# INSYDER - An Information Assistant for Business Intelligence

Harald Reiterer, Gabriela Mußler, Thomas M. Mann, Siegfried Handschuh
University of Konstanz,
*Harald.Reiterer, Gabriela.Mussler, Thomas.Mann, Siegfried.Handschuh*
*@uni-konstanz.de*

**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 studies 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 systems [15] 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 [9] proposes such an approach called Web farming: *"Web farming is the systematic refining of information resources on the Web for business intelligence."*

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.



## 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.

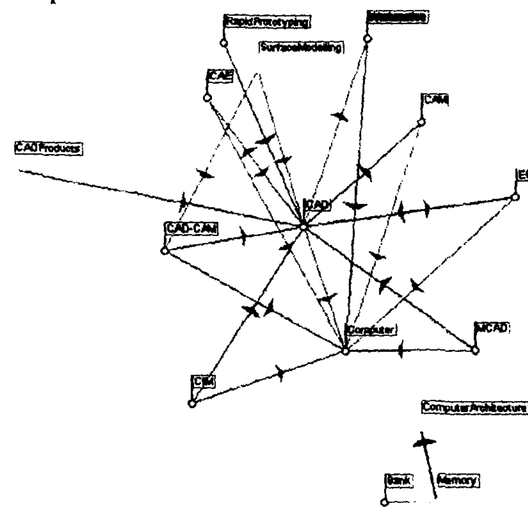


Figure 1: INSYDER semantic network [2]

### 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 to 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.

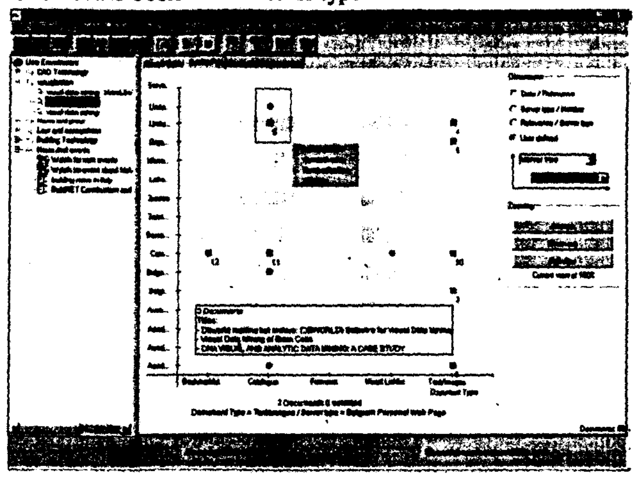

Figure 2: Classification view of INSYDER

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 bookmarklist (a list of links to external servers), a catalogue (list of links to the same server), a mixed linklist (not a bookmarklist, not a catalogue), a frameset 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 (see Figure 1) is the common view users know. The figure below shows an example for the *visual data mining* 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

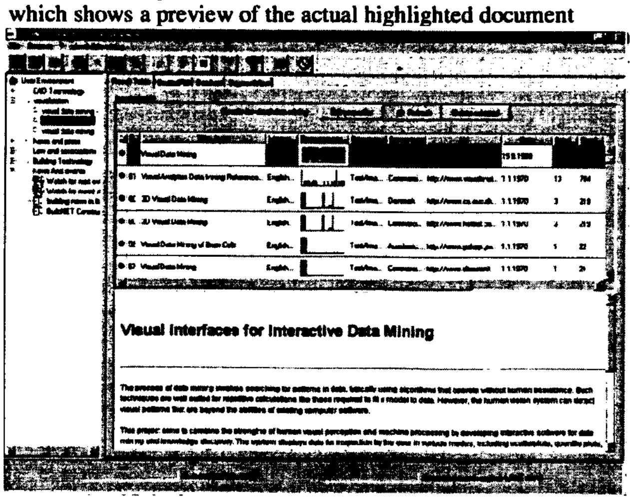


Figure 3: Result Table view

#### 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 classification of server types on the y-axis and the document type classification on the x-axis. 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 documents in the view seem 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 documents from a search with the three keywords. Whereas the agents can handle the single terms "data", "mining" and the concept "data mining" at the same time without problems, we 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. As shown by the blue, red and yellow tiles in the selected first document (title: Visual Data Mining) there are three segments with a co-occurrence of "visual" and "data" and "mining". If the user puts the mouse pointer over a segment a tool tip occurs showing the text of this segment. A jump-into-segment feature for quick-jumps to the document-parts represented by the segments is also available.

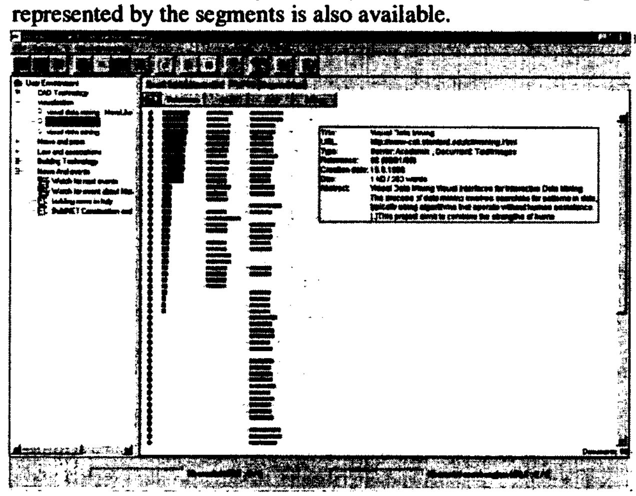

Figure 4: Barchart view

### *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 the system should 'find similar' or 'don't find similar'. 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.

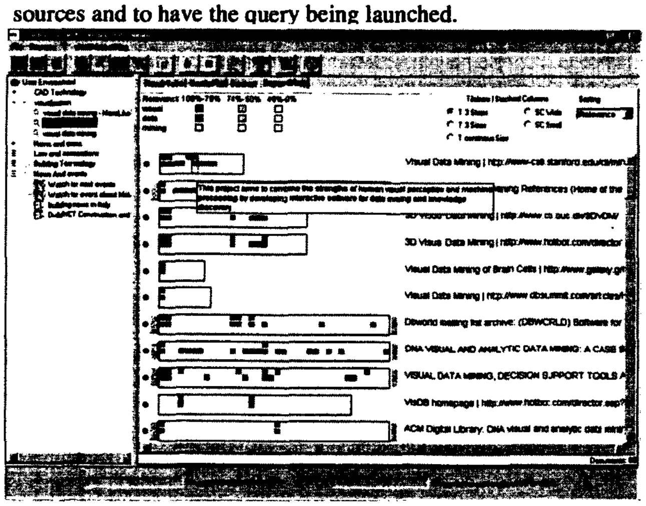

Figure 5: TileBars view

# 4 Evaluation of the Visualisation Views

## 4.1 Formative Evaluation

During the EU project (September 1998 – March 2000) 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 search results. The evaluation followed the method for evaluations of user interfaces, as being proposed by Redmond-Pyle et al. [26]. As a short introduction to the system the users were shown a ScreenCam™ movie. Then the user had to accomplish different test tasks (e.g. create a sphere of interest, launch a search and analyse the documents using different visualisations). During the test tasks the users were requested to "think aloud" to be able to understand and record their current actions. The recording of data was done with ScreenCam™ movies (user interactions and spoken comments) and with written records. The session was moderated so that in the case of problems the experimenter 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 the Internet. Also some beginners were participating in the test. Each user had 45 minutes to fulfil the tasks.

The user tests have shown that the basic idea of the system, giving the user the possibility to create his user environments and sphere of interests, is appreciated. An interesting result of the evaluation of the visualisations was that the test users in Rome preferred the Barchart view and the Result Table. The other views (Scatterplot, TileBars and Stacked Column) should be presented only optionally. When using the TileBars, it was very important to the users that it was possible to jump immediately to a tile of a document by clicking on it. The Barchart was adopted very well and minor problems occurred while using it. The Scatterplot view was well understood by most of the users. Another very helpful feedback 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 integrated in the Scatterplot by adding predefined views.

## 4.2 Summative Evaluation

After these formative evaluations during the development of the system, which gave us a lot of helpful hints to improve the system, we have made a summative evaluation at the University of Konstanz (February - April 2000)[1]. These evaluations mainly focus on the different visualisations.

The primary goal of this summative evaluation was to measure the added value of our visualisations (Scatterplot, Barchart, TileBars, Stacked Column) in terms of effectiveness, efficiency and subjective satisfaction for reviewing Web search results. Knowing advantages of the multiple view approach documented in user studies [23], we didn't intend to measure the effects of having Scatterplot, Bargraph and TileBars/Stacked Column (also called SegmentView) instead of the List and Table. We wanted to see the added value of having these visualisations in addition to the Table and List.

Another goal of this summative evaluation was to measure the influence of three of the four factors (*target user group, type and number of data, type of task*) on the effectiveness, efficiency, and user satisfaction for each.

### *4.2.1 Independent Variables*

*User Interface*: The following user interface configurations have been tested:

- List only
- Table only
- Scatterplot + Table
- Bargraph + Table
- TileBars/Stacked Column (SegmentView) + Table.

From the four factors influencing effectiveness, efficiency and user satisfaction when using visual structures we decided to vary *target user group, type and number of data,* and the *task* to be done. The fourth factor, the *technical environment* was identical for all tests done by using two identical 400 MHz Pentium III PCs, running Windows NT 4.0 SP5 with 256 MB RAM, identical software-configuration, hard disk and 21" monitors.

*Target user group:* To see possible influences caused by the target user group we used two different groups of users. The first group, called "beginners" consisted of 20 female or male students or staff of our department of information and computer science, and a number of non IT-related disciplines. The users classified as beginners all knew the Web and had some limited understanding of search engines, but no deeper knowledge about information retrieval techniques. The second group, here referenced as "experts", with also 20 female and male members consisted of students or staff of our department. All members of

[1] After the official EU-project has been finished in March, the University of Konstanz is still working on the Insyder system in close relation to the French company Arisem.

this group had at least participated in one course on information retrieval and had extensive Web search experience.

*Type and number of data:* To see possible influences caused by the type and number of data, we used queries with three different numbers of keywords (1, 3, 8) and two different sizes of result sets (30 or 500 hits). An additional effect in these dimensions was the quite heterogeneous content of the result sets, which had been prepared by searching the Web with different keywords for 12 topics.

*Task:* To see possible influences caused by the task to be done, we decided to use two of the four different types of information seeking tasks described in [28]. Half of the tasks the users had to fulfil were of the type "specific fact finding"; the other half were of the type "extended fact finding". The main difference between these two types of tasks is that in the first case, there is a clear abort criterion, when the user finds a document to answer the questions. In the second case there is no such clear abort criterion to stop the examination of the result set, and therefore the result investigation process will be much broader and possibly of longer duration.

An example for a test tasks with combinations of type/number of data and task to be done in the field of specific fact finding is: the 1 keyword query "danube" presented 30 hits and the indented information seeking task to find out: "How long is the Danube river?". An example of an extended fact-finding test tasks is: the 3 keyword / 30 hits query "john irving book" with the indented information seeking task "Which books had been written by the author John Irving?".

We decided to eliminate all documents from the result set, which would allow completing the extended fact finding tasks using a single document. Doing this, we ensured that the task would be of the type extended fact finding. This did not influence the size of the result sets, because when eliminating a document from the set presented to the users, it was substituted by the first document not included so far.

The test setting covered all combinations of the above described different kinds of information seeking tasks, different kinds of users, amount of results, number of keywords of each query, and the chosen combinations of different visualisations (see Table 1). E1 to E5 represents the 20 experts divided into the five groups of user interfaces; B1 to B5 represent the 20 beginners divided also into 5 groups. Each group number (e.g. E1 + B1) got the same configuration, so from the group variation level each group consists of 8 test users (4 experts, 4 beginners).

#### 4.2.2 Dependent Variables

*Task completeness (effectiveness):* Completeness with which users achieved the goals of the test tasks (e.g. number of books found written by John Irving). The effectiveness was calculated by relating the found answer by the test user to the number of possible correct answers in the concrete result set (e.g. if 12 books by John Irving could be found in the result set and the user did find 7, his effectiveness was recorded as 58%)

*Task performance time (efficiency):* Time to complete each test task, not including reading the task question. In order not to exceed the overall test time per user much more than two hours, the time to answer specific fact finding questions was limited to 5 minutes per question, for extended fact finding tasks to 10 minutes per question.

*User subjective acceptance (satisfaction):* Test users rate their satisfaction in the following categories: ease of use, self-descriptiveness, suitability for learning, layout, suitability for the tasks, and conformity with user expectations.

| Kind of task | Kind of user | No of docs | No of keywords | List | Table | Scatter-plot + Table | Bar-graph + Table | TileBars/ Stacked Column + Table |
|---|---|---|---|---|---|---|---|---|
| Specific fact finding | Beginner (B) | 30 | 1 | B1 | B2 | B3 | B4 | B5 |
| | | | 3 | B2 | B3 | B4 | B5 | B1 |
| | | | 8 | B4 | B5 | B1 | B2 | B3 |
| | | 500 | 1 | B5 | B1 | B2 | B3 | B4 |
| | | | 3 | B1 | B2 | B3 | B4 | B5 |
| | | | 8 | B3 | B4 | B5 | B1 | B2 |
| | Expert (E) | 30 | 1 | E1 | E2 | E3 | E4 | E5 |
| | | | 3 | E2 | E3 | E4 | E5 | E1 |
| | | | 8 | E4 | E5 | E1 | E2 | E3 |
| | | 500 | 1 | E5 | E1 | E2 | E3 | E4 |
| | | | 3 | E1 | E2 | E3 | E4 | E5 |
| | | | 8 | E3 | E4 | E5 | E1 | E2 |
| Extended fact finding | Beginner (B) | 30 | 1 | B2 | B3 | B4 | B5 | B1 |
| | | | 3 | B4 | B5 | B1 | B2 | B3 |
| | | | 8 | B5 | B1 | B2 | B3 | B4 |
| | | 500 | 1 | B3 | B4 | B5 | B1 | B2 |
| | | | 3 | B5 | B1 | B2 | B3 | B4 |
| | | | 8 | B1 | B2 | B3 | B4 | B5 |
| | Expert (E) | 30 | 1 | E2 | E3 | E4 | E5 | E1 |
| | | | 3 | E4 | E5 | E1 | E2 | E3 |
| | | | 8 | E5 | E1 | E2 | E3 | E4 |
| | | 500 | 1 | E3 | E4 | E5 | E1 | E2 |
| | | | 3 | E5 | E1 | E2 | E3 | E4 |
| | | | 8 | E1 | E2 | E3 | E4 | E5 |

Table 1: Combination of test tasks

#### 4.2.3 Procedure

The evaluation was focused on the activities done in the *reviewing of the results phase* of the four-phase framework of information seeking (see Chapter 2.1). To avoid side effects caused by the activities in the formulation and action phase, the evaluation was done with already prepared searches for 12 topics / questions. For each predefined query the test users had to answer a question representing the intended information-seeking task. To avoid side effects caused by the refinement step the INSYDER system had been modified in a way that all functions, which allow refinement steps other than view transformations, had been suppressed. So the users could use functions like zoom or mark /unmark documents, but they didn't see functions the INSYDER system normally provides, like generating new queries using relevance feedback or re-ranking existing result sets by changing, adding or deleting keywords.

The users were told to answer the questions as quick as possible. All users processed the same 12 questions, with the same keywords and number of hits in the same order, beginning with the 1 keyword / 30 hits specific fact finding task "danube" described above, followed by an extended fact finding 3 keyword / 500 hits question, then followed by a specific fact finding task with 8 keywords / 30 hits, and so on, always alternating specific fact finding and extended fact finding, with the number of keywords changing between every question in the order 1 – 3 – 8 – 1 – 3 – 8 … . The difference between the five groups was the visualisations the user could use to answer the questions (see e.g. the highlighted fields in Table 1 for B1). Based on these plannings of the controlled experiment we could assure that the five combinations of visualisations have been distributed in an equal manner to all variables.

After fulfilling an entry questionnaire with six questions (e.g. age, computer and software experience), the users got a short introduction in the INSYDER system with the help of a ScreenCam™ movie demonstrating and explaining the main concepts and visualisations of the system. Then each user had some time to get to know the system with a test result set and all five visualisations. After finishing this introduction phase the users had to accomplish the 12 test tasks. During the tasks the users were requested to "think aloud" to enable the evaluation team to understand and record their current actions. The recording of data was done by two persons taking written records. An experimenter moderated the test session so that in the case of problems this person could help. After accomplishing the test tasks the user had to answer a questionnaire of 30 questions regarding their subjective satisfaction.

#### 4.2.4 *Results*

The final interpretation of the results will be finished in June. The following results are based on an interim report prepared after finishing the 40 test-sessions and should only give a first impression, because the main statistical analyses still have to be done.

*Added values of the visualisations.* The majority of the users expressed a high satisfaction about the visualisations. Especially the TileBars/Stacked Column (Segment) view got high positive ratings. This subjective impression seems not to be fully supported by the hard facts. Looking on the overall results for task completeness (effectiveness) and task performance time (efficiency) we got the following results:

Considering the average efficiency and the average effectiveness of all factors (independent variables), the List performed slightly best (Ø efficiency: 4:59 min for List, 5:12 min for Result Table, 5:15 min for Barchart, 5:26 min for Segment view, 5:38 min for Scatterplot; Ø effectiveness: 66% for List, 65% for Result Table, 63% for Scatterplot, 62% for Segment View, 60% for Barchart). This may be an effect of experience. People are used to this visualization of search results, and unfortunately our evaluation setting did not allow to examine the effect of training.

When we are talking about results e.g. for "Scatterplot", we correctly should talk of "Scatterplot + Table configuration". Because for the Scatterplot, Bargraph and TileBars/Stacked Column users had also the Result Table as additional view available. Some of the test users had been really visualisation-resistant. One expert and one beginner never used anything else but the List or Result Table. So to get trends it's not enough to see how much time the users needed to answer the question and what level of completeness they reached, but also to see which possibilities they used. For specific fact finding tasks the Segment View setting was the best configuration after the List. For extended fact finding the Segment View configuration was the worst configuration. The List was clearly the "best" visualization, both in terms of average efficiency and average effectiveness. For the other visualizations, this picture is not so clear, because ranking positions are sometimes different for efficiency and effectiveness.

## 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 (e.g. visual query interface, clustering feature). Generally spoken it's planned to improve the overall system performance and to have more user tests to evaluate the system.

### Acknowledgements

The authors wish to thank the following EU-ESPRIT project partners for their contributions and support: Alain Garnier, Olivier Spinelli, Laurent Dosdat (ARISEM, Paris); Guillaume Lory, Carlo Revelli (Cybion, Paris); Rina Angeletti (Innova, Rome); Flavia D'Auria (Promoroma, Rome).

### References

[1] C. Ahlberg and E. Wistrand. IVEE: An information visualization and exploration environment. *Proc. IEEE Information Visualization 95*: 66-73, 1995.

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

[3] W. Brenner, R. Zarnekow and H. Wittig. *Intelligente Softwareagenten: Grundlagen und Anwendungen*. Heidelberg (Springer-Verlag) 1997.

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

[5] S. K. Card, G. G. Robertson and W. York. The WebBook and the WebForager: An Information Workspace in the World Wide Web. *Proc. ACM CHI'96*: 111-117, 1996.

[6] J. Cugini, S. Laskowski and C. Piatko. 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] D. Eichmann. Ethical Web Agents. *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] R. Hackathorn. *Web farming for the data warehouse.* Morgan Kaufmann, San Francisco, 1998 .

[10] S. Handschuh, T.M. Mann, G. Mussler and H. Reiterer. Die Entwicklung eines Business Intelligence Systems zur Beschaffung von Geschäftsinformationen im WWW. *Information Engineering.* R. Kuhlen, W. Semar (eds.). UVK (Konstanz):171-182 , 1999.

[11] M. A. Hearst. TileBars: Visualization of Term Distribution Information in Full Text Information Access. *Proc. ACM CHI'95*: 59-66, 1995.

[12] M. A. Hearst. User interfaces and visualization., *Modern Information Retrieval.* R. Baeza-Yates and B. Ribeiro-Neto (eds.). Addison-Wesley (New York): 257-323, 1999.

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

[14] S. Henninger and N. J. Belkin. Interface Issues and Interaction Strategies for Information Retrieval Systems. *Proc. ACM CHI'96*: 352-353 , 1996.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[34] E. M. Voorhees and D. K. Harman (eds.): *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] M. Wooldridge and N. R. Jennings. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, 10, 1995 http://www.elec.qmw.ac.uk/dai/people/mikew/pubs [1999-11-23].

[36] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. *SIGIR 1998.* http://zhadum.cs.washington.edu/zamir/sigir98.ps [1999-03-23].