, who offered us a full version of their new object-oriented programming environment Mjølner Beta [Knudsen et al. 1993].

We are grateful to Stuart K. Card, the associate editor handling the review process, and Anker Helms Jørgensen, and Peter Naur for their careful reading of an earlier version of the article

giving to us many, judicious proposals for clarifying the description and the interpretation of the experiment. Also, the very helpful feedback from the reviewers is gratefully acknowledged. Finally, for valuable comments, we owe thanks to Ketil Perstrup and David Sands.

References

- Avrahami, J. & Kareev, Y. (1993). What do you expect to get when you ask for “a cup of coffee and a muffin or a croissant”? On the interpretation of sentences containing multiple connectives. *International Journal of Man-Machine Studies*, **38**, 429-434.
- Bates, M. J. (1986a). Subject access in online catalogs: a design model. *Journal of the American Society for Information Science*, **37**, 357-376.
- Bates, M. J. (1986b). Terminological assistance for the online subject searcher. In *Proceedings of the Second Conference on Computer Interfaces for Information Retrieval*. p. 7. Defense Technical Information Center, Cameron Station, Alexandria, Virginia.
- Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review*, **13**, 407-424.
- Belkin, N. J. & Croft, W. B. (1987). Retrieval techniques. *Annual Review of Information Science and Technology*, **22**, 109-145.
- Belkin, N. J., Oddy, R. N. & Brooks, H. M. (1982). ASK for information retrieval: part I. background and theory. *Journal of Documentation*, **38**, 61-71.
- Blair, D. C. (1980). Searching biases in large interactive document retrieval systems. *Journal of the American Society for Information Science*, **31**, 271-277.
- Blair, D. C. & Maron, M. E. (1985). An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, **28**, 289-299.
- Borgman, C. L. (1986a). 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.
- Borgman, C. L. (1986b). Why are online catalogs hard to use? Lessons learned from information-retrieval studies. *Journal of the American Society for Information Science*, **37**, 387-400.
- Broløs, J. H., Frøkjær, E., Hertzum, M., Lárusdóttir, M. K., Pilgaard, K. B. & Sørensen, F. S. (1993). *TeSS-projektet: Udvikling af et system til eksperimentel undersøgelse af brugergrænseflader til edb-baseret tekstsøgning*. DIKU-report 93/18. University of Copenhagen. In Danish.
- Dix, A., Finlay, J., Abowd, G. & Beale, R. (1993). *Human-computer interaction*. Prentice Hall, New York.
- Egan, D. E., Remde, J. R., Gomez, L. M., Landauer, T. K., Eberhardt, J. & Lochbaum, C. C. (1989). Formative design-evaluation of Superbook. *ACM Transactions on Information Systems*, **7**, 30-57.

- Girill, T. R. & Luk, C. H. (1992). Hierarchical search support for hypertext on-line documentation. *International Journal of Man-Machine Studies*, **36**, 571-585.
- Hearst, M. A. (1994). Using categories to provide context for full-text retrieval results. In *Proceedings of RIAO'94. Intelligent Multimedia Information Retrieval Systems and Management*. pp. 115-130. Rockefeller University, New York.
- Jerke, K.-H., Szabo, P., Lesch, A., Rössler, H., Schwab, T. & Herczeg, J. (1990). Combining hypermedia browsing with formal queries. In D. Diaper et al., Eds. *Human-Computer Interaction - INTERACT'90*. pp. 593-598. Elsevier Science Publishers, North-Holland, Amsterdam.
- Knudsen, J. L., Løfgren, M., Madsen, O. L. & Magnusson, B. (Eds.) (1993). *Object-oriented environments - the Mjølner approach*. Prentice Hall, Englewood Cliffs, NJ.
- Marchionini, G. (1989). Making the transition from print to electronic encyclopaedias: adaptation of mental models. *International Journal of Man-Machine Studies*, **30**, 591-618.
- Marchionini, G. & Shneiderman, B. (1988). Finding facts vs. browsing knowledge in hypertext systems. *IEEE Computer*, **21**(1), 70-80.
- Meadow, C. T., Cerny, B. A., Borgman, C. L. & Case, D. O. (1989). Online access to knowledge: system design. *Journal of the American Society for Information Science*, **40**(2), 86-98.
- Michard, A. (1982). Graphical presentation of boolean expressions in a database query language: design notes and an ergonomic evaluation. *Behaviour and Information Technology*, **1**, 279-288.
- Mills, C. B. & Weldon, L. J. (1987). Reading text from computer screens. *ACM Computing Surveys*, **19**, 329-358.
- Nielsen, J. & Levi, J. (1994). Measuring usability: preference vs. performance. *Communications of the ACM*, **37**(4), 66-75.
- Radecki, T. (1988). Trends in research on information retrieval—the potential for improvements in conventional boolean retrieval systems. *Information Processing & Management*, **24**, 219-227.
- Smith, S. W. (1976). Venn diagramming for on-line searching. *Special Libraries*, **67**, 510-517.
- Spoerri, A. (1994). InfoCrystal: integrating exact and partial matching approaches through visualization. In *Proceedings of RIAO'94. Intelligent Multimedia Information Retrieval Systems and Management*. pp. 687-696. Rockefeller University, New York.
- Tversky, A. & Kahneman, D. (1974). Judgment under uncertainty: heuristics and biases. *Science*, **185**, 1124-1131.

Appendix A. The experimental tasks

The full set of tasks is presented below. Note, that the tasks were originally stated in Danish and only partly in the terminology of the X Window System. The set of tasks consists of two training

tasks, twenty information retrieval tasks, and three implementation tasks.

Training tasks

Task 1. The first task is a guided tour through TeSS intended to uncover what mademoiselle Viola Fraculanski [the principal character in the walk-up-and-use introduction to TeSS] learned about the standard width of buttons in the X Window System.

Try, initially, to find text blocks with the word 'button' in their heading. This can be done by means of 'find string'. Glance through some of the text blocks found.

Shift to 'Venn diagram mode' and search for the text blocks containing 'button' and 'standard'.

Replace 'button' with 'button*' to make the search insensitive to, for instance, plural endings and repeat the search. How many text blocks contain words matching 'button*'?

Look at the headings of the text blocks containing both 'button*' and 'standard'. Find the text block '5. Pop-up Widgets' and glance through it.

Mademoiselle Viola Fraculanski's first encounter with TeSS ended here, and so does this task.

Task 2. What is the difference between a Form Widget and a Box Widget?

Retrieval tasks

Task 3. In the user interface of TeSS text fields and buttons look much the same. How can the buttons be changed to appear with rounded corners?

Task 4. A function is to be executed every time a button of the class Command Widget is activated. How should this function be registered?

Task 5. Radio buttons are used in situations where exactly one option must be chosen from a group of options. Which widget class is used to implement radio buttons?

Task 6. TeSS is divided into four subwindows, so-called panes, separated by thin horizontal lines with square handles. To distinguish these lines more clearly from other horizontal lines in the interface, they could be made thicker. Find out how, if at all, this can be done.

Task 7. The font in 'TeSS Control Window' should be changed to Times. How is that done?

Task 8. How can a dialog box with, for example, an error message and an OK button be made?

Task 9. Which facility is available for string searching in Text Widgets?

- Task 10. Widgets, such as the list with hits from a search, can be made temporarily invisible and thereby inaccessible. How is that done?
- Task 11. The caption on the button ‘done’ should be changed to ‘quit’. How is that done?
- Task 12. The three list fields in TeSS have both a vertical and a horizontal scroll bar. As the items in these list fields are usually shorter than the width of the list fields, the horizontal scroll bar is usually superfluous. Which facility makes it possible to show the horizontal scroll bar only when it is needed?
- Task 13. In Athena Widgets the creation of a widget is a three-step process, where each step can be performed with an Xt-function. What are the names of these three Xt-functions?
- Task 14. The text blocks that match a query are indicated with an asterisk in the list with hits from the search. This indication should be changed to the use of bold font. Given the information that the hit list is implemented in Athena Widgets by means of the widget class List Widget, find out how, if at all, this can be done.
- Task 15. The size of the titles of ‘TeSS Control Window’ and its four subwindows should be made a little larger than the remaining text. How is that done? Assume, for the sake of convenience, that it is not necessary to change the size of the fields containing the titles.
- Task 16. Each widget class has a number of resources controlling its appearance and functionality. Which facility is used to set the values of these resources?
- Task 17. Which widget class communicates with the Window Manager?
- Task 18. The three list fields in TeSS are implemented in Athena Widgets by means of the widget class List Widget. Do List Widgets provide the possibility to select more than one text block and copy them simultaneously to the list field with text blocks selected for querying?
- Task 19. How is a field with a picture or figure included in the user interface?
- Task 20. A Composite Widget has a Geometry Manager which positions the children of the composite widget within its window. Which technique shall the Geometry Manager use if the children are to be treated equally?
- Task 21. Where in the manuals can information about the initialisation of a widget class be found?
- Task 22. The edit field that appears when the button ‘find string’ is activated should be placed under the cursor so that the search string can be entered right away. Find out how, if at all, this can be done.

Implementation tasks

- Task 23. TeSS is divided into four subwindows, so-called panes, separated by thin horizontal lines with square handles. These handles make it possible to change the relative size of the subwindows within the ‘TeSS Control Window’. Restrict the possibilities for resizing the subwindows in such a way that their height can not become less than 2 cm.
- Task 24. In the subwindow ‘Formulation of Queries, Venn diagram mode’ the label ‘Select subset to show list of hits’ should be made invisible. Then the button ‘show number of hits in subsets’ should be moved from above the three circles to below them, i.e. to the place currently occupied by the label.
- Task 25. In the subwindow ‘Formulation of Queries’ (both query modes) the input fields get a scroll bar whenever the length of their contents exceeds their width. This facility is called ‘scroll bar when needed’. Make these scroll bars thinner, for example eight pixels wide.