INSYDER - An Information Assistant for Business Intelligence

Abstract

The WWW is the most important resource for external business information. This paper presents a tool called INSYDER, an information assistant for finding and analysing business information from the WWW. INSYDER is a system using different agents for crawling the Web, evaluating and visualising the results. These agents, the used visualisations, and a first summary of user tests held in Great Britain, France and Italy are presented.

Keywords

UIs/visualization organizing and displaying retrieval results, (semi) automated search assistants, user studies

1   Introduction

The benefits of using external information for business intelligence[1] are significant. An enterprise must know more and more about its customers, its suppliers, its competitors, government agencies, and many other external factors. Valuable information about external business factors is readily available on the Web and its amount is increasing every hour. While a few WWW resources are used as data sources, the immense resources of the Internet are largely untapped. What is needed is a continuous and systematic approach to make use of these untapped resources. Hackathorn proposes such an approach called Web farming: *"Web farming is the systematic refining of information resources on the Web for business intelligence."* [9]

This paper presents a WWW-application called INSYDER, an information assistant for finding and analysing business information from the Internet. INSYDER is a system using different agents for crawling the WWW, evaluating and visualising the results, which can be used as one important tool for the Web farming approach. The research project INSYDER was funded by a grant from the European Union, ESPRIT project number 29232. The paper is organised into the following chapters: Chapter 2 describes an Information Assistant approach and how this approach can support the user during information seeking tasks. Chapter 3 describes the functional characteristics, the technical architecture, the agents used and the available visualisations of the information assistant INSYDER. Chapter 4 presents first results of evaluations of the INSYDER system. Chapter 5 summarises the main results of this paper and gives an outlook on future work.

---

[1] "A business intelligence system ... provides a set of technologies and products for supplying users with the information they need to answer business questions, and make tactical and strategic business decisions." [15]

2   Information Assistants for Information Seeking in the WWW

2.1   Accessing Information

In recent years the number of documents published on the WWW has been increasing dramatically. This brought the research about information retrieval systems into the focus of people, dealing with the WWW. For most of them searching the WWW is just to formulate few query terms and to get back the results in a relatively short time. But information seeking is more than that, e.g. analysing and preparation of results found. One of the first steps when dealing with information seeking systems is to get an idea how to describe the information seeking process best.

A good example for a high level task approach is the four phase framework for information seeking by Shneiderman [28]:

- Formulation: expressing the search
- Action: launching the search
- Review of results: reading messages and outcomes resulting from the search
- Refinement: formulating the next step

For designing INSYDER we have chosen this framework, because from the user's point of view it covers all phases of the information seeking process in an easily understandable way. Various other models of the information seeking process can be found in [12].

In the literature a series of surveys concerning user interaction with the WWW as an information source can be found. One of the conclusions is that users often don't know how to express their information need [25], [22], users have problems with the current paradigm of information retrieval systems simply presenting long lists of results [36]. The following chapter presents our proposed solutions to these problems, the Information Assistant approach.

2.2   An Information Assistant for Information Seeking

In a debate of 1997 Ben Shneiderman and Paettie Maes argue about direct manipulation and interface agents [29]. Maes states that agents are no alternative for direct manipulation and that agent applications still need a good interface. Shneiderman on the other side pleads for visualisations to give the user the possibility to navigate within the data under their own control. We think that both of them are necessary and subsume these two approaches by speaking of INSYDER as an Information Assistant [17]. Also Eichmann demands agents for the information seeking: "*Users are seeking guidance and organisation in a chaotic, dynamic information framework. They are in a process of exploration when using the results of agents [...]*" [7]. The INSYDER Information Assistant acts on the user's behalf and is

built up using different agents using Information Retrieval techniques and a synchronised visualisation approach (see below). The agent literature may be classified into two main categories: agents concerned with intelligent behaviour (approaches resulting from AI research) and others, focusing on a strong interaction component. Wooldridge and Jennings [35] distinguish between weak and strong notions for agents. Many definitions of the term agent can be found in the literature. They all characterise agents with the help of attributes, e.g. reactiveness, proactiveness, autonomy, communication, co-operation or mobility [3].

2.3 Visualisation of Results of the Information Seeking Process

Two general aspects of "Information Visualization" can be found in the literature: Data presentation versus data exploration [12]. In the case of the presentation the message stands in the foreground, in the case of the exploration it is the discovery ("Visual data-mining"). In the INSYDER project visualisation is used to support the handling of the result sets, exploration and discovery.

The visualisations used focus on the result phase in the framework by Shneiderman et al. [28], which is the most interesting one from the "users point of view". Here the user gets the suggestions to satisfy his information need and it would be a good idea to help him finding the needle in the haystack by applying adequate visualisations. On the set level, which means the representation of the whole set of results, it will be interesting to get an overview. A lot of authors discuss the use of different visualisations in different forms. An overview is presented in [19]. For the document level much fewer ideas can be found. Approaches range from Tilebars [11] to Thumbnails [5].

3 The Information Assistant INSYDER

The idea behind INSYDER is that the user has different kinds of information needs, called spheres-of-interest. Each sphere-of-interest (SOI) represents an information need of the user [10]. The user might have a sphere-of-interest called "*Competitors*", one "*Development tools*", "Technology" and maybe one "*Recruitment*". In each of these spheres the user can define searches, watches and portals. A typical example for a search in the competitors sphere would be to look for new competitors, while a typical watch activity would be to monitor the Web-Site of a distinctive competitor. A web portal presents interesting links to different competitors WWW-sites. The advantage of SOIs is in accordance to the user's information needs in a structured way. This will help the user to navigate between and keep different information interests at the same time, easily. User studies in July 1999 on a mock-up version of the INSYDER system with six typical users have already proven this basic approach to be intuitive.

When defining a search the user formulates his information need in natural language and chooses

sources as starting points (e.g. web sides, search engines) from a list. After this the search is started and runs until it is stopped by the user. While searching the user may already have a look at the document hits. The relevance ranking of the Information Assistant uses a semantic analysis of documents, which is based on a semantic network provided with the system. This semantic network may individually be adapted to various application domains. It consists of concepts describing the environment of the system by using typical relationships, like "is-a", "consists-of" and so on (see Figure 1, different colours express different relationship's). By using this semantic network it is possible to find also documents which do not contain the terms of the query, but for instance a synonym. Another advantage is that the results may be more precise than results from other system as homonyms can be avoided. For instance a search for "bank" could result in the institute for money, it could be the memory bank in the computer or the bank at the shore. By specifying a subject-specific semantic net e.g. for the computer industry, INSYDER can determine that bank must have to do with the hardware of a computer.

The system stores all retrieved documents in an own database. The database is used as a repository, no further operations (e.g. build an index file etc.) are made. For the future it is planned to use this information also to get a description of the user and his likes resp. dislikes.
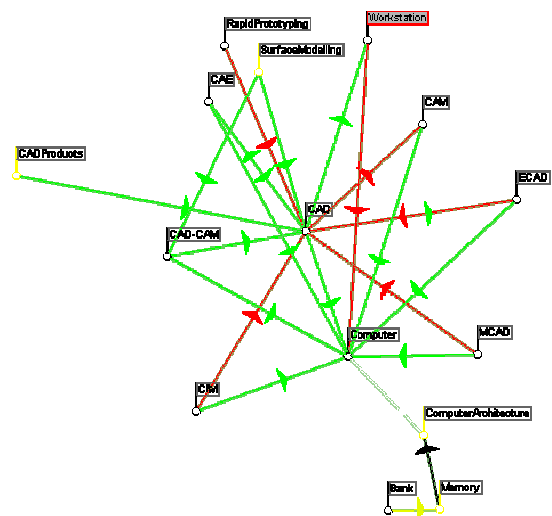

Figure 1: INSYDER semantic network [2]

3.1 INSYDER's agents

The INSYDER agents are simple agents from their architecture, but powerful from the point of view of user assistance. For the information seeking activities a multiagent system is proposed [21]. The INSYDER agents are using procedure calls for communication and a delegation principle for the co-operation. Using the semantic network as a representation of the domain knowledge the agents can be personalised and customised.

As the INSYDER agents are designed to run in a heterogeneous information space like the WWW, we choose to have stationary agents, instead of mobile ones. When thinking of the crawling agent (see below) it can be stated that the INSYDER agents have a degree of autonomy, meaning that they do not always need the interaction with the user to be able to fulfil a task. The implementation of the system was done in Java (interface components, visualisations and agents) and C++ (semantic kernel) using a COM interface.

3.1.1 Agents supporting the Formulation of Queries

A basic criticism brought up in the literature is that the users mostly are not capable to inform the system about their demand of information [25]. Shneiderman et al. propose therefore to use large text-entry fields, to encourage the users to type long search strings [28]. Other solutions in this field are methods of automatic query expansion [30], [33], which are successful (see [34]).

We propose a **formulation agent** to help the user formulating the query. This agent will have the task to visualise the query and especially to support the user by formulating the query. At first, the support should suggest further terms for the query and in a next step there should be a visualisation of the query. However these concepts are not implemented till now. Only a part of the extension of the query has been implemented. For this the concepts of the users query are worked out and are shown to the user. According to the importance of these concepts the results found will be assessed with the help of the semantic net.

3.1.2 Agents supporting the Search Actions

Today the action task is already supported by agents called Robots, Crawler, or Wanderer. Their common task is finding documents as well as listing these and giving these onto the database [31]. The INSYDER **crawling agent** obtains a list of pages from standard text indexes, e.g. AltaVista, Excite (it is configurable which sources should be taken) for a further active crawling. In this way of doing it, the users do not only get the limited (in terms of actuality and availability) results of the search engines. INSYDER works therefore like a metacrawler on the one hand (collecting the results of different search engines, eliminating duplicates, presenting resulting document hints to the user). On the other hand INSYDER is a robot itself, crawling to find more relevant document hints.

3.1.3 Agents supporting the Review of Search Results

In the INSYDER system a **concept ranking agent** is used. This takes into account that the user might want to rank the documents found according to different weight of occurring concepts. This means that the concepts get different ranks, presenting how well a particular concept describes a document. So a high ranked concept (e.g. 100) is seen to describe the document very well, while a low or even negative rank shows, that this concept doesn't describe the document well, if at all. In the INSYDER system a simple method is used to get the ranks of the concepts in the document, just by counting the frequency of the concept in the document. Another way would be to use the interrelation of concepts, meaning that the concepts, which have the most interrelations to other concepts in the same document, are seen to be those describing the document best. For the moment only the frequency model has been implemented and will be tested during the next month. From the query the concepts are extracted and shown to the user with default weightings, which can be changed by users. Documents are search upon the original query, the ranking is calculated based on the decision of the user about the importance and the nearness (proximity) of each concept compared to the concepts occurring in the documents found. Another possibility of using the concept query ranking is to use it in conjunction with the relevance feedback in the refinement phase (see below).

Another kind of assistance is to present the results not as a long list, but to cluster them. Therefore a **clustering agent** has been proposed. The task of this agent is to find out whether documents have similar common features and to cluster them by the occurrence of this feature (e.g. [24], [36]). While clustering describes the structuring by features which are derived from analysing a set of documents, the classification of documents is the organisation by given categorisations [13]. The project INSYDER has two kinds of classification agents: classification by type of source and by type of document. In the first case the source of a document is determined by its URL. This is done by a description file consisting of metainformation about the source. For instance, given the URL http://pcfolini.eng.unipr.it it is not evident to the user that this URL is a source from the scope of 'CAD-CAM Tools', but with the help of the description file the user sees 'Resources for CAD-CAM Tools' as the server type.
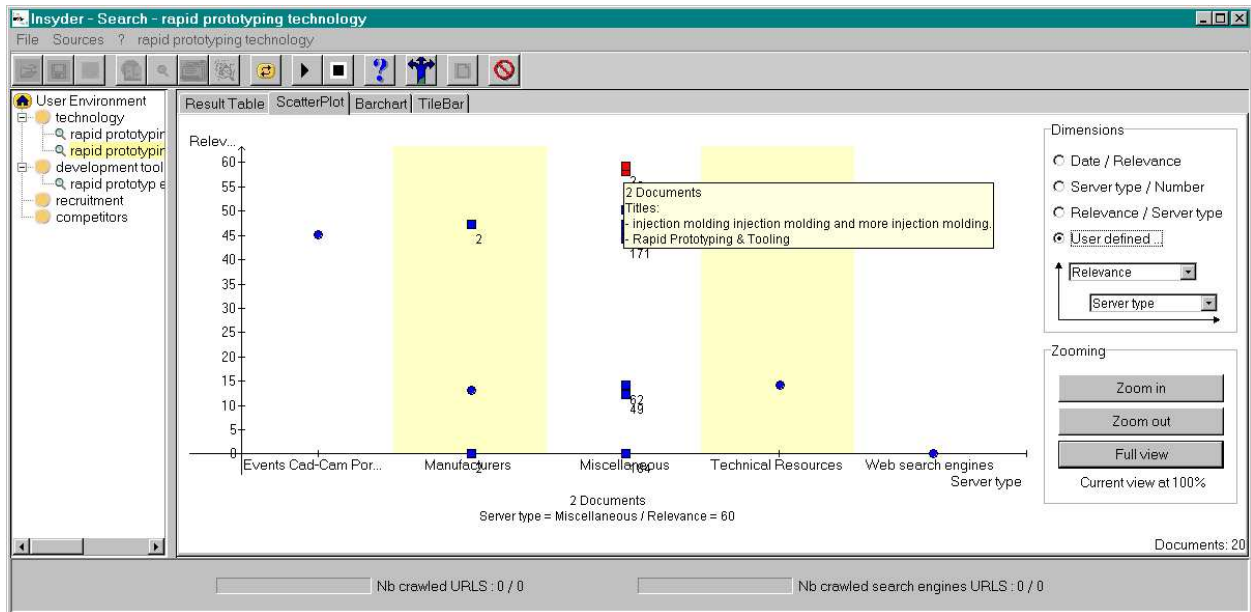
Figure 2: Classification view of INSYDER

The second type of classification is the classification by document type. In this case formal criteria are used to determine if a document is a webliography (a list of links), an entry-point (e.g. like Yahoo) or other text and image information. The output of the classification agents is the input for various kinds of visualisation. Figure 2 is showing a classification view using a Scatterplot.

The results of the crawling and classification agents are a substantial input for the **visualisation agents** which helps the user to analyse the documents retrieved in various ways and which are a substantial assistant in this review of results phase. We decided to use a combined approach, as from the literature can be determined that there is no best visualisation. Our approach offers users the possibility to choose the most appropriate visualisation for their current demand. Different systems in other application domains also follow this approach [1], [14]. But there are also some drawbacks: The user interface of the system becomes more complex and therefore will be harder to use, the user can choose an inappropriate visualisation for a specific situation and others. To intercept the possible drawbacks a number of guidelines have been considered, which are described in [19]. The followed approach initially had six different visualisations grouped around the traditional result list, as a familiar entry-point for the user. Adapting the components to

each other in colour, orientation and the overall style as far as possible, mock-ups and prototypes had been developed using a Vectorplot, a Scatterplot, Barcharts, Tilebars, Relevance Curves and Thumbnail views. Each of them offering selected information and a somewhat new viewpoint for different levels of details: from the document set to the single document. Another important point is the synchronisation of the visualisations: every selection in one representation of the result set will be updated immediately in the other representations, too. This approach has many similarities with "Multiple Coordinated Views" [23].

Due to the results of user evaluations only four of the initially six visualisations are contained in the current version of INSYDER. Despite their potential value Thumbnail views dropped out because of crawling demands and technical implementation reasons. The Vectorplot evolved in different steps from the Document Spiral idea [6] has finally been integrated in the Scatterplot as special case.

3.1.3.1   Result List

The result list view is the common view users know. The figure below shows an example for the query *rapid prototyping technology* and its results. On the left there are the different SOIs, while on the right the user sees the result table and a browser, which shows a preview of the actual highlighted document .
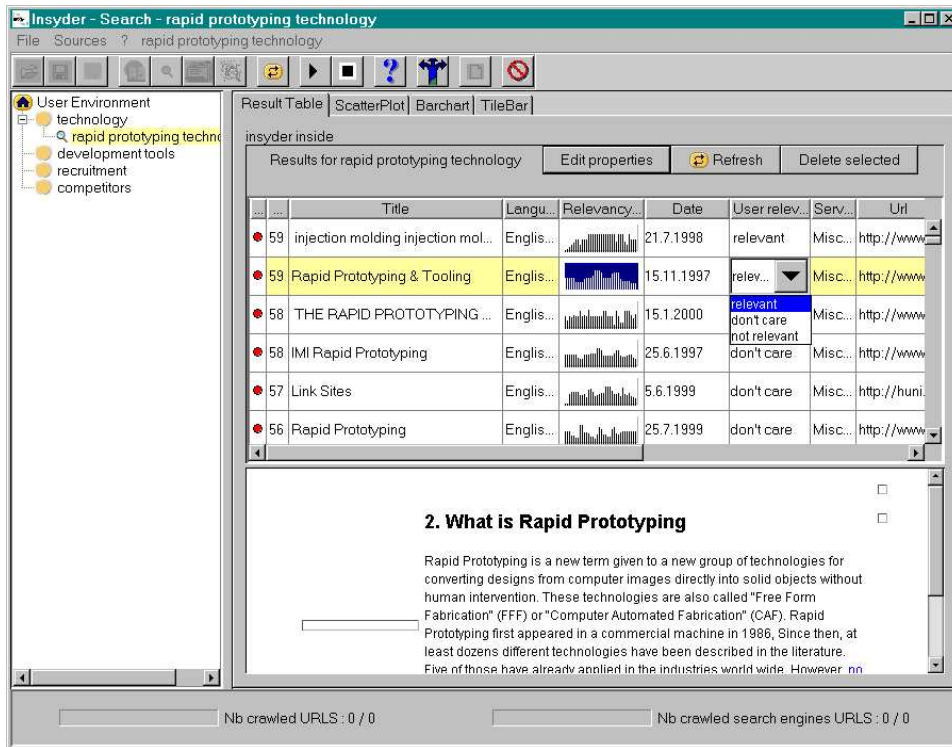
4

Figure 3: Result Table view

3.1.3.2 Scatterplot

The main goal of the Scatterplot is to give the user impressions about the distribution of the document set found, analysed and rated by the agents of the system. In the Scatterplot view two variables can be shown at the same time. On one hand there are the default choices for the user including relevance versus age of document and servertype versus number of documents (Vectorplot). On the other hand there is the possibility for the user to choose the dimensions himself, e.g. relevance for keyword A versus keyword B.

Figure 2 shows the Scatterplot. Here the user has chosen the dimension himself with the relevance on the y-axis and the servertype classification on the x-axis. In the example it can be seen that the most relevant documents of the query come from manufacture servers, CAD-CAM portals and others which are not specified in detail (miscellaneous). The documents are grouped into 5 categories. The fact that a hit is represented as a square-box shows that this is a document group. This can also be seen from the tooltip, showing the amount of documents and the titles of each document in the group (see Figure 2). If only one document is in the focus of the mouse, than the tooltip shows document features, like title, size, date, category and an abstract. Groups, or any other interesting documents, can be marked with the mouse. The selection will then be highlighted (selected documents are represented in red, while unselected are blue) in this and all the others views, including the traditional list. The selection can be changed in all views. In Figure 2 two document groups are selected.

3.1.3.3 Barchart

The Barchart view shows overall relevance and single relevance for each entered keyword. The original idea of Barcharts [32] is adapted in several ways. First, to have the same way of displaying the documents like in the other views where document details are given, the Barchart is rotated 90 degrees: top down instead of from right to left. Secondly the impression of a document as an entity is emphasised using Gestalt principles, without disturbing the keyword orientation too much. The colours used are the same for Tilebars and Enhanced Relevance curve.

Figure 4 shows the same document collection as the Scatterplot view. The red dots a the beginning of each line symbolise that these documents have been selected (in the former Scatterplot view). From the visualisation the user can see that the first document in the view seems to be the most relevant one, as all three keywords that have been searched for appear in the document with a high relevancy. As described above also in this view documents can be de-or selected.

3.1.3.4 Tilebars

Tilebars [11] have been integrated into the system to support the user in judging the potential value of a document for his demands at a glance. In contrast to the Scatterplot and Barchart, this view is clearly targeted to the single document perspective. Figure 5 shows six documents from a search with five keywords. As shown by the blue and red tiles in the selected documents there is most of the time a co-occurrence of "rapid" and "prototyping". Whereas the agents can handle the single terms "rapid", "prototyping" and the concept "rapid prototyping" at the same time without problems, we

5

have difficulties here to show the concept relevancies. Nevertheless the user gets information about lengths of the documents (shown by length of the tilebars) and distribution of keyword-occurrence in the document. In the 5th document it can be easily seen, that all five keywords appear in the same part of the document. The 6th document has only three out of the five keywords. Additional functions planned in this visualisation is the implementation of the jump-feature for quick-jumps to the document-parts represented by the tiles.
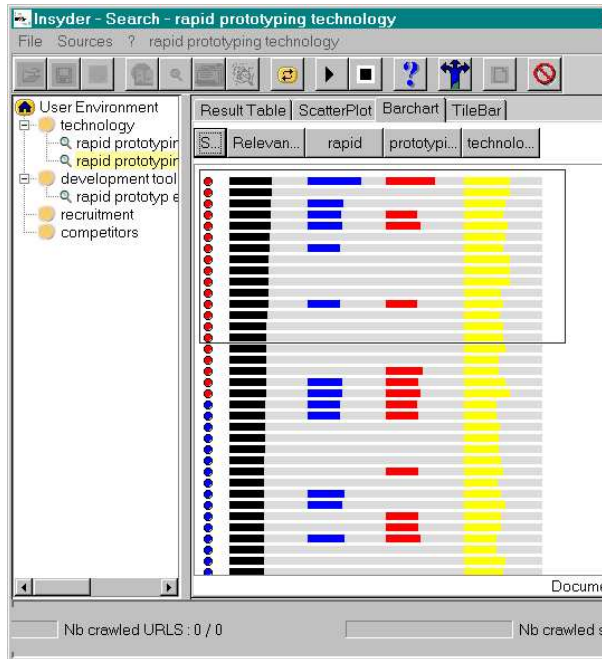

Figure 4: Barchart view

3.1.4 Agents Supporting the Refinement of Queries

INSYDER supports the redefining of the query based on a relevance judgement of the user. This is generally known as relevance feedback and seen to be a beneficial mechanism [16]. The new query terms are generated automatically by the feedback agent based on the documents found by the original search. The relevance feedback is based on the idea to extract concepts from selected documents. The user decide whether they find a document 'relevant' or 'not relevant' (see Figure 3). Only the documents chosen to be relevant or irrelevant are taken into account for the next step, when formulating the new query. This query is built up by the

- Extraction of concepts
- Analysing the concepts to get feature concepts (good and bad ones)
- Creating a document vector with feature concepts giving the good concepts a positive emphasis and the bad ones a negative one.

In this regard a feature concept is any concept from the document describing the document best. The next step for the user is to edit the automatically created query, to modify the sources and to have the query being launched.
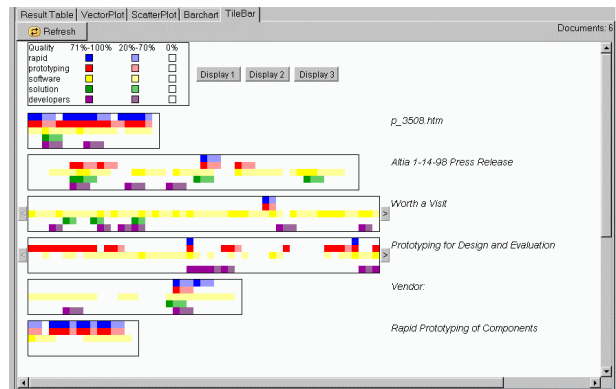

Figure 5: Tilebar view

4 Evaluation

From the start of the project in September 1998 to the mid of November 1999 three user tests with users from small and medium size enterprises (SMEs) in Luton/Great Britain, Nancy/France and Rome/Italy were conducted to test the overall system, especially the user interface and the visualisations of the results. The evaluation followed the method for evaluations of user interfaces, as being proposed by [26]. As a short introduction to the system the users were shown a screencam. Afterwards they had to fulfil different task scenarios (e.g. create a SOI, create a search and analyse the documents found by different visualisations). During the task scenarios the users were requested to "think aloud" to be able to understand and record their current actions. The session was moderated so that in the case of problems the moderator could help. The total number of companies attending the evaluations was 38. 18 companies in Rome, 13 in Nancy and 7 in Luton. The overall number of users was 48. The majority of the participants had good knowledge of Internet. Also some beginners were participating in the test. Each user had 45 minutes to fulfil the tasks.

The user test has shown that the basic idea of the system, giving the user the possibility to create his own user environments, is appreciated. There are some minor problems in handling the current version, which have been solved after the test.

For the evaluation of the visualisation part it is interesting to see that the test users in Rome preferred the Barchart view and the result table as the only visualisation to be presented to the user. The other views (Scatterplot, Tilebars and Relevance Curve) should be presented only optionally. Still when using the Tilebars, it seems to be very important to users, that it is possible to jump immediately to a tile of a document by clicking on it. This seems to be the real added-value of this visualisation.

The Barchart was adopted very well and minor problems occurred while using it. E.g. the users found that it was necessary not only to be able to sort by global quality or by quality of the single keywords, but also by a variable number of keywords. The Scatterplot view was well understood by most of

the subjects. Another very helpful criticism was that a Vectorplot view, which has been intended first, showing on the x-Axis the specific attribute to be analysed (e.g. document length) and on the y-Axis the number of documents fitting in this dimension, could be represented by the Scatterplot by adding a "total number" dimension.

5   Related Work

Excite [8] is a system that makes use of Intelligent Concept Extraction™, so that it is possible to find not only terms occurring in documents but also related terms, e.g. when searching for *cancer* documents not containing *cancer*, but *tumour* would be found, too. As in INSYDER a relevance feedback option is offered (*Search for more documents like this one*). A disadvantage is that the system only takes the current document into account. From the agent point of view different approaches have been undertaken to solve the problem of lacking supply of information. *SAIRE* for example is an example for an agent system that is designed to provide access to Earth and Space Science data over the Internet, giving support for native and expert users [27]. In [20] the authors describe the system *Amalthaea*, which uses information agents for the discovery and filtering of information. The overall system of these agents (e.g. their lifetime, how they do the query formulation) is based on genetic algorithms.

6   Conclusion and Outlook

The Information Assistant INSYDER assists the user in finding relevant business information on the WWW. The system is designed to help users to overcome certain problems when searching for information on the WWW. Therefore information retrieval and information visualisation techniques have been used. A first analysis of user tests conducted in November 1999 resulted in a number of enhancements and showed that the project team is on the right way with this assistance system. For the future further enhancements of the agent and visualisation part are planned. For example it is foreseen to have a clustering feature, by grouping e.g. similar concepts and the implementation of the enhanced Relevance Curve will be finished. Generally spoken it's planned to improve the overall system performance and to have more user tests to evaluate the system. Temporarily another user test is conducted. This time the software is installed at the user's site. By doing this the user have the possibility to test the software independently from given scenarios, in the real world of their everyday tasks. The comments of these users will be a great input for further development.

Acknowledgements

References

[1] Ahlberg, C; Wistrand, E: IVEE: An information visualization and exploration environment. In: Proc. IEEE Information Visualization 95, pp. 66-73.

[2] Arisem S.A., http://www.arisem.com [2000-01-18]

[3] Brenner, W., Zarnekow, R., Wittig, H., unter Mitarbeit von Schubert, C.: Intelligente Softwareagenten: Grundlagen und Anwendungen. Heidelberg (Springer-Verlag) 1997.

[4] Card, S.K.; Mackinlay, J.D.; Shneiderman, B. (Eds.): Readings in Information Visualization. Using Vision to Think. Morgan Kaufmann Publishers, Inc, San Francisco, CA, 1999.

[5] Card, S.K.; Robertson, G.G.; York, W.: The WebBook and the WebForager: An Information Workspace in the World Wide Web. In: Proc. ACM CHI'96, pp. 111-117.

[6] Cugini, J.; Laskowski, S.; Piatko, C.: Document Clustering in Concept Space: The NIST Information Retrieval Visualization Engine (NIRVE). http://zing.ncsl.nist.gov/~cugini/uicd/cc-paper.html [1998-09-10]

[7] Eichmann, D. Ethical Web Agents. In Proc. of the The Second WWW Conference. 1994. http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents/eichmann.ethical/eichmann.html [2000-01-18]

[8] Excite: Information Retrieval Technology and Intelligent Concept Extraction (TM) Searching. o.O. 1996. http://www.excite.com/ice/tech.html [1999-08-25]

[9] Hackathorn R., Web farming for the data warehouse, Morgan Kaufmann, San Francisco, 1998

[10] Handschuh, S.; Mann, T.M.; Mussler, G.; Reiterer, H.: Die Entwicklung eines Business Intelligence Systems zur Beschaffung von Geschäftsinformationen im WWW. UVK (Konstanz). 1999. S.171-182

[11] Hearst, M. A.: TileBars: Visualization of Term Distribution Information in Full Text Information Access. In: Proc. ACM CHI'95, pp. 59-66.

[12] Hearst, M. A.: User interfaces and visualization. In: Baeza-Yates, R.; Ribeiro-Neto, B. (ed.), Modern Information Retrieval. Addison-Wesley (New York), 1999.

[13] Hearst, M. A.: The use of categories and clusters for organizing retrieval results. Aus: Strzalkowski, Tomek (Hrsg.): Natural Language Information Retrieval. Dordrecht (Kluwer Academic Publisher) 1999. S. 333-374.

[14] Henninger, S.; Belkin, N.J.: Interface Issues and Interaction Strategies for Information Retrieval Systems. In: Proc. ACM CHI'96. pp. 352-353

[15] http://www.software.ibm.com/data/pubs/papers/bisolution/index.html [2000-01-18]

[16] Koenemann, J.; Belkin, N. J.: A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness. In: Bilger, R.; Guest, S.; Tauber, M. J. (Hrsg.): CHI 96 - Electronic Proceedings. http://www.uni-paderborn.de/StaffWeb/chi96/ElPub/WWW/chi96www/papers/Koenemann/jk1_txt.htm [1999-11-11].

[17] Kuhlen, R.: Die Konsequenzen von Informationsassistenten. Was bedeutet informationelle Autonomie oder wie kann Vertrauen in elektronische Dienste in offenen Informationsmärkten gesichert werden? Frankfurt (Suhrkamp-Verlag). 1999

[18] Mann, T. M.: Visualization of WWW-Search Results. In: Wagner, R. (Ed.): Tenth International Workshop on Database and Expert Systems Applications, Florence, Italy, August 29-September 3, 1999, Proceedings IEEE Computer Society, 1999.

[19] Mann, T. M.; Reiterer, H.: Case Study: A Combined Visualization Approach for WWW-Search Results. Aus: Gershon, N.; Dill, J.; Wills, G. u.a. (Hrsg.): IEEE Information Visualization Symposium 1999 Late Breaking Hot Topics Proceedings. Supplement to: Will, G.; Keim, D. (Hrsg.): Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99). Conference: San Francisco, CA, USA, October 24-29, 1999. Los Alamitos, CA (IEEE Computer Soc. Press) 1999. San Francisco 1999. S. 59-62.

[20] Moukas, A.; Maes, P.: Amalthaea: An Evolving Multiagent Information Filtering and Discovery System for the WWW. In: Journal of Autonomous Agents and Multi-Agent Systems http://lcs.www.media.mit.edu/~moux/papers/jaamas98.pdf [1999-08-26]

[21] Mußler, G.: Ein Agentensystem zur Unterstützung bei der Informationssuche im WWW. http://wwwdb.informatik.uni-rostock.de/adi99/endversionGM.ps [1999-10-05]

[22] Nielsen, J.: Search and You May Find. http://www.useit.com/alertbox/9707b.html [1999-03-18]

[23] North, C.; Shneiderman, B.: Snap-Together Visualization: Coordinating Multiple Views to Explore. University of Maryland, technical report CS-TR-4020 June 1999

[24] Pazzani, M.; Muramatsu, J.; Billsus, D.: Syskill & Webert: Identifying interesting web sites. Aus: (Hrsg.): Proceedings of the National Conference on Artificial Intelligence. Portland, OR 1996. http://www.ics.uci.edu/~pazzani/RTF/AAAI.htm [1999-03-26]

[25] Pollock, A.; Hockley, A.: What's Wrong with Internet Searching. In: D-Lib Magazine, 1997, http://www.dlib.org/dlib/march97/bt/03pollock.html [1999-02-01]

[26] Redmond-Pyle, D.; Moore, A.: Graphical User Interface Design and Evaluation (GUIDE) - A practical process. Prentice Hall (London). 1995.

[27] http://saire.ivv.nasa.gov/saire.html [2000-01-18]

[28] Shneiderman, B.; Byrd, D.; Croft, W. B.: Clarifying Search: A User-Interface Framework for Text Searches. In: D-Lib Magazine, 1997, http://www.dlib.org/dlib/january97/retrieval/01shneiderman.html [1999-08-17]

[29] Shneiderman, B.; Maes, P.: Direct Manipulation vs. Interface Agents. Excerpts from debates at IUI 97 and CHI 97 In: interactions, 6. Jg. (1997), H. november+dezember, S. 42-61.

[30] Smeaton, A. F.; Crimmins, F.: Relevance Feedback and Query Expansion for Searching the Web: A Model for Searching a Digital Library. Aus: Pisa, P. C.; Thanos, C. (Hrsg.): Research and Advanced Technology for Digital Libraries, First European Conference, ECDL'97 (Springer) 1997.

[31] Turau, V.: Web-Roboter. In: Informatik Spektrum, 21. Jg. (1998), H. 3, S. 159-160.

[32] Veerasamy, A.; Navathe, S.B.: Querying, Navigating and Visualizing a Digital Library Catalog. In: Proc. DL'95. http://www.csdl.tamu.edu/DL95/papers/veerasamy/veerasamy.html [1999-03-24]

[33] Voorhees, E. M.: Query Expansion using Lexical-Semantic Relations. Aus: Croft, B. W. (Hrsg.): Proceedings of the seventeenth annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '94, 3-6 July 1994, Dublin, Ireland London, Berlin u.a. (Springer) 1994. S. 61-69.

[34] Voorhees, E. M.; Harman, D. K. (Hrsg.): NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC-7) Gaithersburg, Maryland (Government Printing Office (GPO)) 1998. http://trec.nist.gov/pubs/trec7/t7_proceedings.html [1999-12-20]

[35] Wooldridge, M.; Jennings, N. R.: Intelligent Agents: Theory and Practice. In: Knowledge Engineering Review, 10. Jg. (1995) http://www.elec.qmw.ac.uk/dai/people/mikew/pubs [1999-11-23]

[36] Zamir, O.; Etzioni, O.: Web Document Clustering: A Feasibility Demonstration Aus: (Hrsg.): SIGIR 1998. http://zhadum.cs.washington.edu/zamir/sigir98.ps [1999-03-23]