A Search Interface for my Questions

Mark A.C.J. Overmeer

AT Computing bv
Toernooiveld 104, 6525 EC,
Nijmegen, The Netherlands
markov@ATComputing.nl

Abstract

Current interfaces to Search Engines are based on speed of delivery by the engine, and ease of use by everyone. However, if you are really looking for information, the results are often poor. In this paper, an experimental interface is demonstrated which has more ways to control the search and a very abstract representation of results. It demands more of the human and the machine, but should deliver better answers.

Key words: Interface for search-engines.

1 Introduction

People with little experience on the Internet are usually overwhelmed by the amount of information what can be found on it. They start learning to use search-engines to find interesting sites. Typical users type "games", and they get millions of locations which point to pages, which point to pages, which may actually contain some games. After a few clicks, they find a game they like.

For professional use of the Internet, the requirements are harder. Professionals do not just want *some* answer on their query, but preferably *the* answer. This calls for extended ways of asking questions which are not provided by the usual interfaces. If using the search-engine becomes a bit harder is less important for professional users, because they can spend time on learning to work with the more complex interface. An investment in learning is returned by the speed gained when searching.

What kind of questions will a professional searcher — a librarian, a scientist, a journalist — ask?

- “Give me information about keeping monkeys at home. I like to find small sites which are specialized in the subject. But, I’m not interested in Zoo’s.”
- “I am looking for a page which list links to sites about monkeys.” an indexing-site for the subject.
- “I want the most popular site about monkeys.”

When I say “monkey”, then pages about Gorilla’s, Chimpanzees, Baboons, etc. should be found, too.

Can you put any of these questions to existing search-engines? To be able to ask such specialized question, the search-engine must produce more data, and the interface must be able to handle the various results. The human must have tools to play-around with the initial results for some time to find the answer, and be continuously assisted by the engine.

When we look at modern search engines, it is obvious that *getting the thing to work* is hard. When a question is asked to the machine, it responds very quickly. You hear the developers cry: “hey! it works, at last!”, and then go on a well deserved four weeks holiday. There is no energy or money left to develop an intelligent interface. Of course, there are some experimental interfaces on Internet, like the “refine” option which was available for some time from AltaVista [1], but they are extremely rare. Presentation of the results is usually based on pages which are *hit*, showing a part of that page. Huge lists of answers are return, and you have to filter-out the data yourself... with little to no assistance by the search-engine (no after-care). Effort is put in ranking the results, but not in reducing the amount of results.

Search Engines should spend more time on after-care (and also pre-care) to help people find the best answer. Only at the end, when enough details about the question are known to give a small set of good answers, real pages or sites should be shown. Whilst all current spiders focus on simplicity of their interface, to attract as many users as possible, the interface presented in this paper concentrates on the requests of *experienced and trained users*. This interface is designed to handle *huge quantities* of information, and is not designed for indexing one single site, even if it is a very large site.

2 The Selection Process

The basic structure for the guided exploration on the question first concentrates on finding the right query based on a number of keywords. These keywords are to be qualified as the exploration progress. Thus, the focus is on reducing the number of sites, before actually showing any sites. The steps are shown in figure 1 and consist of progressively applying *limiters* to reduce the

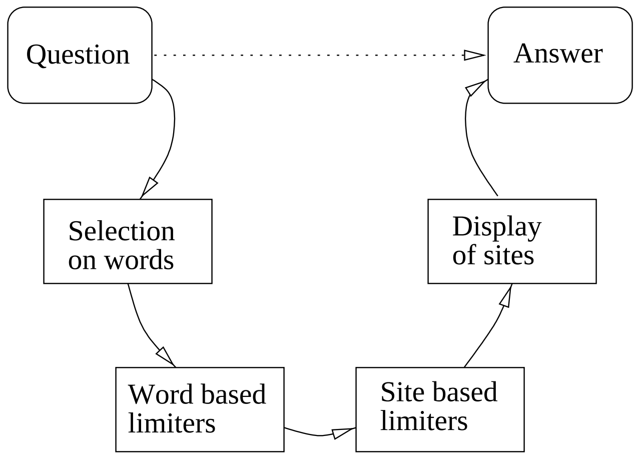


Fig. 1. The elements of the interface.

number of hits. This scheme is further complicated by the need for feedback on the criteria the searcher sets: the more ways the search can be influenced, the harder it is to find-out which limiter is doing what to the results. The full picture, feedbacks added, is shown in figure 2.

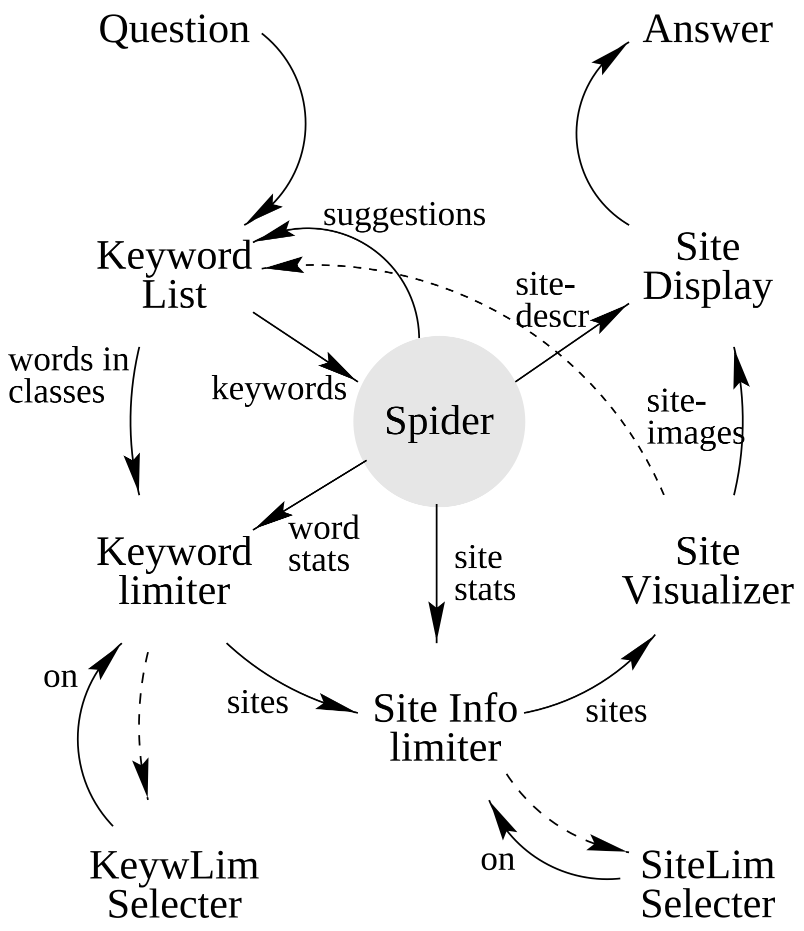
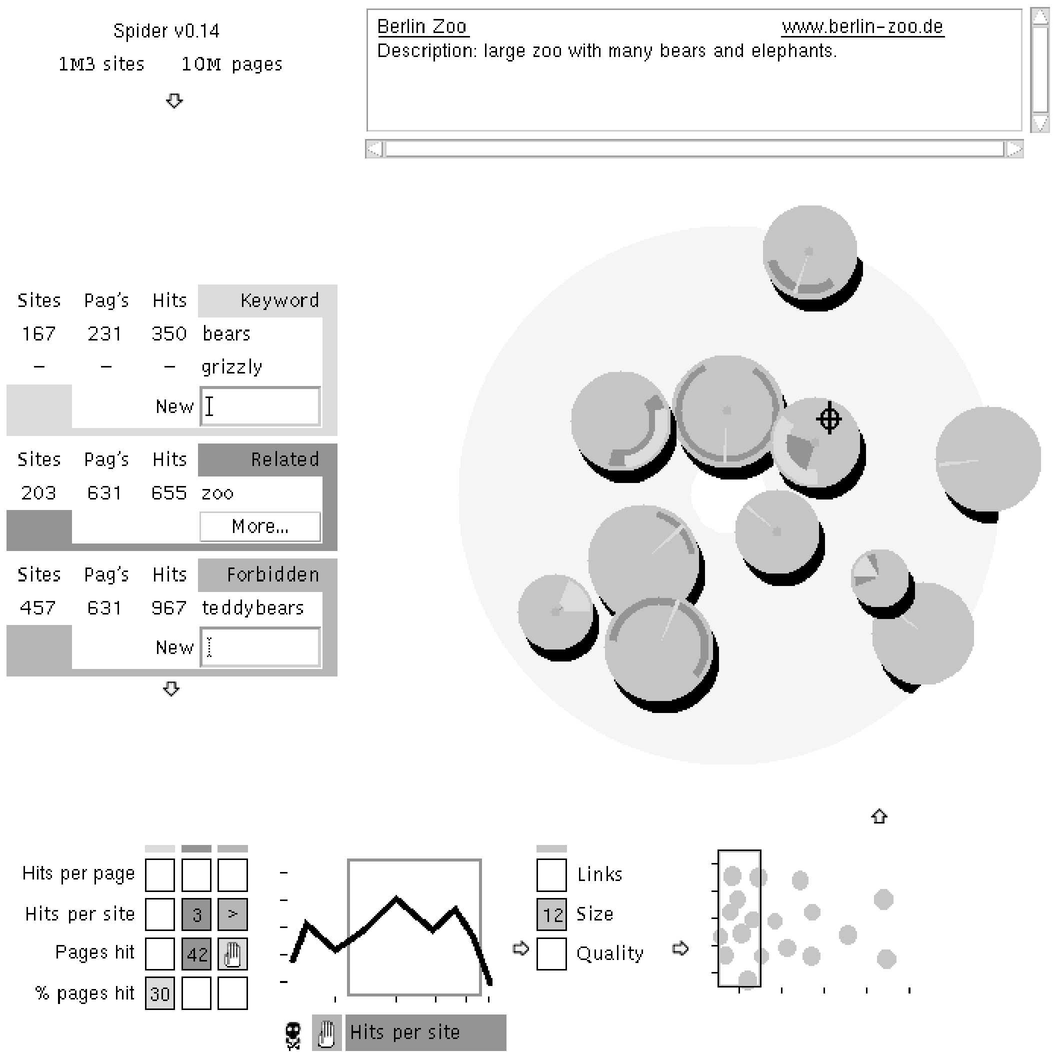


Fig. 2. The structure of the interface.


Spider v0.14

1M3 sites 10M pages

Berlin Zoo www.berlin-zoo.de

Description: large zoo with many bears and elephants.

Sites	Pag's	Hits	Keyword
167	231	350	bears
–	–	–	grizzly
		New	

Sites	Pag's	Hits	Related
203	631	655	zoo
			More...

Sites	Pag's	Hits	Forbidden
457	631	967	teddybears
		New	

Hits per page			
Hits per site		3	>
Pages hit		42	
% pages hit	30		

	Links
12	Size
	Quality

Fig. 3. The implemented interface.

In this section, we will sequentially visit the steps from initial question to the answer. Each time, a part of the scheme is translated into a part of the interface. The total interface is show in figure 3.

2.1 The Spider

The spider has the central role in supplying the raw information in answer to the user's question. In the uttermost top-left corner of the interface, the status of the spider and the connection to it is show. Figure 4 shows the functional design (from figure 1). of the spider interface on the left hand side and on the right it shows the corresponding implementation (from figure 3).

Connections to the spider are short lived because the spider has to maintain a process for the query. For possibly many thousands of requests in parallel,

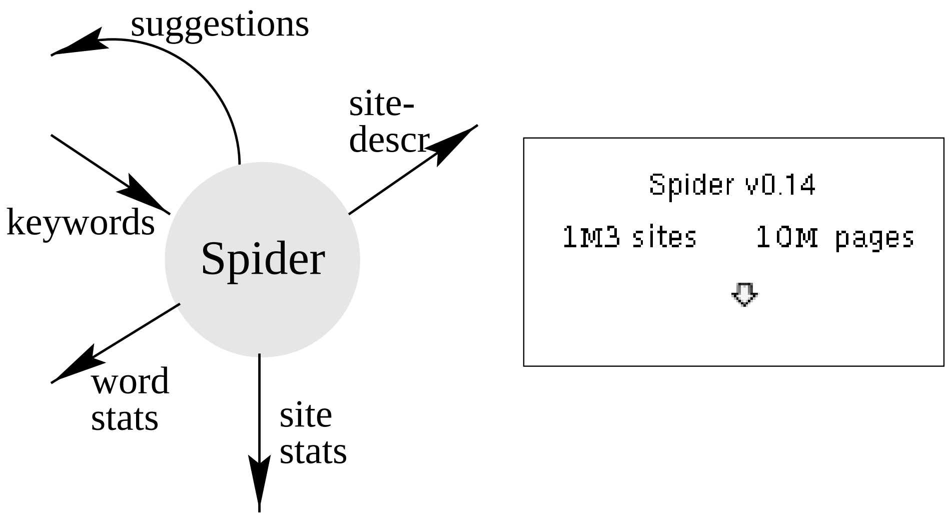


Fig. 4. Contact with the spider.

this is not acceptable.

Examples on useful facts which can be displayed here are

- the size of the databases of the spider;
- the connection-time and -status to the spider;
- the amount of data still to be produced by the spider to have a fully updated screen; and
- the load on the spider.

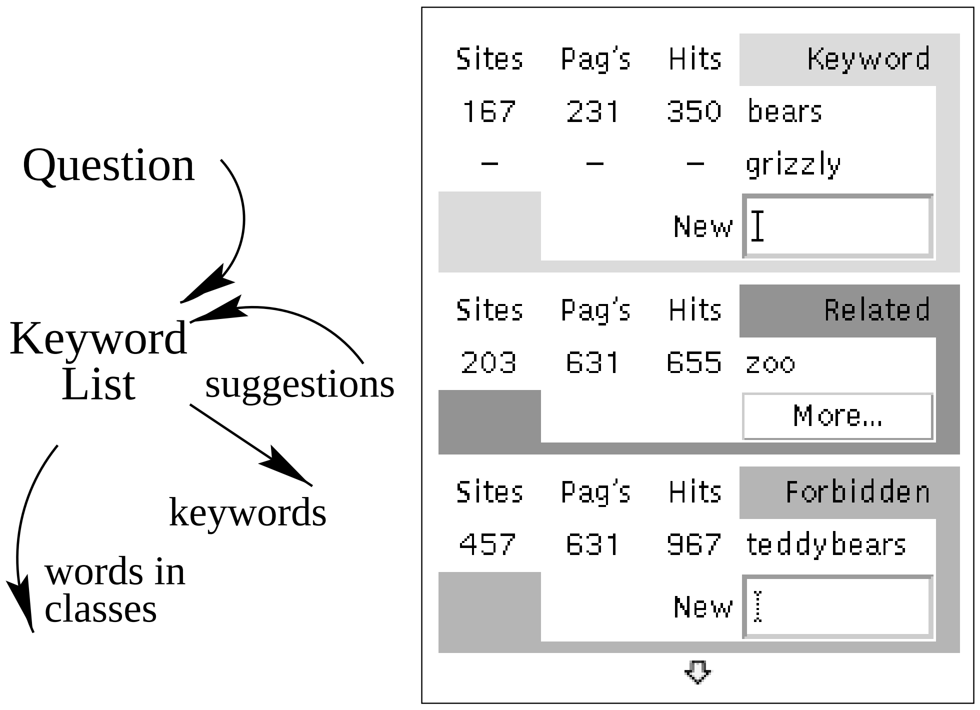
At this point "we start with everything".

2.2 Selection on Keywords

From "everything" we can select sites based on keywords. The keywords are part of the user's question. Although, it might seem contra-productive, but when the keywords are entered, the search engine will immediately start producing a list of words which may be related to the words given by the user. Relationships to other words can be made in various ways. A list of synonyms would be great (but a lot of work to produce). Words which are often found on the same page as the words from the query can also be used. Combinations of words made by other users in previous requests would be extremely helpful.

The user has to group the words from the question and the suggestions made by the spider into three categories: primary keywords (in the interface represented by the color green), related words (blue), and forbidden words (red).

Figure 5 shows how this works.



Sites	Pag's	Hits	Keyword
167	231	350	bears
–	–	–	grizzly

Sites	Pag's	Hits	Related
203	631	655	zoo

Sites	Pag's	Hits	Forbidden
457	631	967	teddybears

Fig. 5. Selection on keywords.

- The primary keywords. Initially, these are the words found in the question from the user.
- The (probably) related words, as suggested by the search-engine. The user can promote them to primary keywords. They can be removed when not related at all, or they can be demoted into a forbidden word.
- The forbidden words. Useful to exclude pages which contain the wrong meaning of a word, or for specialization of a subject.

For each word, the user gets an overview about

- the number of sites which contain the word;
- the number of pages over all sites which contain the word; and
- the number of hits: the word-count over the whole Internet.

These are easy figures to produce, because the spider can count them when it scans the pages found on the Net.

For each of the three word-groups, special limiters can be set. The limiters currently work only at whole word-groups, not at single words. This might change in future versions of this interface when experiments show that working with groups does not show sufficient detail.

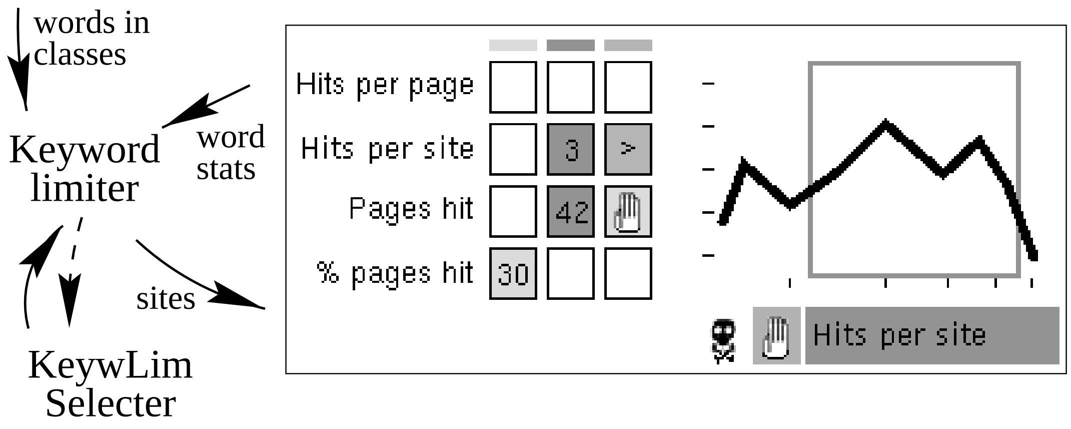


Fig. 6. Limiters based on keywords.

The action of the limiters are all shown as histograms (figure 6, the data is for example) with a possibility to adjust the lower- and upper-bound for your request, linear or logarithmic scaling, and cumulative, spike, or Gaussian presentation. Each word-group has a column (respectively keywords, related words, forbidden words). Selecting a box will set a limiter. A stopping-hand signals that the limiter is temporarily disabled. The value in the box shows the effectiveness of the limiter on the number of answers on the question.

At the moment, the following limiters are intended:

- *Hits per page per site.* The average number of hits per page which is hit in relation to the number of sites with hits. This visualizes the density of the hits, with respect to sites.
- *Hits per site.* The total number of hits for a site, in relation to the amount of sites which have that many hits. With this limiter, you can find sites with a large amount of information about the subject, independent from the size of the site.
- *Pages hit per site.* The total number of pages for a site which contain any of the words, in relation to the amount of sites which have that many pages hit. This indicated sites which are likely to have a specialized section about the subject.
- *Pages hit percentage per site.* The number of pages hit in a site as percentage of the pages of the whole site, in relation to the number of sites which have that word. This limiter can be used to find sites specialized in the subject

you look for.

- *Limiter on location.* This is not a histogram, but a checklist with possibilities to restrict the appearance of words to (a combination) of
  · the title of the page,
  · the meta-keyword line in HTML-pages,
  · the meta-description line in HTML-pages, and
  · the content of the page.
  This limiter is not yet implemented.

More than one limiter can be set at the same time, for any of the three groups of words.

These limiters require more detailed information from the search-engine than is currently available. The first implementation of the interface will not ignore overlapping hits: when two words from the same group meet on the same page, this page will be counted twice. Hence, the histogram will not show the real distribution of suitable sites. This is certainly not optimal, when we consider that many words *will* have overlaps because they are related. The reason not to implement the best solution is the exponential behavior of this data: the search engine has to recalculate all the possible combinations, and do this over for each word each time a word is added, moved or deleted from the list of selected keywords. The intention is to have to spider produce *a lot* of suggestions on words to pin-point the question optimal. Exponential behavior will be destructive. By just ignoring the overlaps, the situation changes to simple linear lookups. Experiments shall show if this is simplification will give acceptable results.

2.4 Limiters on Sites

Based on the keywords, we will get a large number of sites which may give our answer. Now we add limits on the sites themselves (figure 7). These selectors are displayed with the same type of histograms as used for the word-limiters. The display of the site-limiter's details overlaps the display of the word-limiters, to save screen-space.

The following limiters are currently planned:

- *External links per kb text.* Indexing sites can be your special interest, or just not what you are looking for. The number of links to external sites is shown.
- *Number of images per kb text.* Do we want someone's holiday photo's or documentation about this beautiful island in the Pacific?
- *Size of the site in kb text.* Large site are often more serious and are better maintained, but small sites can be more inspiring.

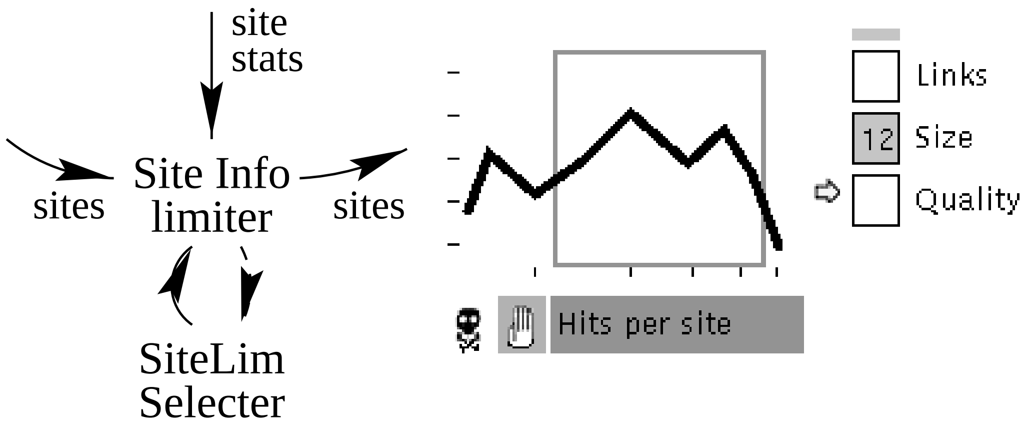


Fig. 7. Limiters based on site-information.

- *Quality of the site.* The quality of a site is a subjective figure which may combine information about
  · the frequency of site-update,
  · the detected availability of the site,
  · the existence of meta-keyword and meta-description for the pages; and
  · the quality of HTML used.

2.5 Displaying sites

With all words in their right group, the required word-limiters in their place, and the other limiters set-up, we can (finally) start showing sites and pages.

The display of sites (figure 8) is very abstract: it shows small circles representing sites, 'floating' in the direction of the center of one large circle which represents the 'universe'. Five dimensions are used to assess the information about a site. These dimensions are displayed on the abstract display in such a way as to allow a site is human pattern-recognition to make the final choice whether to visit a site or not. For example, the closer a circle is to the center, the better the site fits the request.

Using the slider below it, the scale of the diagram can be set: sites are shown to float away from the center or closing in. When the user moves the mouse-pointer on a site's circle, the description of the site (as stored in the HTML) is shown in a separate message window. A click will bring the user to the part of the site where most hits where found.

Various details about the search results are shown by the circle which represents the site:

- *The distance* to the center of the picture shows the chance that the site

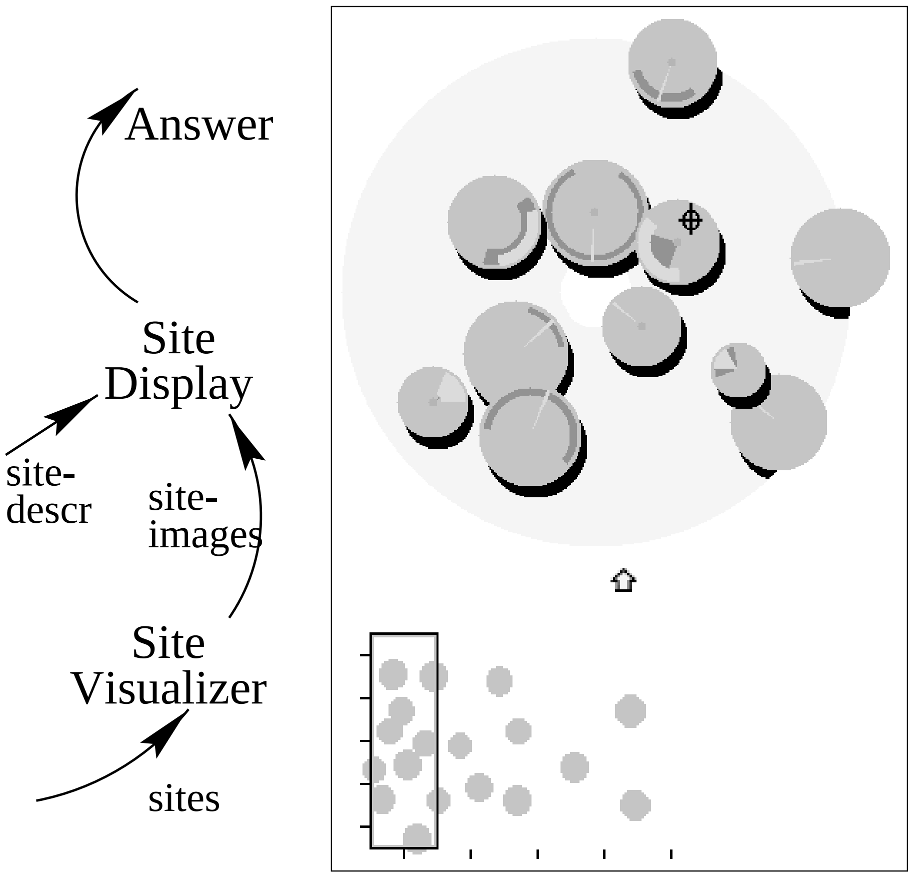


Fig. 8. Display of the results.

contains the answer on your question. The closer to the center, the higher the chance.

- *The size* of the circle shows the size of the site. This uses non-linear scaling so that very small sites shall be visible and very large sites shall not cover the whole screen.
- *The amount of backdrop* of the circle is an indication about the quality of the site.
- *The colors filling parts* of the circle indicate how many hits are found for each word-group. Each group has its own color: green for primary words, blue for related words, and red for forbidden words. By looking at the relative amount of colors in the circle, you can estimate how the site got its rating: primarily by keywords, or by the related words.
- *The angle of the color-bows* is used to show the percentage of pages hit by the keywords. A narrow pie-piece, colorful filled from the center to the border of the circle means highly concentrated hits. A wide bow, along the outside of the circle means a few hits spread over many pages. The bows/pie-parts are directed to the center of the display, as if the center glows them.

A large variety of information concentrated in one display. The eye will recognize trends after a few queries.

3 Project Status

The visual part of the interface is implemented and can show randomly generated data. The protocol to talk to a simulated spider is under development. The interface is written in Java, so can be run stand-alone and from any browser on most computers.

Real data is needed to prove the concept, but before that can be done, the search-engine has to be developed. Plans for constructing a spider are described in [2].

4 Conclusions

Finding our way in huge quantities of non-homogeneous data requires more assistance than current search-engines are offering. There are ways to help us finding answers to our questions, but these are not implemented in any practical interface yet.

A sure thing is that more searching power also adds more complexity for the user. Experiments with real data on the interface described here have to show if the search will be improved by it.

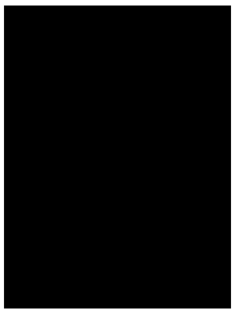
References

[1] AltaVista Inc. The AltaVista Search Engine. `http://www.altavista.com`

[2] M.A.C.J. Overmeer, My Personal Search-Engine, Proceedings of TERENA-NordUnet conference 1999.

My gratitude to Duncan Barclay for corrections and improvements.

Vitae



Mark Overmeer got his MSc in Informatics from the University of Nijmegen, The Netherlands in 1990. Since then, he gained professional experience in maintaining a large variety of UNIX-systems, from tiny to super-computers.
In his current occupation, this knowledge is taught to system-developers and -maintainers at AT Computing bv, a leading training institute on UNIX and UNIX-related languages in the Netherlands.
Next to his professional activities in computers, he maintains a very popular Dutch Internet-site since 1995, and actively participates in development of Public Domain software.
`http://www.dhp.nl/~markov`