The CircleSegmentView:
A User Centered, Meta-data Driven Approach for Visual Query and Filtering

Dissertation

zur Erlangung des akademischen Grades
des Doktor der Naturwissenschaften (Dr.-rer. Nat.)
an der Universität Konstanz, Mathematisch- Naturwissenschaftliche Sektion

vorgelegt von

Peter Klein

Tag der mündlichen Prüfung: 02.06.2006
Referent: Prof. Dr. Harald Reiterer
Referent: Prof. Dr. Daniel Keim

Acknowledgement

From the many people who contributed in one way or another to me finishing this project, I have to thank first and foremost my adviser, Prof. Dr. Harald Reiterer, who gave me the chance to work and do research on visualization in his team at the University of Konstanz. During the previous four years, he motivated, stimulated and supported me in this finally successful project. Without him this thesis would never have been possible. My thanks also to Prof. Dr. Daniel Keim and Prof. Dr. Rainer Kuhlen from the University of Konstanz for their interest in my work and for taking the roles of second / third referees.

From my colleagues I have to thank especially Frank Müller, Tobias Limbach, Torsten Grust and Georg Odenthal. They provided a unique environment (inside and outside of the university) with patience, ideas, help and entertaining coffee breaks. I want to thank Alina Bey and Christian Grün, who contributed a lot of effort in the implementation of the CircleSegmentView.

From the other members of the INVISIP team I want to thank Stefan Göbel and Jörg Haist in particular. They were the heart and soul of the INVISIP project - without them it certainly would not have succeeded.

Special thanks to Michael Dengel, Anne Haeming, Bob White and a number of other people for reading preliminary versions of the thesis and providing me with helpful advice.

Many thanks also to Prof. Dr. Harald Zimmermann and Prof. Dr. Wolfgang J. Paul of the University of Saarbrücken, who inspired me with their work, attitude and energy.

Last but not least, I want to thank Stephanie for years of love, patience, and support.

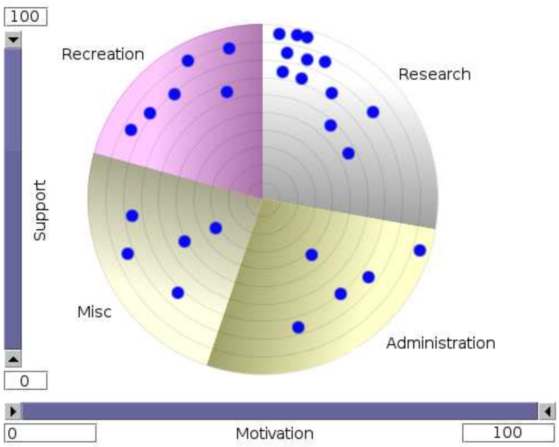


***Figure 1**: CircleSegmentView visualizes the acknowledgements*

”Our heads are round so that the thoughts can change their direction. / Unser Kopf ist rund, damit das Denken die Richtung ändern kann. ”

Francis Picabia

Zusammenfassung

Kurzzusammenfassung

Diese Arbeit entstand im Rahmen eines EU IST Projektes mit Namen INVISIP. Es handelte sich um ein Forschungsprojekt mit weitreichenden und differenzierten Zielen im Bereich der Geo-Metadaten. Der Aufgabenbereich der Arbeitsgruppe Mensch-Computer Interaktion der Universität Konstanz befate sich mit der Entwicklung eines Visuellen Metadaten Browsers, dem benutzerzentrierten Entwicklungsproze, sowie der Erforschung kontextbezogener Bewertungsmechanismen für Metadaten.

Meine Forschungen spiegeln diese Aufteilung in meiner vorliegenden Dissertation wider. Während der Schwerpunkt im Bereich 'Informationsvisualisierung' liegt, werden auch die Bereiche des 'Usability Engineerings' und des 'Information Retrieval' bearbeitet.

Ziel meiner Forschung war und ist es dem Nutzer von (visuellen) Suchsystemen ein Maximum an Information auf effizienten und effektiven Weg zur Verfügung zu stellen. Zielgerichtete Suche als auch explorierendes Verhalten soll für den Benutzer möglichst intuitiv geboten werden. Durch diese Zielsetzung ist ein benutzerzentrierter Entwicklungsansatz unumgänglich. Darüberhinaus muss ein Optimum an vorhandenem Wissen über die Daten dem Benutzer zur Verfügung gestellt werden, damit dieser seine Entscheidungen mit dem bestm¨glichen Wissenschatz trifft.

Als Resümee meiner Arbeit steht der Prototyp einer Visualisierung, welcher für dynamische Anfragen an ein visuelles Suchsystem als auch als visueller Filter bei der Ergebnisvisulisierung eingesetzt werden kann. Darüberhinaus wird eine Vorgehensweise für die Entwicklung visueller Suchsysteme skizziert und dokumentiert. Letzlich wird in dieser Arbeit auch ein Bewertungsschema für Metadaten Informationssysteme vorgestellt, welches Suchverhalten von Nutzern mit deren Hintergrundwissen verbindet und zu effizienterern L¨sungen führt.

Short Abstract

This work is embedded in an EC funded project named INVISIP. This project covers a diversity of different goals concerning the field of geo-meta-data. The work packages for the human-computer interaction group of the university of Konstanz dealt with the development of a visual meta-data browser, the user centered design process and the exploration of meta-data ranking schema.

My research reflects these separate work packages. Despite the fact that the main part of my research is covered by the field of 'information visualization', it also includes large parts of research in the fields of 'usability engineering' and 'information retrieval'.

The goal of my research was (and still is) to offer the user of visual information seeking systems a maximized set of relevant data, using the most efficient and effective way. Both straight search and exploring should be supported in a satisfying and enjoyable approach. Therefore a user-centered design process is inevitable. Beyond this, an optimum must be known about the data (so called meta-data) and be offered to the user, so that user's decision-making is based on the best possible knowledge.

As a rsum of my work ther is the prototype of a visualization that is capable of offering dynamic query features as well as visual filter facilities. Additionally I present a model for the effcient implementation of visual information-seeking systems based upon meta-data and a user-centered design process. This research is supplemented by a ranking scheme for meta-data, which takes the user's knowledge as well as the user's search behavior into account. This will lead to more effective and more efficient information-seeking systems.

Contents

List of Figures

List of Tables

1 INTRODUCTION

Contents

1.1 Information Needs

Some of the main challenges of the Web are problems related to the user and his interaction with an information-retrieval system. People with different educational, social, cultural and economic backgrounds use the internet to gather information or to search for knowledge. Information-seeking systems like Web Search Engines should help the users with this task. The information-seeking process is covered in computer science by various fields of research:

- Usability Engineering
- Information Visualization
- Information Retrieval

Using research results from the above fields is a good starting point for the creation of an effective and efficient information-seeking system. Research results from 'neighboring' fields of research like cognition psychology should add up to a more reliable system. The basic terms used in this work are defined as follows.

Definition 1.1 (Usability engineering) *is a well defined process which is performed as part of the application development process. It can be part of the development process of any type of electronic information application. Although each development project is different, the approaches, methods, techniques and activities applied to achieve usability do not vary much [Pro04].*

Definition 1.2 (User-centred design) *UCD is a highly structured, comprehensive product development methodology driven by: (1) clearly specified, task-oriented business objectives, and (2) recognition of user needs, limitations and preferences. Information collected using UCD analysis is scientifically applied in the design, testing, and implementation of products and services. When rigorously applied, a UCD approach meets both user needs and the business objectives of the sponsoring organization.*[1]

Because the user is in the center of the whole process it is necessary to analyse human search behavior (see Figure 1.1), examine the available data, investigate the possible tasks and finally build a visualization that optimally supports this process.

Definition 1.3 (Information Visualization) *The use of computer-supported, interactive, visual representations of abstract data to amplify cognition [CM99].*

Definition 1.4 (Information Retrieval) *(IR) is part of computer science which studies the retrieval of information (not data) from a collection of written documents. The retrieved documents aim at satisfying a user information need usually expressed in natural language [BYRN99].*

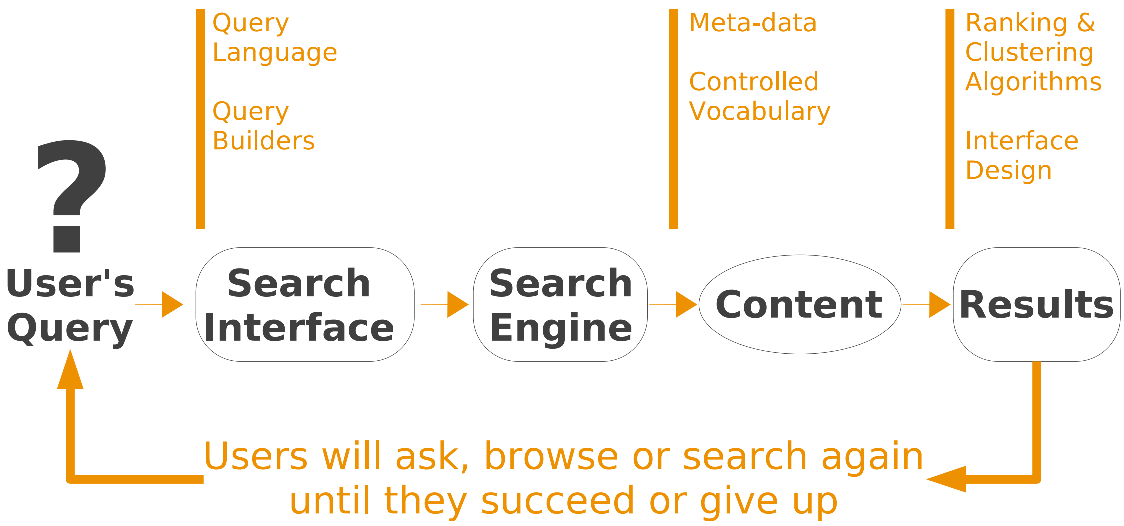


***Figure 1.1**: A user's typical search behavior.*

If we take a look at conventional information retrieval systems we will see the result set as a long list of ranked documents that users are forced to sift through to find relevant documents. The user usually is not informed about the ranking mechanism, making

[1]taken from http://www.taskz.com/definitions.php

it difficult to understand that sequence. Not understanding how a search engine works and how to properly use a query language or even simple boolean expressions add to the user's confusion. Even if a search engine produces good results, the user could end up being frustrated when the information presented does not give any hints about the documents. This can be solved if users receive some useful meta-data about the documents (e.g. language, date, size, ... depending on the kind of documents). The short and common definition for meta-data is 'data about data'; a more detailed definition for business data can be found in [Mar00]:

Definition 1.5 (Meta-data) *Meta-data is all physical data (contained in software and other media) and knowledge (contained in employees and various media) from inside and outside an organisation, including information about the physical data, technical and business processes, rules and constraints of the data, and structures of the data used by a corporation [Mar00].*

The lack of useful meta-data, transparency of ranking mechanisms combined with poor result visualizations adds up to the frustration in information-seeking: Instead of focusing on the search task, the user must invest time in understanding the interface and the retrieval system. There are basically two problems: how to specify a query and how to interpret the answer provided by the system. Surveys have shown that users have problems with the current paradigm of information-retrieval systems for Web search simply presenting a long list of results [CD00]. Such long lists of results are not a very intuitive method for finding the most relevant documents in the result set.

1.2 Problem Scenarios and Solution Approach

Before we take a closer look at the different aspects of substantiating the problem scenarios for a visual query tool and the approach taken in this thesis, the role of the user must be discussed. There is no 'ISO standard human', so we have to define the *'typical'* user for the scenario. Because the development took place in an EC-funded project[2], the definitions of that project were the foundation for the specification. In general, one can define the target user group as information experts, trained on the system and needing a powerful interface, so the focus is on features and ease-of-use. The objective of creating a visualization that is also usable by a novice or intermediate user was not dropped, but the priority now was clearly the professional user.

This objective can be seen in the case of the query formulation stage. Since an expert is usually familiar with the data repository and a query language like SQL, it seems unnecessary for him to use a visualization. This visualization is indeed

[2]INVISIP: The project INVISIP is part of the EU Information Society Technologies Programme and facilitates location site-planning processes and supports involved parties: www.invisip.de

more useful to a novice user than a bare-text input box. Nevertheless a visual representation can offer the expert user possibilities such as overview, distribution patterns, etc.

Another point of the query stage is the rapid changing of parameters to broaden or narrow a search result set. Whereas a command line interface usually offers little or no support for altering complex statements, a visualization can support this with the relevant widget, and thereby enhancing efficiency.

One would not normally suggest that the designer of a search engine supports human search behavior. Simplistic interfaces that range from a command-line to simple, form-based ones tend to ignore certain strengths and capabilities of the human visual and cognition system. The inclusion of recent research in the fields of human-computer interaction and cognition will increase effectiveness and satisfy the user. On the other hand it is essential that one does not rely purely on good-looking user interfaces, but also on useful interfaces that conform to certain usability standards.

Another group of problems arises in the field of data and its characteristics. Problems include such topics as quality of meta-data and the management of those meta-data from their creation to their storage and accessibility. Traditional database topics will not be addressed in this thesis: instead, the aspects concerning visual information-seeking systems are brought into focus.

Filtering search results in a visual information-seeking system (VISS) imposes more requirements. Typical VISS make use of several visualizations simultaneously. Even if only one visualization exists at a time, an additional filter should use techniques to synchronize with it, considering the filter as a modal application dialog.

1.3 Outline of this Thesis

The problems mentioned above will be addressed. A newly-developed visualization called 'CircleSegmentView' (CSV) will be presented. The purpose of this visualization is to serve as a dynamic query interface for an information-seeking system called VisMeB (Visual Meta-data Browser), which is engineered at the Human-Computer Interaction workgroup at the University of Konstanz (Germany). This visualization can also act as a filter tool for the result-set representation.

The development followed a usability engineering lifecycle. During my research I focused on topics about 'human search behavior' and 'Meta-data generation', 'analysis' and 'retrieval'. In short: After understanding human search behavior, as well as users and their tasks, one has to focus on the available data and how to generate appropriate meta-data to help users complement their picture of the information space. The visualization-based access to this information should follow human-computer

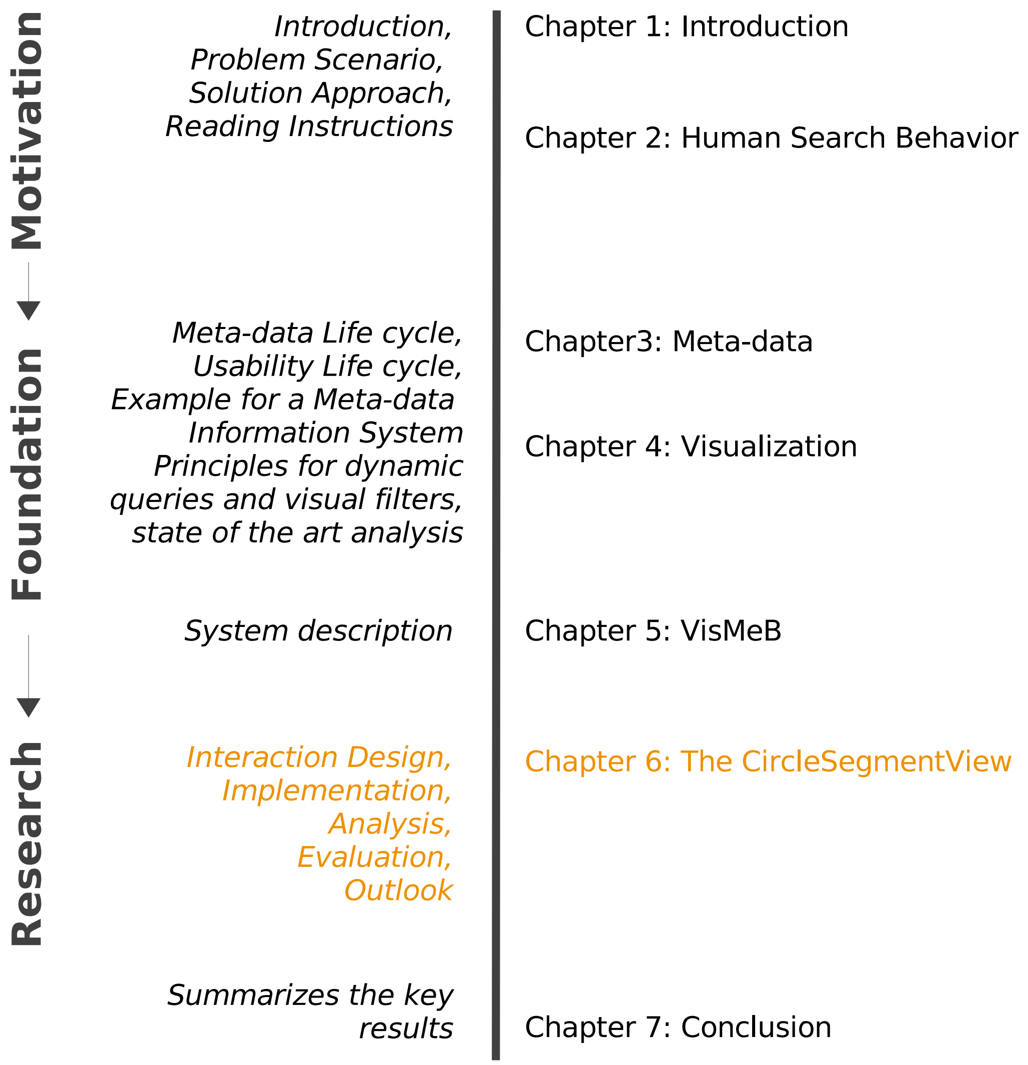


Figure 1.2: *Structure of this Thesis.*

interaction guidelines as well as established usability paradigms. These different aspects are reflected in the structure of this document (see Figure 1.2 on the preceding page).

A fundamental introduction, as well as an examination of human search behavior, describe the underlying motivation for this work (chapters one and two).

This is followed by basic principles and paradigms regarding the fields of human-computer interaction (chapter two (page 9) and chapter three, section 3.3 on page 38), meta-data (chapter three, page 31) and visualization (chapter four, page 59).

Chapter five (page 81) provides a short overview of the framework in which my work is embedded. The framework is called *VisMeB*, standing for visual meta-data browser.

Based upon these 'foundation chapters', the main part of my research is presented in chapter six (page 99). A detailed description of the implementation of the Circle-SegmentView (CSV) along with the design philosophy for the user interface and the interaction techniques is depicted. The results of the evaluations and analytic examinations followed by a review of the possibilities for further versions and improvements.

The last chapter gives a summary of my research (chapter seven, page 127).

Trademarks are respected but not explicitly marked within this work. A list of abbreviations used in this thesis can be found in table 1.1 on page 8. A look at the CSV from the web search engine Kartoo is shown in Figure 1.3 on the next page. According to this search engine my work is well-positioned around the key terms: visual, filter, interaction, suchterm (german for 'query term') and query preview.

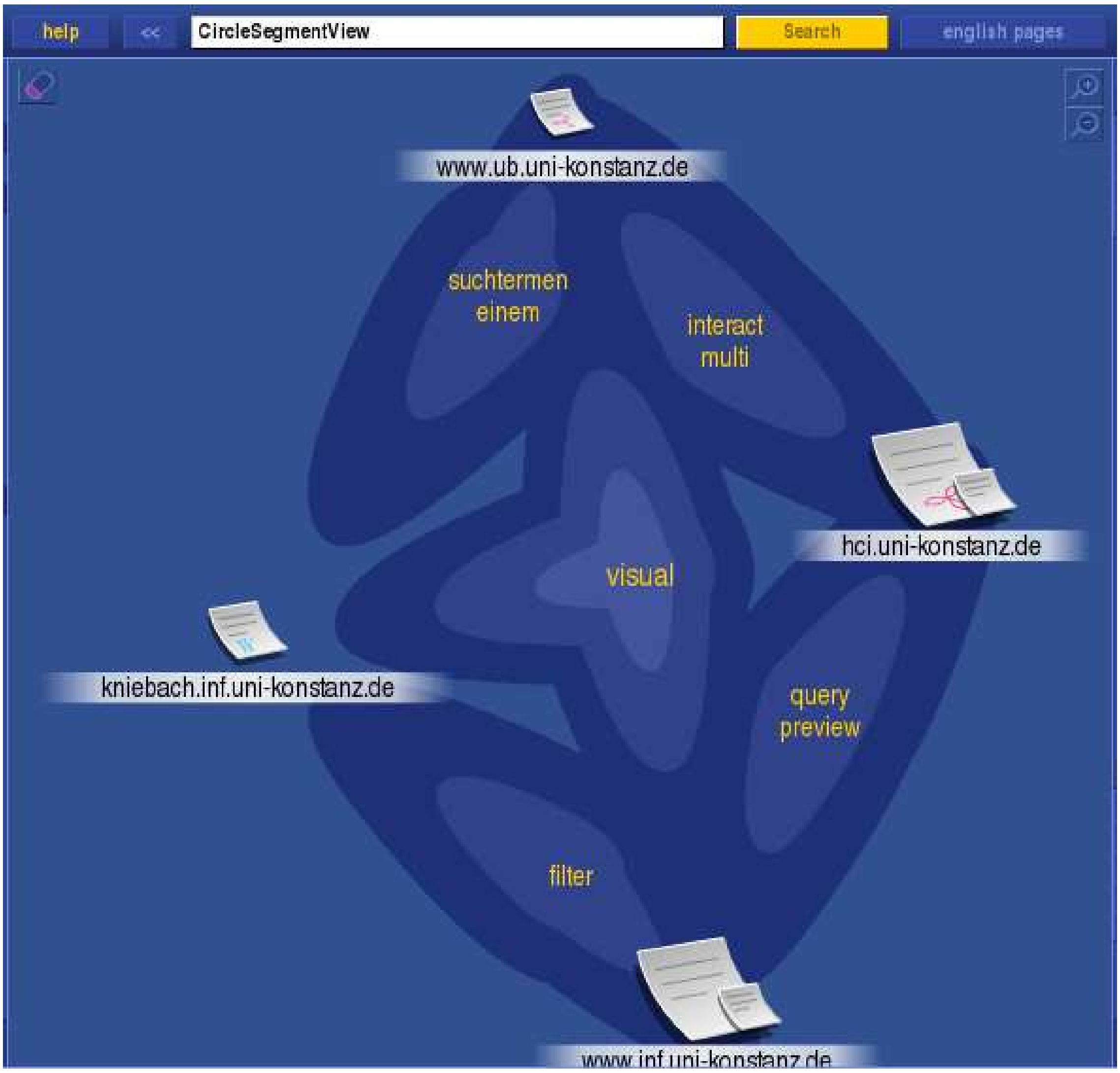


Figure 1.3: *Positioning of the CircleSegmentView according to the search engine Kartoo: key relations found: visual, filter, interaction, suchterm (query term) and query preview*

***Table 1.1**: Abbreviations used in this thesis.*

API	Application Interface
DTD	Document Type Definition
DW	Data Warehouse
EIS	Executive / Everybody's / Enterprise Information System
GUI	Graphical User Interface
HTML	Hypertext Markup Language
INSYDER	Internet Systme de Recherche
INVISIP	Information Visualization for Site Planning
IR	Information Retrieval
IV	Information Visualization
MIS	Meta-data Information System
RDF	Resource Description Framework
SOI	Sphere-of-Interest
SQL	Structured Query Language
URL	Uniform Resource Locator
VISS	Visual Information Seeking System
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

2 HUMAN SEARCH BEHAVIOR

Contents

Search engines, or rather the combination of search interfaces and the invisible blackbox functionality of the search engine, are a source of great frustrations for many users. The following assumptions about user interaction with 'search' are a useful guide to designing and evaluating the usability of an information seeking system.

Users may not understand the schemas or processes involved in searching. The report 'What's Wrong with Internet Searching' [PH96], a study involving novice Web users, noted that:

> Nearly all participants from both trials had difficulty formulating good searching keywords even when they had all the information they needed. In the real world, users go into a library or a shop and express their requirements in verbose or imprecise terms, or alternatively they browse through items on offer. They are not used to elaborating an artificial text string to match their requirements. [PH96]

The iterative process of searching often requires refinement of search queries and sifting of search results. Moreover, users may equate a negative response to their search query as indicating the non-existence of valid results.

When a user under-specifies a query, for example searching on Google for the term ’information’, the extremely large number of returned hits (known as the ’mega-hit’ problem) and the concomitant difficulty of wading through them may discourage users from searching further. With any large amount of information, if the first lots of information are not valuable, users may conclude that either none of the information is of value, or that the energy required to find something of value is too great [SJWS02]. Likewise, users may not understand that when they over-specify a query and then do not receive valuable results, they need to make their query more general (broaden the search).

Not all search interfaces are the same, and there are few standards on search interface construction[1], so there are no guarantees that a user will be able to successfully use any given search interface.

A further point is that users may not understand what they are searching against; if a search interface is querying against a subset of documents, the search form’s context and placement may suggest that it is searching against the superset of documents. Usually a user does not know much about the content of the underlying database.

If the search form offers more than one input box, users may not understand the different functions or roles of the different inputs. For example, an online music store offers a music album search that has a field that queries against title information, and another that queries against performer information, and these fields are part of the same form, which means a high cognitional load for the user. Users must differentiate between what they know about the title and what they know about the performer; users must predict what they should do if they only know information for one of the fields; users may not know which field is more valuable to the success of the search. At this stage one can easily see that meta-data equals knowledge as stated in definition 1.5 on page 3 and as such, it helps the user fulfill the task.

The complexity increases as more distant (for a definition of a ’distant’ function see 4.2.5 on page 70) types of information are requested on a search form, so a search form that has fields for title, author and (for example) price becomes significantly more difficult for the user.

Multiple search forms present similar problems, even when each form is optimized for a certain kind of query or for certain kinds of search parameters. For example, if a

[1]but there are common guidelines and DIN/ISO Standards for Human-Computer Interaction which should be used as a first starting point

site has a basic keyword search form that queries against multiple fields in the catalogue or document collection, as well as a form that queries only against product titles, the user may not understand that one form may be better – in terms of accuracy or performance – than the other form.

If a search engine provides the ability to specify the collection against which a search will be performed, some of the users are going to miss the distinction and will be confused when they get back results from a query against a 'wrong' or unexpected data set. Moreover, users may not be able to identify which collection might hold the answer to their query.

If the query form provides context-sensitive search, users must be aware of the context, and understand that their search is contextual, to avoid confusion. Likewise, if no contextual searching is offered, and there are distinct sections or topical areas of the document space, this should be made clear as well.

Users often cannot translate what they know about an information item into a successful query. Sometimes this is just a matter of phrasing, as when users enter natural language queries – 'I am looking for DVDs about Dinosaurs'.

At other times this may be a problem of clarity, as when the user cannot clearly enunciate what they know about the product or information they want – 'I don't remember the exact title, but it was a book about dogs and the author's first name was John.' With clarity problems, the quality of results cannot be predicted because of the uncertain validity of the search parameters.

This issue may be about conversion; for example the customer knows something about the product that does not translate into a parameter that can be included in the search query – if the customer remembers the color of the cover, that characteristic (read: meta-data!) is unusable by most bookstore search engines.

A more severe problem is when the user knows how to formulate a good query, but that query doesn't match the terminology or syntax required by the current search interface. In other words, users can frame and state their query correctly with regards to the subject domain, but not correctly with regards to the search interface/engine they are trying to use.

According to a report titled 'Mapping Entry Vocabulary to Unfamiliar Meta-data Vocabularies':

> The explosive increase in heterogeneity assures that the lack of familiarity required for efficient, effective searching is an increasing problem. When an index or categorization scheme is encountered, how is one to know what word or value has been assigned to the topic that one is interested in? Ex-

> pert human search assistance is often needed, but a sufficient population of human expert search intermediaries is unaffordable. The challenge, therefore, is to provide automatically the kind of expert prompting that an expert human search intermediary would provide. It has been argued that the most cost-effective single investment for improving effectiveness in the searching of repositories would be technology to assist the searcher in coping with unfamiliar meta-data vocabularies [Buc92].

Users may have trouble using differentiators to make the query specific enough to return the desired information. The report 'What's Wrong with Internet Searching' noticed many problems using search with their user sample:

> The central problem was that users did not seem to understand what were likely to be good quality differentiators. Thus one participant looking for Reebok trainers searched for 'sports shoes' rather than the more discriminatory 'Reebok'. Another entered 'competitive market share' when trying to find information about competitors for the Rover car company, not realising that he had to enter at least some contextual element [PH96].

Users type in a string of characters or words into a search form, then submit the query; users may not understand what then happens to what they typed in. The search system may perform logic on the search parameters to derive word stems or to wildcard the string, or any of a range of possible parameter tweaks designed to increase the effectiveness of the search or the number of query 'hits'. The problem with this logical processing of the search parameters is that the user may not understand why certain results are returned, and why other 'obvious' hits are not returned.

The subject domain for a commerce site includes the collection of products that comprise the site's catalog, product characteristics and usage assumptions unique to the market, terminology and definitions unique to the market, and schemes for organizing and categorizing the products unique to the market. The fact that customers may understand the products or how to use the products is no guarantee that they will understand how the market processes the products.

For example, an avid cyclist who has purchased many bikes and is an expert rider may still lack understanding about how the bicycle industry categorizes bicycles. Bikes are made for different purposes – road racing, road touring, off-road downhill racing, cyclocross, etc – and of different materials – various kinds and qualities of steel tubing, aluminum, magnesium, titanium, fiber, etc. – in different configurations – diamond frame, front suspension, full suspension, monocoque, etc. – in different countries – USA, Japan, Taiwan, France, Italy. Every one of these different attributes (meta-data) is important, but how much of the full spectrum of bicycle-related information will the typical user understand? If the user is searching against a bicycle catalog online, which of these data points is necessary or important for finding the bike they want?

Various studies [Jon94] [Fra88] [Nie97] show that some users approach search – especially a search function on a given Website – not as a means of information retrieval but as a mode of site navigation. This is an indication that the design of a search system must be part of a holistic approach to designing the information architecture of a system.

Users are likely to make simple input mistakes. Server search logs show high rates of typographical errors ('typos') in search terms. Typos are dangerous to user success in two ways: first, users may not understand that a typo was the cause of negative search results; second, users may incorrectly view a failure from typos as indicating that there are no results for the query, which may lead to the abandonment of the query.

Incorrect word or word version choice is also an issue with any strict, literal search engine: if the user enters the terms 'save private ryan' when seeking 'saving private ryan', a strict literal engine may fail because the wrong word-form has been used. Some engines can handle stemming and/or linguistic logic to expand the scope of the query, but literal engines will stop on this 'error'.

2.1 Users and the Task of Information Retrieval

Search *is* information retrieval, but neither term really helps when it comes to understanding what users of an information-seeking system are doing. Users want to find something. Online shoppers are more interested in locating that special something than they are in using a search mechanism per se; search is simply a tool, and as a tool should be appropriate to the needs of the user and to the general task of finding information.

Users bring different expectations and goals with them when they approach a search. One of the keys to designing – and evaluating the success of – an information seeking system is understanding what is going on in the user's head when they interact with search.

So, search is a tool for information retrieval, but the transaction of searching encompasses more than just looking something up. When designing and testing search mechanisms, one must consider the following issues:

- As a general tool for information retrieval, considering the domain, does the search mechanism function appropriately?
- Given a range of common information retrieval tasks appropriate to your site, does the search mechanism function appropriately?
- Is search the best way to accomplish common tasks?

- Does the search mechanism accommodate a range of user expectations regarding information retrieval on your site?
- Does the search mechanism accommodate users who do not know how to formulate a query, or who do not enough to uniquely identify what they are looking for?
- Does the search mechanism appropriately handle queries against a subset like a product catalogue?

2.1.1 Searching as a Mode of Navigation

Not all users approach search as a means of information retrieval; various studies [Jon94] [Fra88] [Nie97] show that some users employ search as a way to navigate within a web site. According to Jakob Nielsen,

> Our usability studies show that more than half of all users are search-dominant, about a fifth of the users are link-dominant, and the rest exhibit mixed behavior. The search-dominant users will usually go straight for the search button when they enter a website: they are not interested in looking around the site; they are task-focused and want to find specific information as fast as possible. [Nie97]

With most commerce sites centered around a product catalogue, this tendency to navigate via search makes it more difficult to design a search system that meets the needs of all shoppers.

2.1.2 Searching and Information-Retrieval Expectations

Lou Rosenfeld and Peter Morville devote several sections in their book 'Information Architecture for the World Wide Web' to a discussion of the varying needs that users demand from search. They mention four kinds of expectation (the following descriptions are liberally paraphrased from the book) [Ros02]:

- **Known-item searching.** The user's information needs are clearly defined and have a single, correct answer.
- **Existence searching.** User know what they want but do not know how to phrase the query, or whether the answer exists at all.
- **Exploratory searching.** The user knows how to phrase the query, but does not have a specific answer in mind; the user is essentially just looking around.
- **Comprehensive searching (research).** The user wants everything available on a given topic.

Any information provider must study their customers/users and their market and develop a list of common user tasks. Users must be able to find information/products, and they must be able to follow paths appropriate to their own preferences.

2.2 Behavioral Models of Information Seeking

2.2.1 Modes of Browsing and Searching

Marchionini [Mar95] reviewed the research on browsing and observed that ...

> ...there seems to be agreement on three general types of browsing that may be differentiated by the object of search (the information needed) and by the systematicity of tactics used. (p. 106)

Directed browsing occurs when browsing is systematic, focused, and directed by a specific object or target. Examples include scanning a list for a known item, and verifying information such as dates or other attributes. Semi-directed browsing occurs when browsing is predictive or generally purposeful: the target is less definite and browsing is less systematic. An example is entering a single, general term into a database and casually examining the retrieved records. Finally, undirected browsing occurs when there is no real goal and very little focus. Examples include flipping through a magazine and 'channel-surfing.'

In a similar vein, Wilson [Wil97b] identifies the following categories of information seeking and acquisition after a survey of research that included health-information seeking.

- **Passive attention:** such as listening to the radio or watching television programmes, where there may be no information-seeking intended, but where information acquisition may take place nevertheless;
- **Passive search:** which seems like a contradiction in terms, but signifies those occasions when one type of search (or other behavior) results in the acquisition of information that happens to be relevant to the individual;
- **Active search:** which is the type of search most commonly thought of in the information science literature, where an individual actively seeks out information, and;
- **Ongoing search:** where active searching has already established the basic framework of ideas, beliefs, values, or whatever, but where occasional continuing search is carried out to update or expand one's framework.

It is interesting to observe that in a separate stream of research in organization science, a comparable categorization of modes of organizational scanning or 'browsing' has been proposed, based on both empirical and theoretical research. The initial field work of Aguilar [Agu67] and the subsequent theoretical expansion by Weick and Daft [WD83], [DW84] suggest that users scan in four distinct modes: undirected viewing, conditioned viewing, informal search, and formal search. In this study, the authors amplify the information-seeking implications of each of these modes by elaborating on how directed the scanning would be, and on the amount and kind of effort expended (see table 2.1 on the facing page). The modes of viewing presented here are comparable and compatible with the three general types of browsing that Marchionini [Mar95]) identified. However, because 'browsing' in the next section is used to describe a pattern of micro-moves, the term 'viewing' stays here to avoid confusion and to indicate provenance.

In *'undirected viewing'*, the individual is exposed to information with no specific informational need in mind. The overall purpose is to scan broadly in order to detect signals of change early. Many and varied sources of information are used, and large amounts of information are screened. The granularity of information is coarse, but large chunks of information are quickly dropped from attention. The goal of broad scanning implies the use of a large number of different sources and different types of sources.

In *'conditioned viewing'*, the individual directs viewing to information about selected topics or to certain types of information. The overall purpose is to evaluate the significance of the information encountered in order to assess the general nature of the impact on the user. The individual has isolated a number of areas of potential concern from undirected viewing, and is now sensitized to assess the significance of developments in those areas.

During *'informal search'*, the individual actively looks for information to deepen the knowledge and understanding of a specific issue. It is informal in that it involves a relatively limited and unstructured effort. The overall purpose is to gather information to elaborate an issue so as to determine the need for action by the organization.

During *'formal search'*, the individual makes a deliberate or planned effort to obtain specific information or types of information about a particular issue. Search is formal because it is structured according to some pre-established procedure or methodology. The granularity of information is fine, as search is relatively focused to find detailed information. The overall purpose is to systematically retrieve information relevant to an issue in order to provide a basis for developing a decision or course of action. The four modes of scanning are summarized and compared in table 2.1 on the next page.

2.2.2 Ellis' Model of Information-Seeking Behaviors

Ellis [Ell89], Ellis et al. [ECH93], and Ellis and Haugan [EH97] propose and elaborate a general model of information seeking behaviors based on studies of the information seeking patterns of social scientists, research physicists and chemists, and engineers

***Table 2.1**: Modes of Scanning [Agu67] [WD83].*

Scanning Modes	Information Need	Information Use	Targeted Effort / Number of Sources	Tactics
Undirected Viewing	General areas of interest; specific need to be revealed	Serendipitous discovery 'Browsing'	Minimal / Many	Scan broadly a diversity of sources, taking advantage of what is easily accessible. 'Visioning'
Conditioned Viewing	Able to recognize topics of interest	Increase knowledge about topics of interest 'Learning'	Low / Few	Browse in pre-selected sources on pre-specified topics of interest. 'Discriminating'
Informal Search	Able to formulate queries	Increase knowledge of area within narrow boundaries 'Selecting'	Medium / Few	Search is focused on area or topic, but a good enough search (to solve a problem; move along) is satisfactory. 'Satisfying'
Formal Search	Able to specify targets	Formal use of information for decision-policy-making 'Retrieving'	High / Many	Systematic gathering of information about an entity following some method or procedure. 'Optimizing'

and research scientists in an industrial company. One version of the model describes six categories of information-seeking activities as generic: starting, chaining, browsing, differentiating, monitoring, and extracting.

'Starting' comprises those activities that form the initial search for information - identifying sources of interest that could serve as starting points of the search. Identified sources often include familiar sources that have been used before as well as less familiar sources that are expected to provide relevant information. While searching the initial

sources, these sources are likely to point to, suggest, or recommend additional sources or references.

Following up on these new leads from an initial source is the activity of *'Chaining'*. *'Chaining'* can be backward or forward. 'Backward chaining' takes place when pointers or references from an initial source are followed, and is a well established routine of information seeking among scientists and researchers. In the reverse direction, 'forward chaining' identifies and follows up on other sources that refer to an initial source or document. Although it can be an effective way of broadening a search, forward chaining is much less commonly used.

Having located sources and documents, *'Browsing'* is the activity of semi-directed search in areas of potential search. The individual often simplifies browsing by looking through tables of contents, lists of titles, subject headings, names of organizations or persons, abstracts and summaries, and so on. 'Browsing' takes place in many situations in which related information has been grouped together according to subject affinity, as when the user views displays at an exhibition, or scans books on a shelf. ('Browsing' in Ellis' model is different from 'viewing' in the previous section: here browsing describes looking for information at the micro-event level whereas viewing was described earlier as a broader context of looking at information.)

During *'Differentiating'*, the individual filters and selects from among the sources scanned by noticing differences between the nature and quality of the information offered. For example, social scientists were found to prioritize sources and types of sources according to three main criteria: by substantive topic; by approach or perspective; and by level, quality, or type of treatment [Ell89]. This again shows the importance of meta-data. The differentiation process is likely to depend on the individual's prior or initial experiences with the sources, word-of-mouth recommendations from personal contacts, or reviews in published sources. An interesting research prototype for arranging one's personal information repository based on individual knowledge is 'stuff I've seen' [DCC^{+}03]

'Monitoring' is the activity of keeping abreast of developments in an area by regularly following particular sources. The individual monitors by concentrating on a small number of what are perceived to be core sources. Core sources vary between professional groups, but usually include both key personal contacts and publications.

'Extracting' is the activity of systematically working through a particular source or sources in order to identify material of interest. As a form of retrospective searching, extracting may be achieved by directly consulting the source, or by indirectly looking through bibliographies, indexes, or online databases. Retrospective searching tends to be labor intensive, and is more likely when there is a need for comprehensive or historical

information on a topic.

Marchionini [Mar95] (pages 49-60) proposes another often-cited model of the information-seeking process, tuned perhaps to electronic environments. In his model, the information-seeking process is composed of eight subprocesses which develop in parallel:

1. recognizing and accepting an information problem,
2. defining and understanding the problem,
3. choosing a search system,
4. formulating a query,
5. executing search,
6. examining results,
7. extracting information, and
8. reflecting/iterateing/stopping

The subprocess of *'extract information'* bears the same name as Ellis' 'extracting' activity but the two processes are different. Marchionini [Mar95] describes extracting thus:

> 'There is an inextricable relationship between judging information to be relevant and extracting it for all or part of the problem's solution. [...] To extract information, an information seeker applies skills such as reading, scanning, listening, classifying, copying, and storing information. [...] As information is extracted, it is manipulated and integrated into the information seeker's knowledge of the domain' (pp. 57-58).

In Ellis' model, *'browsing'* and *'differentiating'* are activities separate from 'extracting,' which is 'systematically working through a particular source or sources to identify material of interest' ([Ell89]; p. 242). On the Web, we expect extracting (in Ellis' sense) to mean systematically working through a selected Web site or set of Web pages (typically using search engines) in order to search and retrieve material of interest.

Ellis [Ell89] thought that hypertext-based systems would have the capabilities to implement functions indicated by his behavioral model. If we visualize the World Wide Web as a hyperlinked information system distributed over numerous networks, most of the information-seeking behavior categories in Ellis' model are already being supported by capabilities available in common Web browser software. Thus, an individual could begin surfing the Web from one of a few favorite starting pages or sites (*'starting'*), follow hypertextual links to related information resources - in both backward and forward linking directions (*chaining*), scan the Web pages of the sources selected (*'browsing'*), bookmark useful sources for future reference and visits (*'differentiating'*), subscribe to e-mail based services that alert the user of new information or developments (*monitoring*), and search a particular source or site for all information on that site on a particular topic (*extracting*). Plausible extensions of the activities to Web information seeking (labelled Web Moves), are compared with the original formulations (Literature Search Moves) in table 2.2.

***Table 2.2**: Information-Seeking Behaviors and Web Moves [Ell89].*

Information-Seeking Behaviors	Literature Search Moves	Anticipated Web Moves
Starting	Identifying sources of interest	Identifying websites containing or pointing to information of interest
Chaining	Following up references found in given material	Following links on starting pages to other content related sites
Browsing	Scanning tables of contents or headings	Scanning top-level pages: lists, headings, site-maps
Differentiating	Assessing or restricting information according to their usefulness	Selecting useful pages by bookmarking, printing, copying and pasting, etc. Choosing / starting at differentiated pre-selected sites of known content
Extracting	Systematically working through a source to identify material of interest	Systematically searching a local site to extract information interest at that site

2.2.3 Towards a Behavioral Model of Information Seeking

Aguilar's [Agu67] modes of scanning and Ellis' seeking behaviors are combined and extended in a new behavioral model of information-seeking on the Web by Choo et. al [CDT99]. This is interesting because more and more information-seeking systems contain browsing modes. The table 2.3 on the next page below identifies the four main modes of information seeking on the Web as seen in section 2.2.2 on page 16. For each mode, the figure indicates which information-seeking activities or moves are likely to occur frequently, as suggested by theory.

In the *'undirected viewing'* mode on the Web, [CDT99] expect to see many instances of *'starting'* and *'chaining'*. *'Starting'* occurs when viewers begin their Web use on pre-selected default home pages, or when they visit a favorite page or site to begin

***Table 2.3**: Behavioral Modes and Moves in Information-Seeking on the Web.*

	Start	Chain	Browse	Differ	Monitor	Extract
Undirected Viewing	√	√				
Conditioned Viewing			√	√	√	
Informal Search				√	√	√
Formal Search					√	√

their viewing (such as news, newspaper, or magazine sites). *'Chaining'* occurs when viewers notice items of interest (often by chance), and then follow hypertext links to more information on those items. *'Forward chaining'* of the sort just described is the most typical during undirected viewing. *'Backward chaining'* is also possible, since search engines can be used to locate other Web pages that point to the site that the user is currently at.

In the *'conditioned viewing'* mode on the Web, [CDT99] expect browsing, differentiating, and monitoring to be common. Differentiating occurs as viewers select Web sites or pages that they expect to provide relevant information. Sites may be differentiated based on prior personal visits, or recommendations by others (such as word-of-mouth or published reviews). Differentiated sites are often bookmarked. When visiting differentiated sites, viewers browse the content by looking through tables of contents, site maps, or list of items and categories. Viewers may also monitor highly differentiated sites by returning regularly to browse, or by keeping abreast of new content (through, for example subscribing to newsletters that report new material on the site).

During *'informal search'* on the Web, [CDT99] expect differentiating, extracting, and monitoring to be typical. Again, informal search is likely to be attempted at a small number of Web sites that have been differentiated by the individual, based on the individual's knowledge about these sites' information relevance, quality, affiliation, dependability, and so on.

'Extracting' is relatively 'informal' in the sense that searching would be confined to looking for information within the selected site(s). Extracting is also likely to make use of the basic, 'simple' search features or commands of the local search engine, in order to get at the most important or most recent information, without attempting to be comprehensive. Monitoring becomes more proactive if the individual sets up push

channels or software agents that automatically find and deliver information based on keywords or subject headings.

During *'formal search'* on the Web, [CDT99] expect primarily extracting operations, with some complementary monitoring activity. Formal search makes use of search engines that cover the Web relatively comprehensively, and that provide a powerful set of search features that can focus retrieval. Because the individual wishes not to miss any important information, there is a willingness to spend more time in the search, to learn and use complex search features, and to evaluate the sources that are found in terms of quality or accuracy. Formal search may be two-staged: multi-site searching that identifies significant sources is then followed by within-site searching. Within-site searching may involve fairly intensive foraging. Extracting may be supported by monitoring activity, again through services such as Web site alerts, push channels/agents, and e-mail announcements, in order to keep up with late-breaking information.

2.2.4 Integrated Model of Seeking and Searching

An interesting and novel approach is deployed by Bates [Bat02]. She integrates the social and cultural with underlying biological and physical anthropological layers of human experience with respect to modern information seeking. Bates classifies the searching in four different modes (see Figure 2.1):

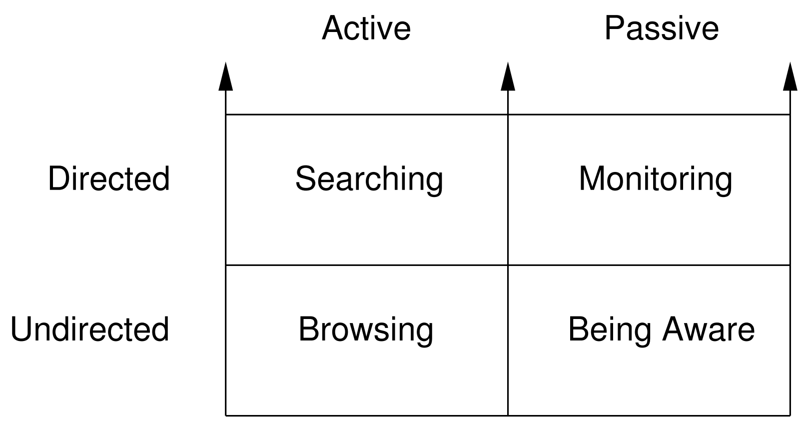


Figure 2.1*: Modes of Information Seeking by Bates [Bat85].*

> 'Directed' and 'Undirected' refer, respectively, to whether an individual seeks particular information that can be specified to some degree, or is more or less randomly exposing themselves to information. 'Active' and 'Passive' refer, respectively, to whether the individual does anything actively to acquire information, or is passively available to absorb information, but does not seek it out [Bat85].

In this context she defines *monitoring* as the state were humans maintain a 'back-of-the-mind' alertness for things of possible interest. The gap for the desired information is not big enough in order to invest more energy. *Browsing* is seen as opposite to *monitoring*. It is most often initiated by curiosity. *Browsing* can be seen as a complex smorgasbord of different actions like 'orientation', 'place-making', 'comparison', 'resolution of anomalies' and so on [Kwa92]. Bates integrated model states that *Directed search* takes only 1% of all search and seeking tasks (being aware: 80%, directed search: 1%, browsing and monitoring: 19%). She justifies this point with some examples and the principle of 'least effort':

> Countless studies have shown that people use the principle of least effort in their information seeking, even to the point that they will accept information they know to be of lower quality (less reliable), if it is more readily available or easier to use. A large number of these studies are reviewed in [Poo85].
>
> ...
>
> Directed searching is further complicated by another factor in our modern lives. It has only been in the last 200 years or so that the amount of recorded information available has grown to such an extent that complex and sophisticated access mechanisms have had to be developed to enable access.

With this more complete understanding of the information seeking behavior it should be possible to design visual information seeking systems that are more efficient and effective. The implementation of such a system should focus on offering information under various different aspects and count on the *'being aware'* aspect of Bates integrated model. Elaborate preview and overview tecniques could provide this kind of access methods (see section 4.1 on page 60 for more on these visualization concepts). An easy method to alter the information space should support the *'browsing'* attitude of the user.

2.3 Result set presentation

To be good, a search engine must respond to a query by returning a list of documents with the maximum quantity of relevant records and the minimum quantity of irrelevant records. Yet, there exists a separate set of problems that pertain to the user interface for viewing these search results. Typically, search results are presented in the form of a ranked list, broken down so that only 10 or 20 are viewable on a given Web page. Even if precision and recall are optimized, a list of search results will contain some documents that are not useful for the searcher and others that are useful. The list of search results is likely to contain subsets of documents that are similar, or that are related to the search query in a similar way. If precision and recall are not optimized (as is more commonly the case), then the list of search results will also contain irrelevant documents scattered among the relevant ones. Susan Dumais of Microsoft Research and Hao Chen of UC

Berkeley have conducted research into alternatives to the traditional ranked list display of search results [CD00]. They have found that users are able to find documents more efficiently when search results are organized into topical categories than when they are presented with a standard ranked list. Dumais and Chen tasked the study participants with finding documents via a traditional list interface as seen in Figure 2.2 on the next page, and then by means of category-style interfaces, one of which is shown in Figure 2.3 on page 26.

Dumais and Chen used four variations on category interfaces like the one shown in Figure 2.3 and three variations on list interfaces like the one shown in figure 2.2. In every case, users were more efficient at locating information through a category interface than through a list interface. The relative advantage of a category-based interface was even greater for 'difficult' searches as opposed to 'easy' ones (see Figure 2.4 on page 27).

Interestingly, as of April 2004, only one high-profile commercial search engine appears to have incorporated categorization of search results into its user interface. Teoma is a Web search engine that went live to the public in early 2002. Figure 2.5 shows the Teoma interface after it has completed a search on the term 'Knowledge Management.' At the bottom-left of the screen is the standard ranked list of search results. However, at the top right of the screen is a section labeled 'Refine Suggestions to narrow your search.' Although Teoma packages the links in this section as suggestions for query refinement, they function as subcategories within the domain of knowledge management.

During Teoma's beta release, the user interface even used the Windows Explorer 'folder' iconography to represent these links explicitly as categories and subcategories within the realm of 'Knowledge Management'. It is unclear why they switched metaphors, but the fact remains that clicking on links in the 'Refine' section of the Teoma interface will yield a subset of documents from the primary search as well as a new list of links (sub-subcategories) for further refinement. Regardless of what metaphor is used to represent the idea of categorization of search results, in the future it is likely that other search websites will follow and incorporate categorization of search results into the user interface.

Another notable system is *'grokker'* (http://www.grokker.com). It is an innovative desktop-based search application that performs dynamic topic maps and renders clusters of related information. It offers powerful filter and advanced search tools. The architecture is not yet part of a scientific publication, but the interface and its metaphors look very promising (see Figure 2.6 on page 28).

Using semantic-clustering visualizations is another approach to structuring search results. Usually, search engines give the user results 'flat-out', making it difficult to search

***Figure 2.2**: A ranked-list interface for search results.*

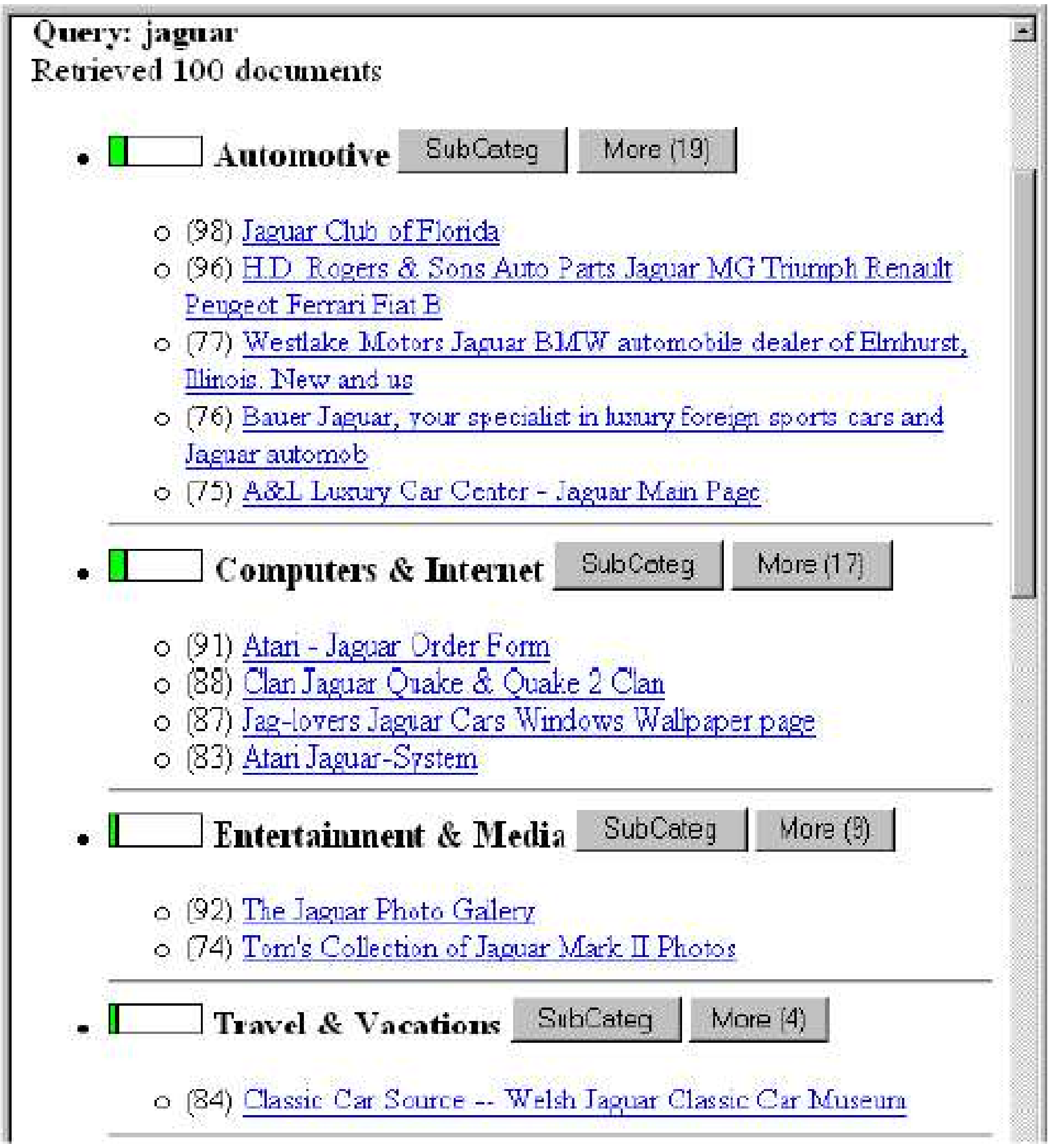


Figure 2.3: *A category-based interface for the same search results as shown in figure 2.2 on the preceding page.*

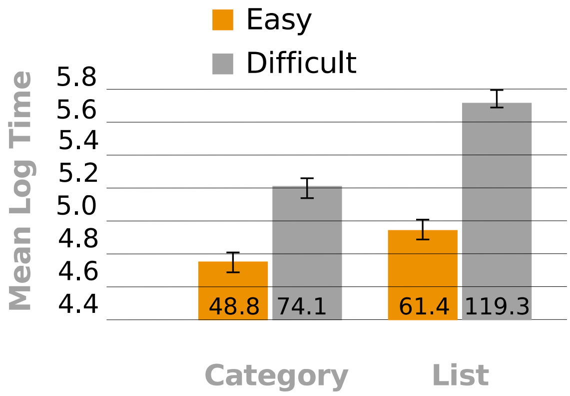


Figure 2.4: *Mean log time to complete tasks for easy and difficult queries for each interface type.*

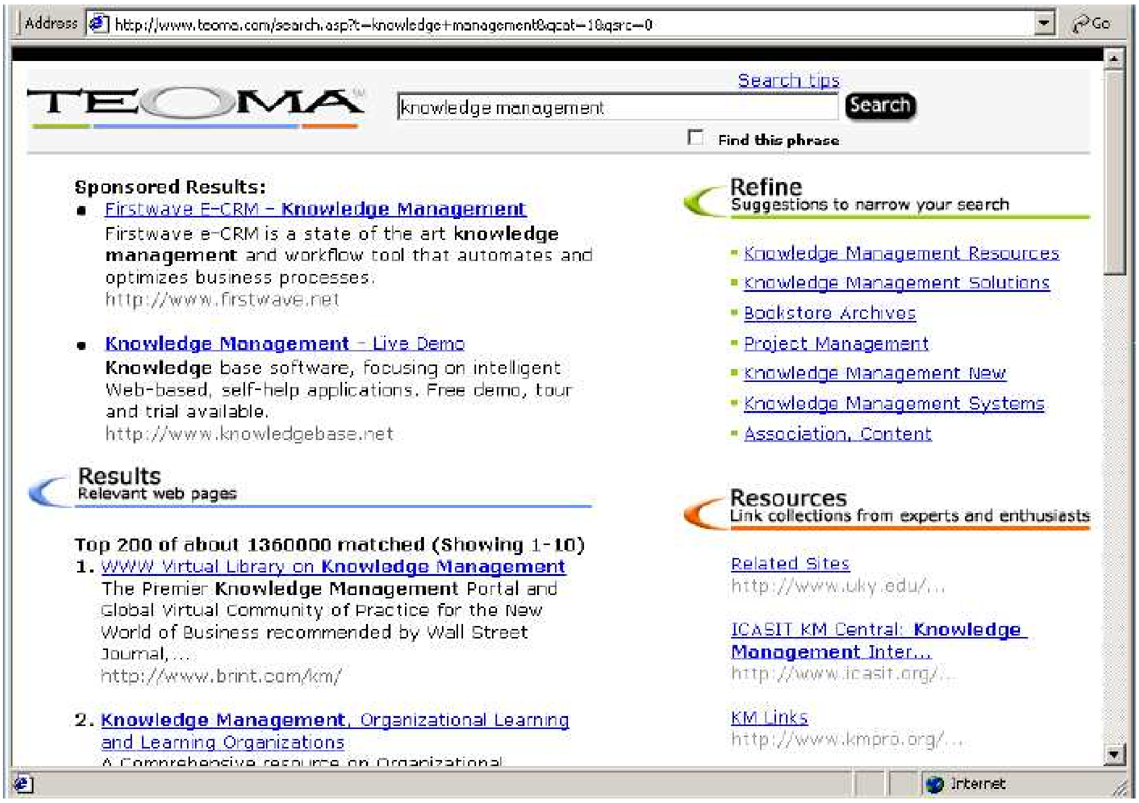


Figure 2.5: *Teoma.com user interface for search results.*

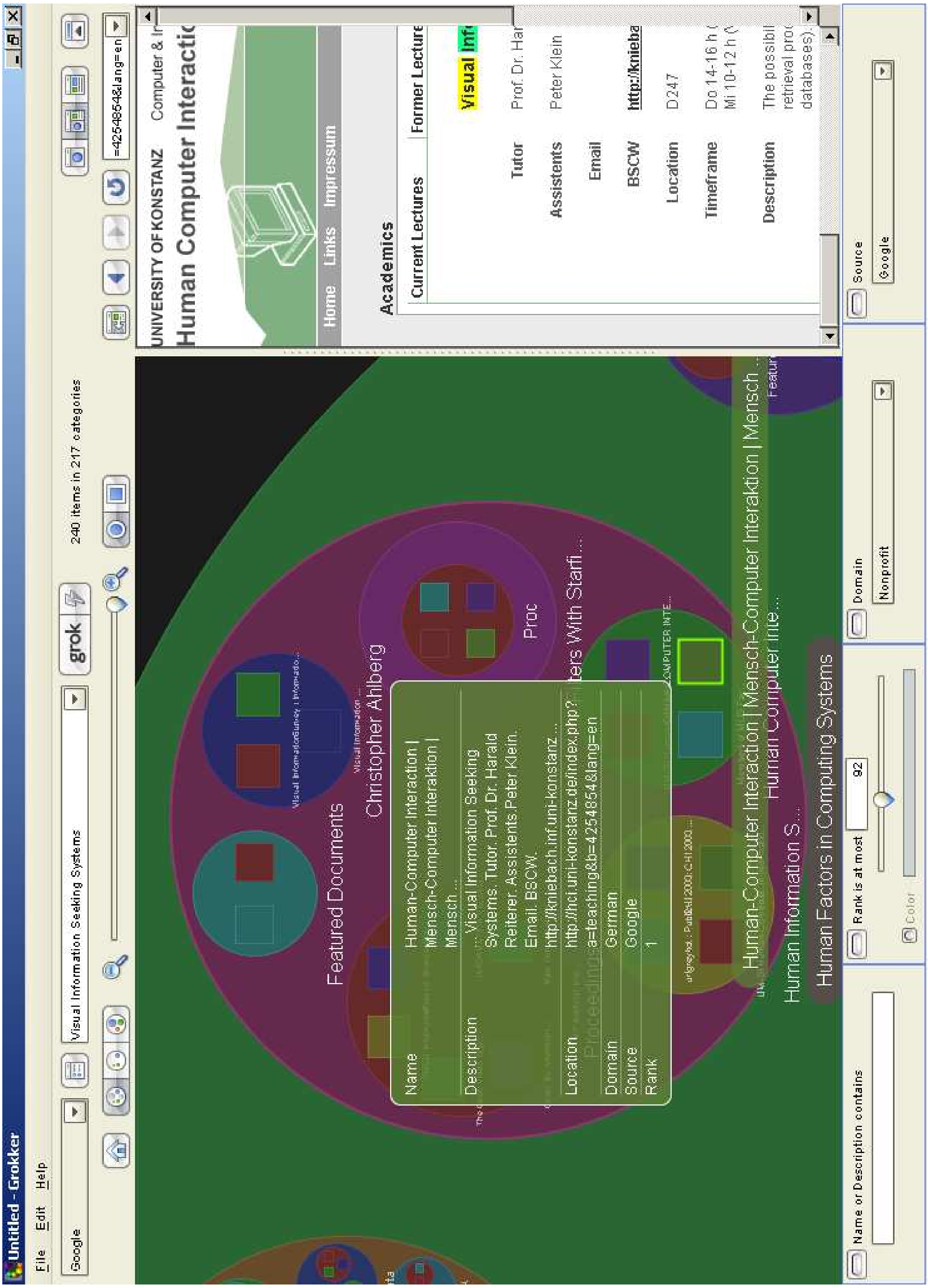


Figure 2.6: *Grokker's user interface for search results.*

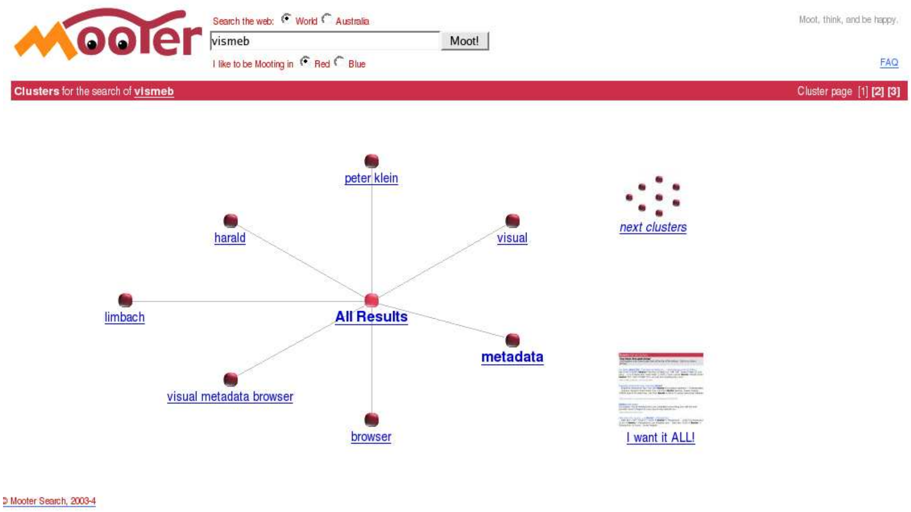


Figure 2.7: mooter.com user interface for search results.

through them. Mooter (http://www.mooter.com) likes to group the results so that the user can pick out the themes that they are interested in. The initial 'starburst' page gives the user the most likely clusters (see Figure 2.7). If nothing here inspires the user, there are further pages ('next clusters') to peruse. Once users see something that makes sense, they are free to select that cluster. Once the user is viewing results, the cluster list is down the left-hand side of the screen. The user can still go through the pages of clusters from the top of that column. When clusters are hiding near the URL of a result, that means that the result belongs to several clusters - there are several themes to the result. This shows that related results will have the same sort of themes. Additionally the original keywords are highlighted in the result set, along with the current cluster. Basic refinement is offered: pressing this button adds the selected cluster to the original query, and performs another search, giving you more specific results.

2.4 Summary

Understanding the user's way of searching and organizing information plays a key role in successfully developing an information seeking system. Besides the fact that the principle of 'know your enemy'[TzuBC] is a few centuries old, the design of today's information-seeking systems lacks a conversion of the known facts into working systems. One reason for this negative circumstance could be the unwillingness of different fields of research to share their results and merge them.

But it is not only the knowledge of human search behavior that opens the door for an effective and efficient information system. The *'other'* side of the retrieval scenario contains the data part. To turn pure data and facts into knowledge, information and wisdom, an in-depth examination of this aspect is unavoidable.

In this chapter, the necessity of examining the retrieval aspect from a user's point of view was established. The following chapter will deal with retrieval and data management and will complete the information-retrieval circle.

3 META-DATA

Contents

In principle, one can distinguish two approaches to satisfy a user's query: The first one is followed in the field of *'Information Retrieval'* (IR)[1] and consists of research into retrieval and ranking algorithms. Here the system acts as a black box where the different stages (like indexing, parsing, lexxing, ranking) covered by IR shield the user from this complex task and outputs a ranked list of documents. One can optimize these algorithms for specific tasks. An example for such an approach will be given in section 3.4 on page 43.

The other approach exposes its knowledge of the available data and offers users various possibilities to browse, examine and query the information space based upon their knowledge of the data. Here the user is more or less in control of the query parameters.

[1]see page 2 for a definition

Following the second approach, this chapter will cover an aspect of Information Retrieval known as meta-data[2]. Meta-data is information on the organization of the data, the various data domains and the relationship between them. Meta-data can be a valuable part in the information seeking process. It supports the user when his information need is not obvious. It helps by offering information about the data. For example, the 'Dublin Core Meta-data Element Set'[3] proposes 15 fields to describe a document. Following Marchionini [Mar95], this kind of information is referred to *'descriptive meta-data'*, meta-data that is external to the meaning of the document and pertains more to how it was created. Another type of meta-data characterizes the subject matter that can be found within the document's content. This type is referred to *'semantic meta-data'*. It can increasingly be found in closed environments like geo-information sytems (GIS), medicine databases or library catalogs where hierarchical taxonomies of terms (ontologies) describe vertain knowldege topics. A third kind of meta-data is represented by those attributes that specify the document repository under certain points of view like the amount of available documents with a specific feature. This data is usually computed on the database and allows (when offered to the user) filtering and preview of the power of the estimated result-set.

3.1 Meta-data Management

Meta-data does not just appear out of nowhere nor does it just fade away mystically. Meta-data is managed around the life of an *'asset'* [Ste03]. The value of meta-data is slowly degraded over time for various reasons such as quality, staleness, lack of use, etc. Stephens [Ste03] loosely defines an asset as any person, place or thing within the technological community. Examples of assets include databases, logical models, physical models, XML structures, components, documents, metrics, systems, interfaces, etc. Figure 3.1 on the next page provides a high-level view of the meta-data management life-cycle around an asset.

The asset itself can be described as a container of data, information, knowledge and/or wisdom that needs to be surgically removed. There is a process that will acquire the meta-data information from the asset. This process can be an automated extraction process or done by hand. Performing the data load by hand can be used in conjunction with an extraction utility and in most cases is required in order to fill in the information gaps.

A third option may be fairly obvious and that is to integrate a tool or collection of tools into the system development lifecycle. This would increase the efficiency of meta-data and push it into the most active role possible.

[2] see page 3 for a definition

[3] http://dublincore.org

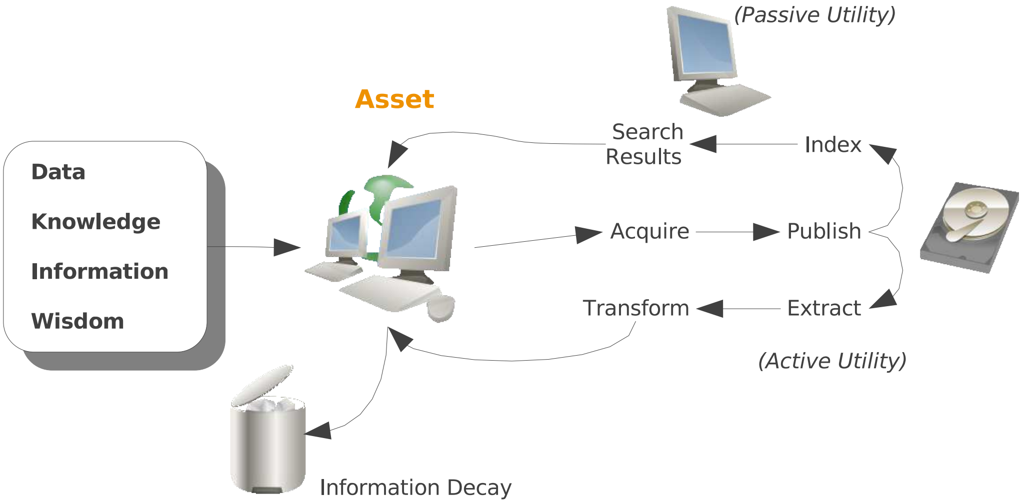


Figure 3.1: *Meta-data Management Life Cycle by [Ste03]*

However, in a large enterprise the environment is not as homogeneous as it could be. In addition, the odds are that the majority of the technology available is not built upon a current set of standards, which in turn makes the automatic enterprise integration nearly impossible, if not extremely expensive. While these processes are fairly well known and documented by various publications, the next series of steps is a source of much confusion and strife.

To know what an organization knows in principle and to make that knowledge available to the right people at the right time is the key to knowledge, and therefore meta-data, management [Kuh03]. The divergence of thought comes from the value generated from the *'passive'* and *'active utility'* built around the asset. *'Passive utility'* can be defined as the utility of publishing, indexing, searching and result-generation of meta-data information. It is widely recognized that an organizations most valuable knowledge, its essential intellectual capital, is not limited to information contained in official document repositories and databases scientific formula, research data, computer code, codified procedures, financial figures, customer records, and the like [YWB03]. However, in order to develop the know-how, ideas and insights of the community at large, meta-data must be managed at every stage of the asset. Since passive utility is the discovery and knowledge-based re-use of meta-data information, it stands to reason that passive utility must be delivered first. Active utility without information is simply pointless. Figure 3.1 shows the meta-data management lifecycle. Some examples of active utility for the management lifecycle include:

- Impact analysis across the asset population.
- Cross-reference and implied/derived meaning.

- Dynamic data exchange (XML).
- Real-time metrics.
- Web services and the utilization of meta-data-driven architectures.
- Dynamic reuse of asset information (i.e., screen/report field lookup).
- XML file validation using DTD and schemas.

In fact, active utilization may create a new asset in the form of new functionality. For example, providing the ability to cross-search an asset collection is analogous to bundling products that deliver new utility to the customer. Hence, the return arrow back to the asset inventory for the active utility process (see Figure 3.1 on the page before). Therefore, not only can a meta-data services group catalog technical assets, they can also create them.

The final area of Figure 3.1 on the preceding page is the information decay arrow. What this means is that information that stays within the repository will decay, indicating that the accuracy of the data is only 100 percent valid for a period of time. The most obvious reason for this is that the technological community is constantly changing. Even the low-level data constructs are changing. Suppose we take a snapshot of the logical, physical and operating system view of a database. How long will this snapshot be accurate? Perhaps a better question is: how long before the next DBA modifies the data structure or the modeler updates the text on a field? The longer information sits in a repository the greater chance that this information is not only inaccurate but could lead to erroneous decisions from the end-user perspective. A content-aging strategy should be a part of every meta-data implementation [Mar00]. Content aging simply provides the administrator with details of which information has not been updated in the past 30 days or whatever time period is appropriate to the business. Contacts can then be made to determine if the information is to be removed or updated.

What is the rate of decay for information? Considering information collected on a person provides: address, credit score, medical history, etc. A human being is constantly changing and therefore the information about the person is constantly changing. While it is not sure what the rate of information decay is, it can be slowed down by increasing the usage of meta-data information in both passive and active frameworks. This is essential to ensure the quality of the meta-data.

3.2 A Meta-data Lifecycle approach

Meta-data is an emerging approach to organizing digital information in a structured manner and supporting precise retrieval for digital libraries on an extraordinary Internet scale. Although there are many meta-data practices in digital libraries, very little literature has been recorded on the subject of how to choose the right meta-data formats for their own projects [CCCL01].

In an era of digital libraries, meta-data is often used to organize information in an ordered way to support better discovery and retrieval of resources. It is very important to understand the content of documents prior to applying any specific meta-data formats or standards for the digital libraries, so content analysis is an essential requirement for any digital library project. According to Stanton's concept, content analysis can be divided into five stages as follows [Sta95]:

1. determine objectives,
2. define unit of analysis,
3. construct categories for analysis,
4. test coding to assess reliability, and
5. conduct analysis

Stanton's conceptual ideas are mainly focused on document analysis for designing a hypermedia system, so it is a type of computer-system approach to analyzing documents. On the other hand, Hudgins, Agnew, & Brown planned a workflow for a meta-data project based on project management perspective. This approach demonstrates ten tasks in managing a meta-data project including understanding the entire project, documentation, maximizing existing infrastructure, choosing and evaluating the appropriate meta-data standard, meta-data record design, preliminary testing of workflow, initial staff design, workflow testing at midpoint, workflow testing at project conclusion, reporting results, and conclusion [HAB] (pp. 42-53).

One of the best documented meta-data projects took place in Taiwan [CCCL01]. Over 20 projects were in need of a meta-data plan and an implementation of the plan in the Digital Archive Initiative, which has been supported by the National Science Council in Taiwan since 2000. In order to achieve a consistent structure for these projects, a meta-data lifecycle was designed by the Sinica Meta-data Architecture and Research Taskforce (SMART) for this requirement in terms of both project management and content analysis. The meta-data lifecycle is composed of nine (Evaluation and Worksheet completion are left aside) components (see Figure 3.2 on the next page) and can be restarted when new, or changed, project requirements for meta-data are initiated. These tasks for the lifecycle are conducted by a series of questionnaires and tables. However, these components for the meta-data lifecycle are as follows: interview with content experts, analysis of project requirements and attributes, review of relevant meta-data standards and projects, analysis of information requirements, preparation of the meta-data requirement specification, evaluation of meta-data system and development, preparation

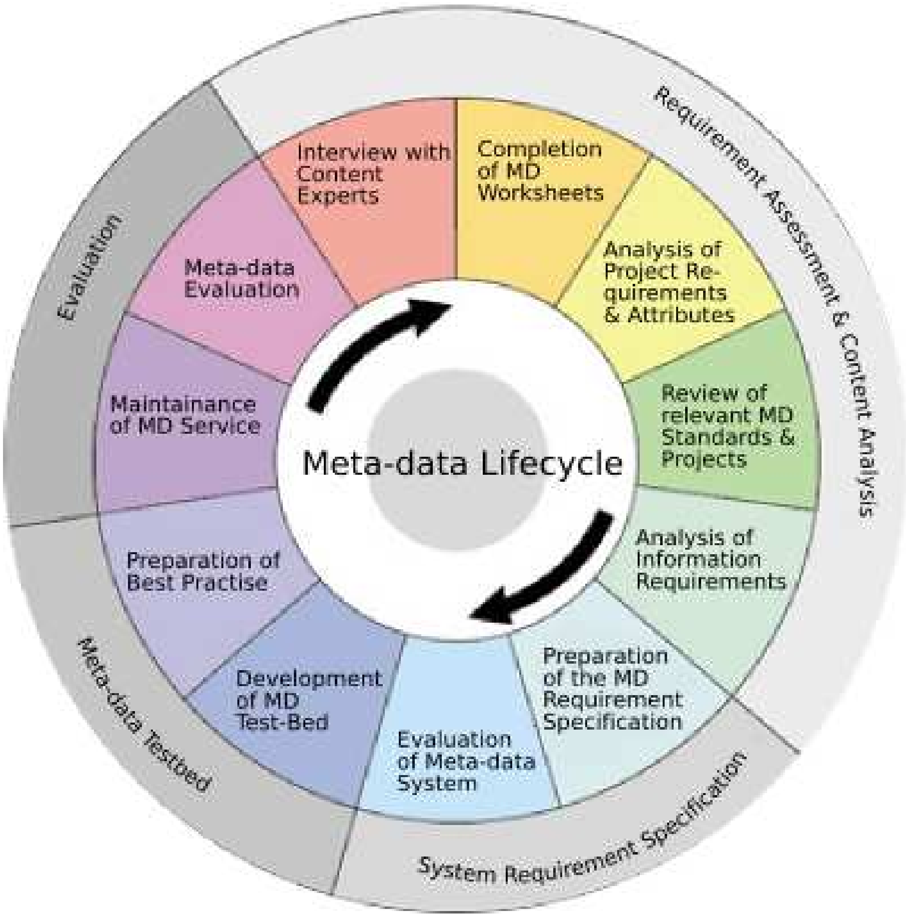


Figure 3.2: *Meta-data Lifecycle by [CCCL01]*

of best practice, development of a meta-data test-bed, and maintenance of meta-data service.

The first step of the meta-data lifecycle is to have a face-to-face interview with content experts and to get an overview of their meta-data requirements for each content project. Prior to the interview session, it is necessary to undertake two tasks. The first step is to make a serious examination of project background information based on review of project proposals such as purposes, goals and expected results. Secondly, questionnaires are send to content projects and requests information about scope, meta-data element and structure, legacy record and system, meta-data context, expected results for different stages, contact information, and so on. During the interview session, several points need to be clarified as follows [CCCL01] :

- **Contact information:** who belongs in the contact window? Contact information on the project participants.
- **Meta-data schedule:** when are the meta-data expected to be accomplished?
- **Meta-data scope:** how many types of meta-data are required for the project? Examples are types of object, person, event, control and expression of timelines, and geographic name.
- **Legacy record and system:** basic information about the learning system, including meta-data elements, structure, and number of records, storage format, input method and system. In addition, it is useful to understand the advantages and disadvantages of the legacy system.
- **Meta-data context:** is only one meta-data database constructed for this project? Are any other databases required to integrate with this meta-data database, such as geographic information system (GIS)?
- **Meta-data role and function:** What kind of meta-data role is proposed for each project? What kind of function should be achieved by meta-data? Examples are resources descriptions, discovery, annotation, or content analysis.

This process is reminiscent of the 'Usability Engineering Lifecycle' [May99] and will be discussed in section 3.3 on the following page. The task of comprehensive analysis is necessary to ensure the requirements and attributes of meta-data. The project requirements of meta-data would be verified in a systematic way after a detailed discussion in interview sessions. The most important meta-data task is to select an appropriate meta-data standard, instead of developing a new standard. In preparation for analysis of information requirements, several tasks should be undertaken as a basis for analyzing and ascertaining the project requirements [CCCL01].

1. The definitions and examples of initial meta-data elements should be offered, and would be clarified after interview and adjustment.
2. The proposed meta-data standard and elements list are selected and explained to project participants.
3. A comparison of selected standards and required elements of the project is conducted.
4. An analytical context diagram with various relationships for meta-data scope and context is defined.
5. Indexing keys and access points would be recommended as a basis for system design, as well as for meta-data role and function. In this session, the benefits are as follows:

- Meta-data elements and categories are chosen and defined, clearly based on a comparison with an existing meta-data standard.
- Distributions of meta-data elements are verified and compared with selected meta-data standards. These include the distribution of description, administration, system management, rights management resources discovery.
- Meta-data scope and context are clarified, and dependent relationships are also clearly drawn and attributed to a diversity of categories.
- It could establish what kind of systems and databases need the integration of a meta-data mechanism such as GIS.
- A quick comparison has been made between existing meta-data standards and the meta-data elements required for the project.
- Real examples and definitions for projects are collected as a basis for the best practise.

3.3 Synergy Effects between Usability and Meta-data Engineering

> We cannot discover how users can best work with systems until the systems are built, yet we should build systems based on knowledge of users and how they work. This is a user-centred design paradox (Marchionini, 1995, p 75). [Mar95]

The term *usability* is usually defined as *effectiveness*, *efficiency* and *satisfaction* with which a specified set of users can achieve a specified set of tasks in a particular environment, a definition[4] taken from ISO 9241 by the International Standards Organization [ISO96]. Based on primarily cognitive and perceptual constraints, it distinguishes between *ease-of-learning* and *ease-of-use* as two of its broad dimensions. Despite its definition, the question of how to define criteria that are able to measure the quality of a user interface raises interesting issues.

The first thing to keep in mind is that the usability of an application depends strongly on the user and the given tasks. We distinguish between the user with his knowledge, skills, experience, education and practice on the one hand, and the tasks with their complexity, difficulty, duration on the other, but between these two, for evaluation of the usability in a certain situation, it also necessary to analyze the social and physical environment of the software application, its hard- and software power and the place of work (e.g. screen size, size of the work space).

[4] http://www.usabilitysa.co.za/usability.htm

A second step is the evaluation of effectiveness, efficiency and satisfaction and how they can be measured.

Definition 3.1 (Effectiveness:) *represents the precision and completeness of how a user can achieve a task. This can be measured by recording the results of a work and their evaluation.*

Definition 3.2 (Efficiency:) *describes the amount of work involved in solving a task in relation to precision and completeness, which can be measured by the amount of time, the amount of keystrokes or the amount of interactive steps which a user must perform to complete a task.*

Definition 3.3 (Satisfaction:) *describes the comfort to a user in using a software application, which can be measured by questioning the user.*

These three criteria are part of the requirement engineering, just as usability engineering is part of requirement engineering, too. Usability is the ability to reduce operational, functional or organisational problems:

1. **Look and feel** - usability design enhances the interaction with the application, and reduces inconsistencies, badly-chosen colors, typography or bad placement (static elements) and unsuitable interaction elements like pull-down menus or dialog boxes, complex navigation (dynamic elements).

2. **Conceptual model** - usability design also focuses on the functionality of an application in solving a specific problem.

3. **Work re-engineering** - analyzes the organisational problems and the problems that arise from the usage of software (e.g. monotonous tasks, flexibility of work). Considering the e-commerce application mentioned previously (ticket reservation system), work re-engineering tries to analyse the way customers place their orders.

In conclusion, specific attributes of how a user interface has to be designed and how it interacts with the user can be noted. A dialog with a user should contain the following aspects. The interface should :

1. be appropriate to solving the user's task;

2. be able to describe itself;

3. have the capabilities the user expects it to have;

4. be fault tolerant;

5. offer the possibility of individual user-settings;

6. should help the user to learn.

Regarding two new aspects that have not been implemented so far, a user interface should also:

1. help users to communicate and cooperate (asynchronous communication), and
2. contain data security

A more detailed way to implement usability was designed by D. Mayhew [May99] in 1999 with the Usability Engineering Lifecycle Model as printed in Figure 3.3 on the next page. Traditional software engineering principles can also be used, as they only require small modifications. But it has to be taken into consideration that these lifecycles not only form a reference, but also focus on design processes, best practices, work products and capability levels like CMM (capability maturity model that describes and measures how to implement quality of use), UMM or DATech2[5], so instead of concentrating on the product it is necessary to concentrate on the development process. If many software products are being developed, this is especially important in order to guarantee that the quality standard on all products remains the same.

The task of building a Visual Information Seeking System is far from being an easy exercise. The main consideration is focused on two areas: Firstly, the user and his/her task; secondly, the available data. The first steps in this phase of the project can be execute in parallel. Whereas the 'User Profile', 'Contextual Task Analysis' and other considerations (like 'Platform capability' ...) result in a definition of 'Usability / Feasibility Goals' and specific 'Design Principles', the accompanying analysis of the available raw data results in a 'Meta-data Specification' (see Figure 3.4 on page 42). This specification also receives input from the 'user-side' so that it is possible to integrate the human factors in the meta-data specification. Such valuable information could contain answers to questions like:

- What kind of attributes are users interested in?
- Which data format do they prefer (or is needed)?
- What information is expected at which level?

Based on these formal specifications the next step includes the building of a so called 'Meta-data Toolbox' which includes utilities for data entry, data evaluation and maintainance as well as a data 'Aging / Quality Strategy'. These tools will define the backbone of the data repository.

The user side is emphatically characterized by the 'mockup-evaluation-cycle'. Firstly *low fidelity* (lofi) prototypes should determine if the principle design is usable and if an effective UI is feasible. Secondly the *high fidelity* (hifi) prototypes are used to check

[5] german counterpart of UMM

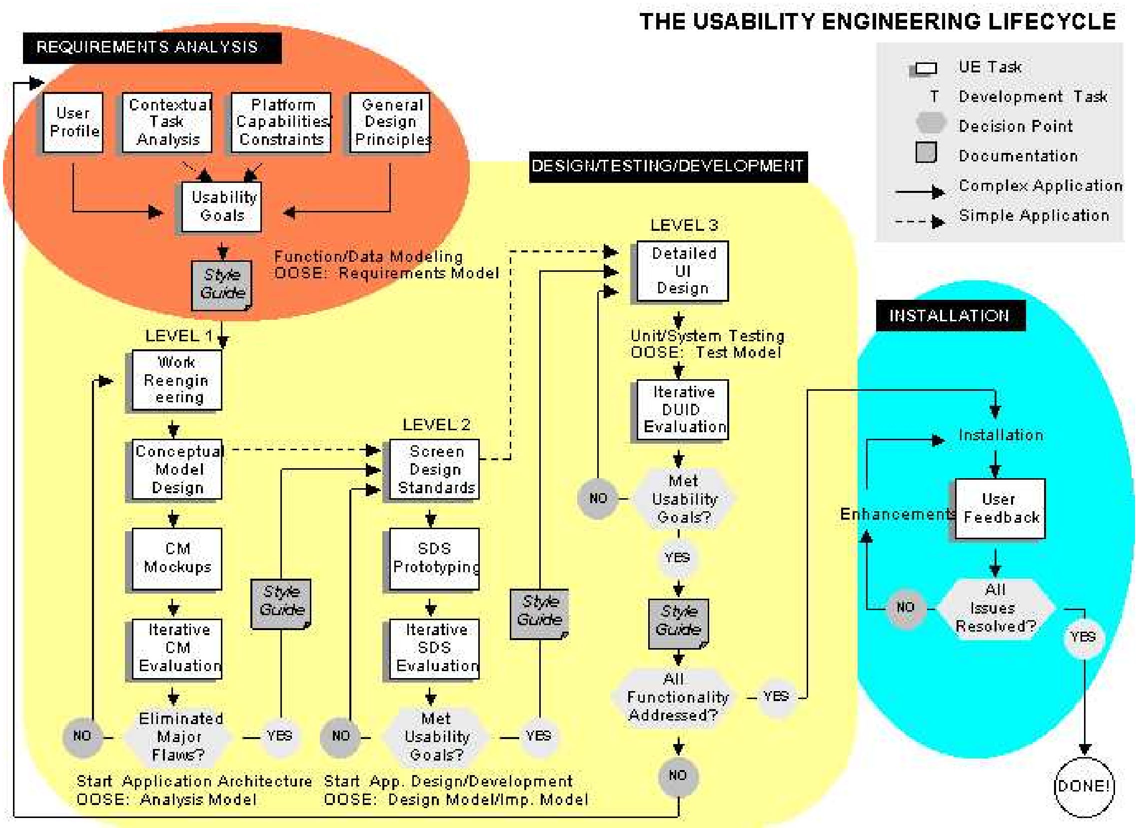


***Figure 3.3**: A schematic representing The Usability Engineering Lifecycle. It indicates all lifecycle tasks, and approximately where each one should be applied within either a modern rapid prototyping - or an Object Oriented - software engineering methodology. It shows how Usability Engineering Lifecycle tasks occur in parallel with traditional development tasks and are tightly intertwined with them [May99].*

the interaction design and to eliminate the last usability flaws. The results of these evaluations should no longer have an impact on the 'Meta-data Specification'. But if this eventuality arises, it should be compensated for the availability of the 'Meta-data Toolbox'.

The last steps of this strategy contain the fine-tuning of the database or the underlying information repository (like: building hashes, indices, views, ...) as well as usability studies with a first version of the user interface.

During the whole process the work on the visualizations must be closely coordinated between both sides. The *visual information seeking mantra* as stated by Shneiderman [Shn04] (Overview first, zoom and filter, then details on demand) is the foundation for the collaboration with the user side. Visualizations must be chosen; interaction design

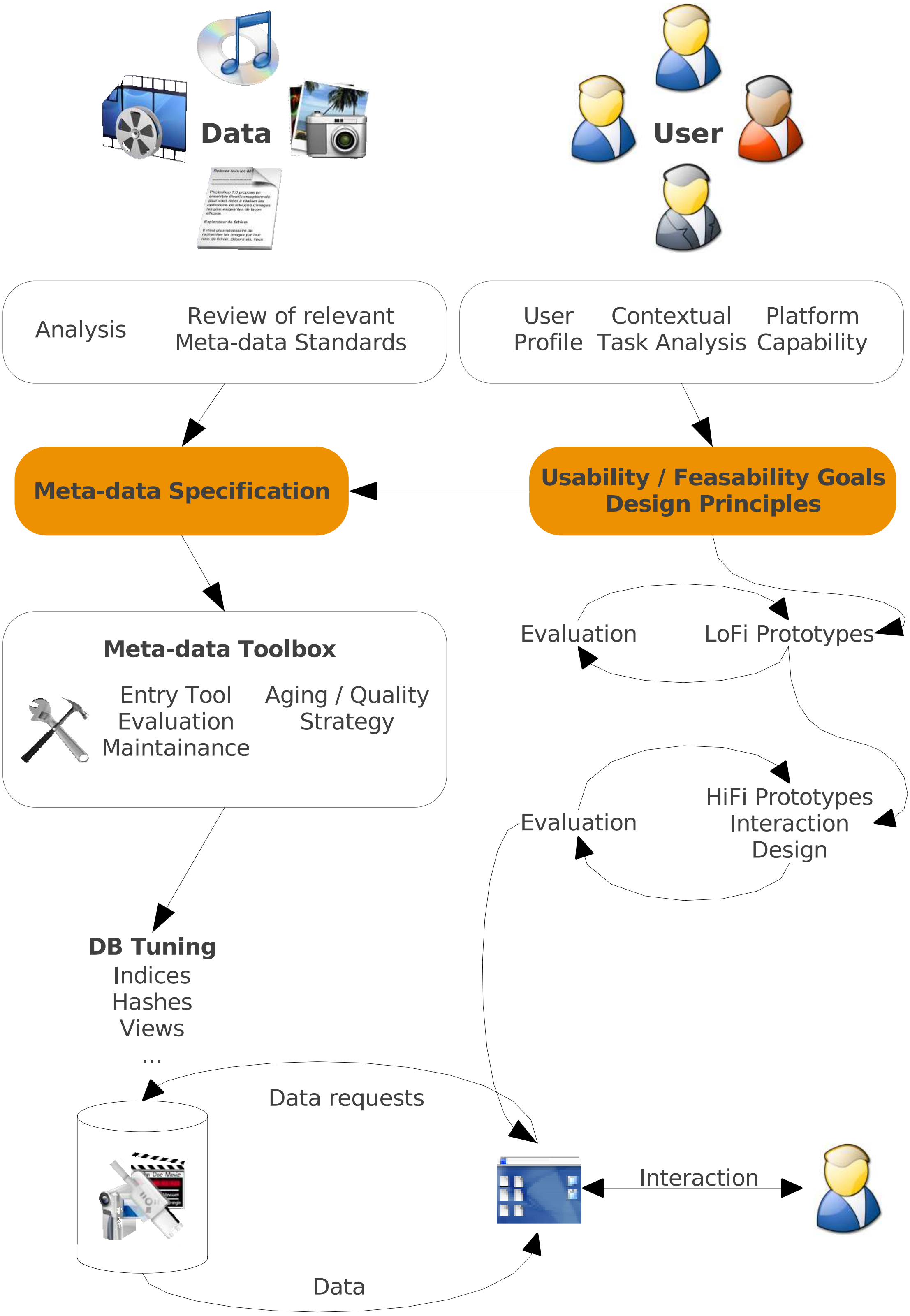


Figure 3.4: Data and user-specific tasks for a successful implementation of a Visual Information Seeking System.

must be specified to feed the testing-evaluation circuit. Results of the evalutions represent input for redesign ideas.

Visualizations rest upon the data. Questions about the data types, data formats and the availablity must be laid down in the 'Meta-data Specification'.

3.4 Example: Meta-data Information Systems for Geo-Spatial Data

This section covers the research activities for a Meta-data Information System (MIS) in the GIS context and should serve as an example of how a complex meta-data standard can be managed with optimized meta-data ranking schema. The abstraction and generalization of the research results can be used as a foundation for any meta-data based information system.

This research was part of the INVISIP Project (Information Visualization in Site Planning, EU-funded project IST-2000-29640, see http://www.invisip.de/).

Complex application examples or scenarios such as site planning need a lot of information, especially spatially-referenced data in order to find the best place for a new building or an industrial area and to solve individual planning tasks such as generating ecological, environmental or traffic reports. With reference to this information demand, in the last decade a lot of scientific and organizational effort has been put into structuring the geodata market and to establishing information systems and global infrastructures, thus enabling data suppliers to describe (→ eCommerce, meta-data) and users to find appropriate data (→ information retrieval, data mining). The most popular features of such information systems are meta-data information systems or catalog systems on regional, national or international bases, on-line-shops or web-portals to geospatial data archives.

Within the R&D project 'Development of the InGeoForum Information Center', the Fraunhofer Institute for Computer Graphics (summarized and described in [Göb02]) has undertaken a comprehensive study concerning meta-data based infrastructures and MIS for Geo-data, application areas, information spectra covered, standards, formats, protocols and relationships between different information systems and involved institutions (see Figure 3.5 on the next page). Göbel and Jasnoch [GJ01b] give an overview of this study and point out technical strengths and weaknesses of existing approaches. Concerning meta-data-based search and visualization techniques within the different information retrieval phases [JGB00], the most important results are:

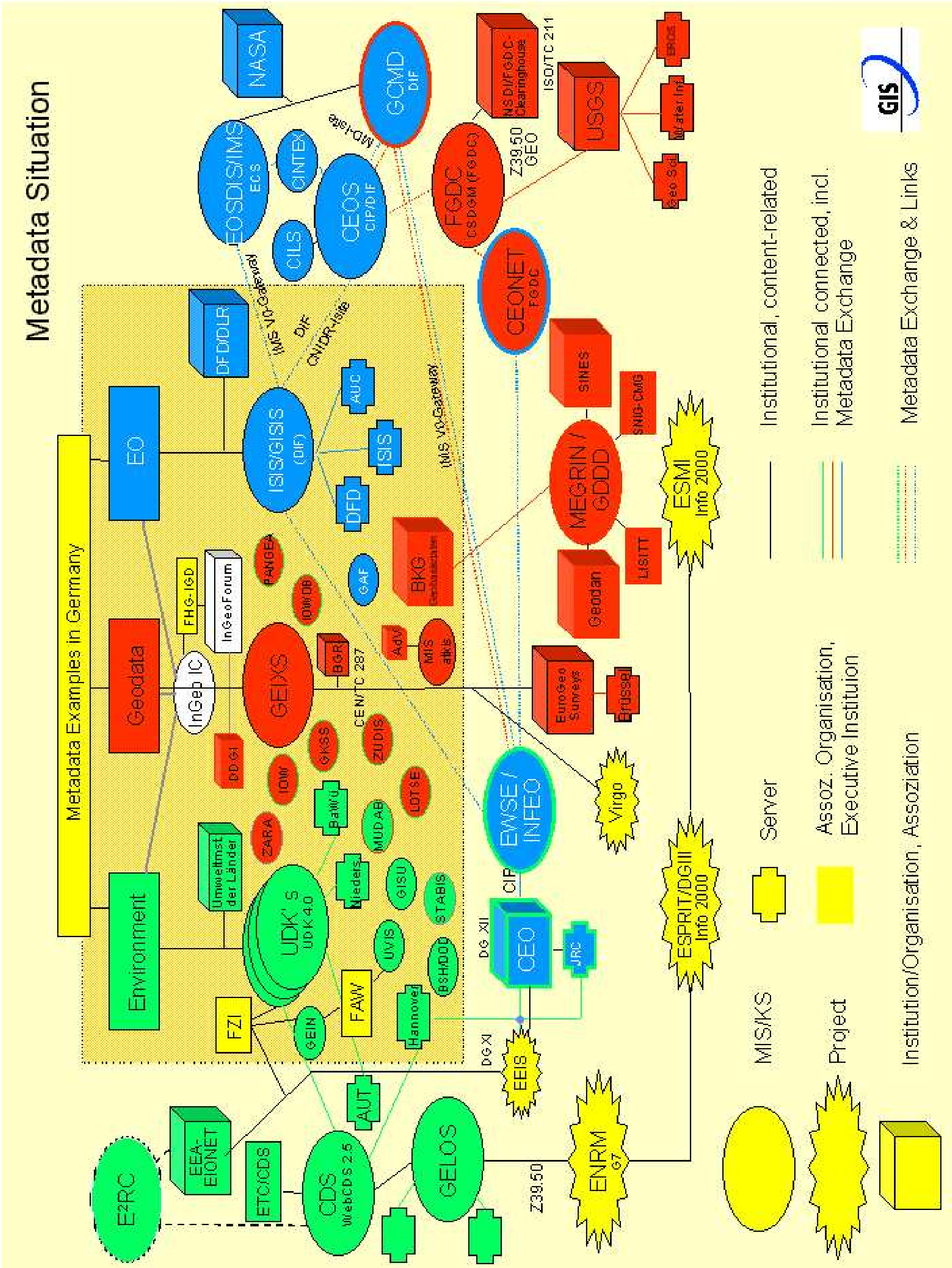


Figure 3.5: *Overview of existing MIS for Geodata [Göb02].*

- All analyzed approaches offer some kind of keyword search and geographic search. Additionally, some systems offer temporal search mechanisms. Thematic access is realized by categories and domain values corresponding to meta-data formats and well-defined terminology used in the special Geo-data application area.

- GUI components consist of text fields, attribute/keyword lists and sometimes maps. The maps are implemented as click-able image maps or web-based GIS (geographic information systems) components to specify geographic entities as spatial search parameters. Figure 3.6 is an example showing graphical user interface elements of GEIXS (Geo-Scientific Electronic Information Exchange System) and GDDD (Geographical Data Description Directory)[6].

- Search results are presented as textual result lists. There are first approaches which show the relevance score of results, but in general it is scarcely possible to compare and interpret result sets, because there is no visual feedback referring to search parameters and its effects on search results. Besides, this also negatively influences query modification.

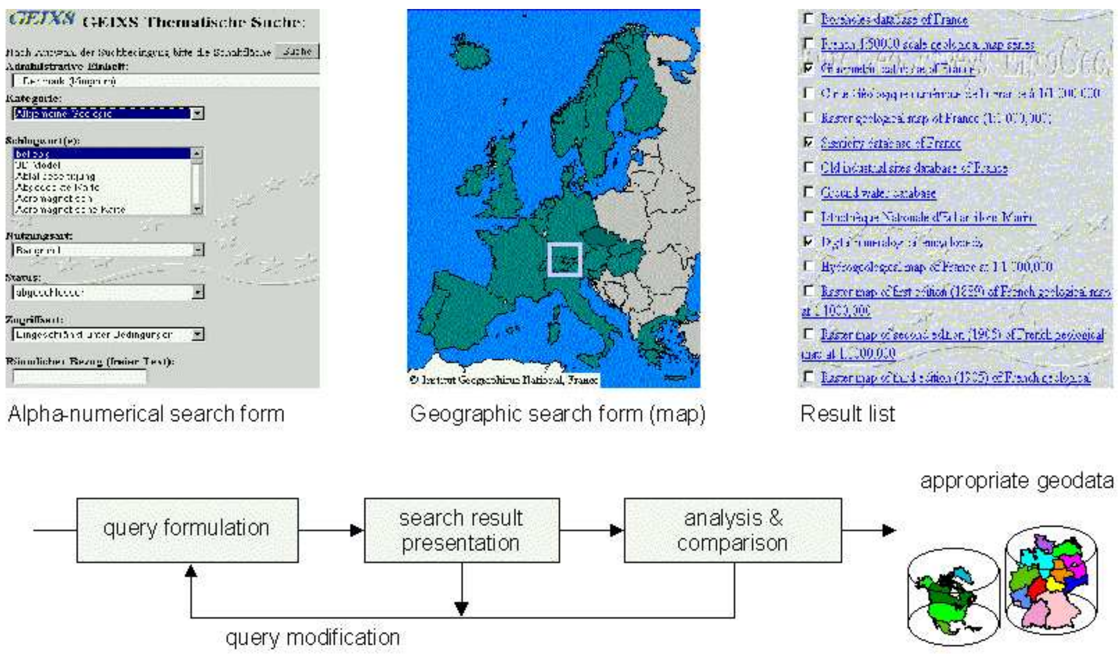


***Figure 3.6**: Information seeking process within MIS for Geo-data [Göb02].*

With regard to search-result presentation, results of the INSYDER [RMMH00] approach have shown that users need support when expressing their information needs

[6]http://www.eurographics.com/gddd/index.htm

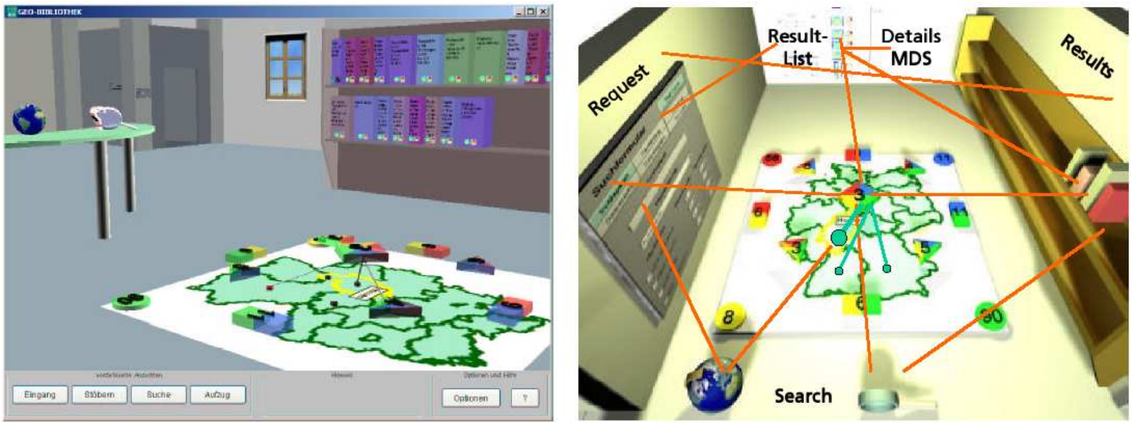


***Figure 3.7**: Domain-specific adoption of Infocrystal: GeoCrystal [GHJ02].*

and reviewing and refining their search results [MR99]).

In general, however, recent initiatives to Geo-spatial data archives offer access to a wealth of distributed data covered by the widespread information spectrum of different Geo-data disciplines (e.g. environmental data, geologic data, cadastral data, remote sensing data or socio-demographic data), but offer only basic levels of interactivity and user assistance. In order to overcome this lack of usability, current initiatives such as INVISIP aim to enhance existing systems by using information-visualization [CM99] techniques. As an example, InfoCrystal by A. Spoerri [Spo93] will be used as a basic approach to improve the information-retrieval process providing visual presentations and intuitive algebraic mechanisms (Venn diagrams) to visualize searches and search-result sets and subsets/classes matching some of the specified search parameters. This approach resulted in GeoCrystal - see Figure 3.7 [GHJ02]. Referring to semantic aspects, the primary trend is to use standards such as ISO 19115 Geographic Information - Meta-data and to provide information brokers merging different application areas and harmonizing terminology (e.g. the alignment of CIP and GEO, see [BHN$^+$98]). Concerning this semantic integration of different information spectra for Geo-data, Göbel has developed a semantic network for Geo-data [GJ01a].

3.4.1 Semantic Network for Geo-data

Whereas fundamental MIS and information visualization techniques primarily help casual users to locate appropriate data and to interpret search results or plans, other mechanisms and more profound knowledge are necessary to facilitate the specific tasks of persons involved in concrete/complex application scenarios, e.g. planning engineers involved in the site-planning process. Here, both visual data-mining techniques as well as expandable context repositories for different application domains are developed within the InGeo IC project.

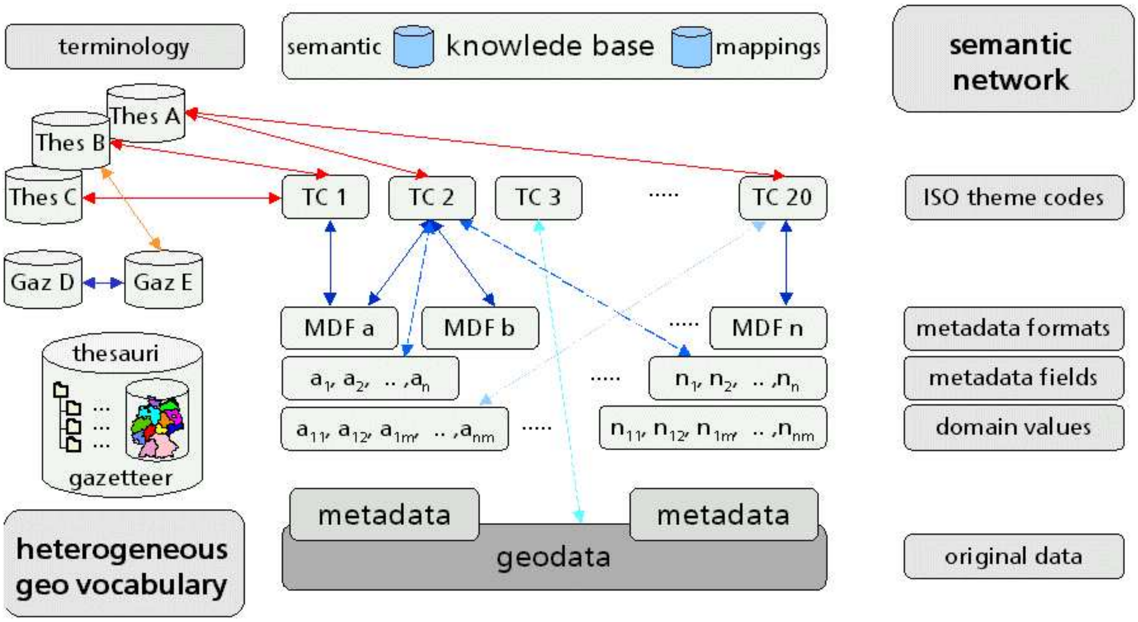


***Figure 3.8**: Semantic Network for Geo-data as basis for the InGeo-MIS [Göb02].*

Figure 3.8 illustrates the development of the Geo-data semantic network, which integrates the different domains of Geo-data disciplines and stores semantic relations among Geo-data archives, meta-data (formats) and theme-specific terminology such as domain values, Valida or thesauri as well as tasks or steps in Geo-data applications.

In this scheme, the ISO theme codes are settled in the center of the semantic network. All Geo-data archives, meta-data formats or keywords are related to these thematic terms and are integrated into the network. Apart from the thematic section of the network that contains content-related semantic relations between terms and theme codes, the network also includes a Geo component. Here, geographic-topological relations are stored to indicate spatial relations between geographic names and entities. These relations build the basis for the spatial-ranking mechanisms presented in section 3.4.2 on the next page.

The semantic network is used as the entry point for the query engine of the InGeo Information Center and it distributes searches to appropriate meta-databases/servers connected to the InGeo-MIS. In addition to traditional access variants to Geo-data archives such as spatial, temporal, thematic (search forms based on elements of the ISO meta-data format) or keyword searches, Figure 3.9 on page 50 shows an abstract thematic access based on the ISO theme codes. This abstract access variant is particulary aimed at casual users who are familiar with the widespread information spectrum of Geo-data disciplines and corresponding terminology within theme-specific disciplines and application areas. Thus, they can implicitly use the ISO theme codes to bridge the different disciplines and to cover semantic relationships among theme-specific application areas.

At a later stage, the semantic network provides a list of keywords (well-known terminology, attributes, domain values, Valida) for each ISO theme code. Subsequently, users can also initiate thematic searches combining the abstract thematic access variant with individual keywords, which can be -optionally- extended by these semantic relationships among the ISO theme codes and further well-defined keywords of special ISO theme codes.

Analogous to the rules for the instantiation and extension of this geographic-topographic semantic network for Geo-data (see [NTK98] and [Wil97a]), the different relationships between application areas or geographic units form the basis for both the information retrieval process in general and individual thematic and spatial ranking aspects.

3.4.2 Ranking Mechanisms

Traditional approaches to access Geo-data archives provide a wealth of infrastructure (networks, search engines, meta-data-based graphic user interfaces) but offer only basic concepts and mechanisms referring to the information-retrieval process within (meta-data based) information systems for Geo-data, taking into account the great variety of Geo-data characteristics.

With regard to thematic access and thematic ranking, most approaches provide keyword searches and are based on full-text mechanisms to retrieve textual documents such as HTML-pages or WORD™documents. Here, most concepts are based on the fundamental retrieval concepts introduced by [Sal89]. The problem with these concepts and solutions (such as ORACLE Intermedia Text™) is that mechanisms and algorithms require a large number of large documents/texts containing high occurrences of keywords in order to obtain high ranks referring to thematic keyword searches. For example, Oracle Intermedia Text™uses the inverted algorithm of Salton, which means there must be more than four occurrences of the keyword to obtain a thematic rank of 100 (in a repository with 1 million documents). For this reason alternatives like LSA/LSI were discussed (see section 3.5 on page 56). On the other hand, the characteristics of Geo-data i.e. the structure of meta-data formats and standards for Geo-data (e.g. ISO 19115 geographic information - meta-data) provides mostly short text fields such as *'title'*, *'keywords'* or *'geo_name'* which are limited to a specific length of characters. Only some meta-data fields enable users to enter long descriptions referring to specific Geo-data attributes. Examples for this are general descriptions such as *'abstract'* or *'quality and lineage statements'* (*quality_narrative*, *lineage_statement*). However, case studies have shown that existing meta-data sets are often not filled very well and comprehensive descriptions/entries within unlimited free-text fields represent the minority.

In Europe this fact is underlined by the abscence of laws or directives such as those that exist in the United States (i.e. the Executive Order 12906 'Coordinating Geographic Data Acquisition and Access: The National Spatial Data Infrastructure', signed in 1994 which obliges data suppliers to document their data via meta-data FGDC[7] format) and provide it to the public. In particular, municipal authorities (in Europe) have not expended much effort on meta-data. However, this is currently changing in the context of the privatization of Geo-data and the necessity to sell Geo-data and establish portals or eCommerce shops for Geo-data.

3.4.3 Spatial Ranking

Based on these facts, Beard and Sharma [BS97] introduced multidimensional ranking techniques taking into account spatial and temporal characteristics of Geo-data. Together they have developed an algorithm which computes a global rank of matching documents/-datasets as the weighted sum of a *'thematic'*, *'spatial'* and *'temporal rank'*, see formulas 3.1 – 3.3:

$$rank_{coordinate} = \frac{target}{candidate} \tag{3.1}$$

$$rank_{coordinate,inside} = \frac{area_{candidate}}{area_{target}} \tag{3.2}$$

$$rank_{coordinate,overlaps} = \frac{q}{r + 100} \tag{3.3}$$

where $q :=$ percentage of overlap between *'target'* and *'candidate'* and $r :=$ percentage of non-overlap between target and candidate (see Figure 3.10 on the following page. Using this algorithm enables the provider of MIS for Geo-data to provide sorted search result lists, but there is still some lack of usability:

- Firstly, the spatial rank takes into account only coordinates. This means that a user specifies a search target (e.g. entering coordinates in a search form or drawing a bounding box on a map) and this target is compared to candidates within a (meta)database. With regard to entering exact coordinates, users often do not know if these are WGS84 or Gauß-Krüger coordinates. This kind of problem could certainly be avoided by using the appropriate meta-data tools (like data-entry tool etc.) so that data does not need to be *'interpreted'*. Besides, it is difficult to specify exact values drawing a bounding box within a web-based GIS component/map. Subsequently this inexactness negatively influences the computation of spatial ranks.

[7] http://www.fgdc.gov/metadata/metadata.html

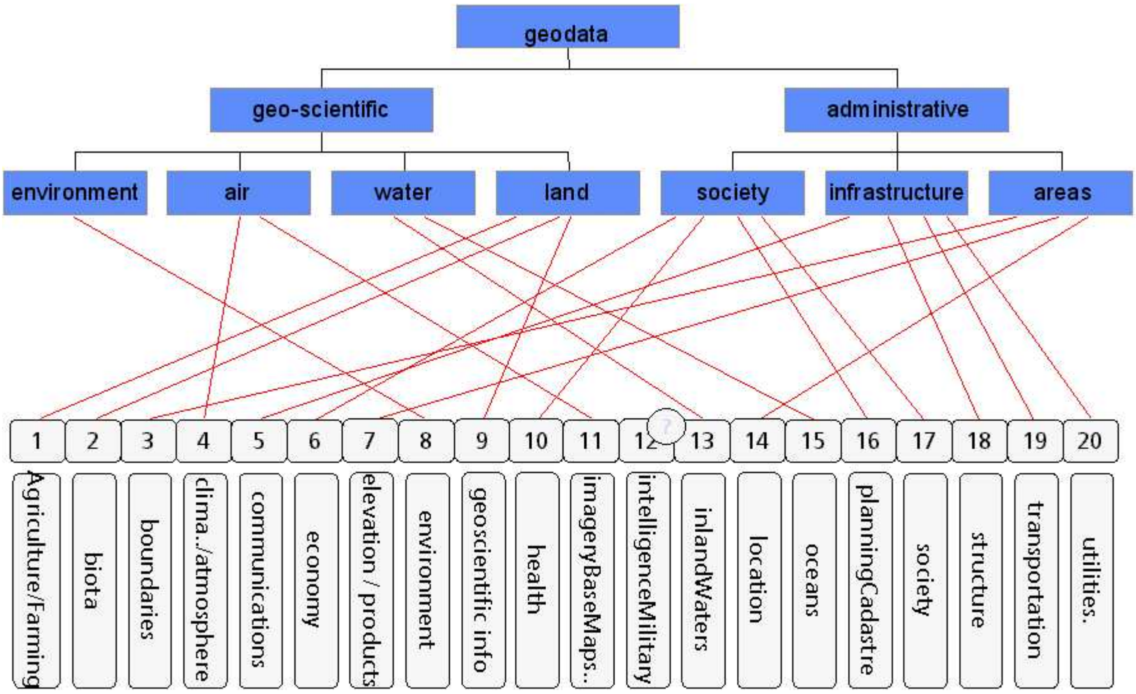


Figure 3.9: *Thematic access to Geo-data based on ISO theme codes [Göb02].*

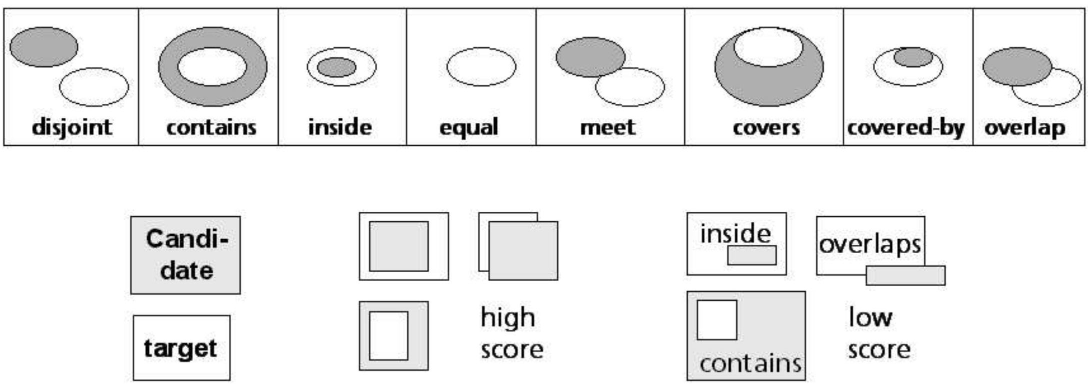


Figure 3.10: *Spatial relationships between search-target and candidates [KG02].*

- Secondly, no hierarchical and topographic relationships are taken into account within the computation of spatial ranks. Of course, hierarchical relationships among spatial units are indirectly addressed by formula 3.2 on page 49, but this takes into account only coordinate-based computations instead of *'hierarchical relations'* and *'topographic relationships'* provided by gazetteers.

In order to overcome these obstacles, Göbel improved the algorithm of Beard and Sharma by increasing the precision of the formula for the spatial rank. This means the computation of the spatial rank is split into a coordinate-based factor $rank_{coordinate}$ and two additional factors $rank_{hierarchy}$ and $rank_{neighborhood}$, see formulas 3.4 – 3.6.

$$rank_{hierarchy} = \frac{level_{target}}{level_{candidate}} \tag{3.4}$$

$$rank_{neighborhood} = \text{target - candidate} \tag{3.5}$$

$$rank_{spatial} = w \cdot \sum rank_{coordinates} + rank_{hierachy} + rank_{neighborhood} \tag{3.6}$$

The $rank_{hierarchy}$ takes into account hierarchical geographic-topographic relationships among geographic/spatial units. This process is supported by gazetteers (= thesauri for geographic units/spatial extents) located at the MIS server within the query engine of the semantic network for Geo-data. The most widespread gazetteers represent administrative areas (e.g. states, counties and districts in the United States or Bundesland, Regierungsbezirk, Kreis and Gemeinde in Germany) or natural structures of countries providing a well-defined set of Geo-spatial entities such as mountain chains (e.g. 'Rocky Mountains', 'Odenwald' or the Alps 'Alpen') or regions (e.g. 'Bay-Area' around San Francisco, 'Great Basin' or 'Rhein-Main Gebiet' around Frankfurt, Wiesbaden and Darmstadt).

Figure 3.11 on the following page is an example showing both map-based and alphanumeric presentations of the two gazetteers 'Administrative Einheiten' and 'Naturräumliche Gliederung' in Germany. Thus, not only the initially specified geographic units/names such as 'Bayern' (state of Bavaria) are used as spatial search parameters, but also broader (next hierarchy would be Germany) and narrower terms (e.g. 'München' - Munich or other regions and townships in Bavaria). Users can employ both access variants to specify spatial searches/search targets in the peculiarities of geographic names of gazetteers or coordinates (polygon, bounding box). Then, the gazetteers are used to extend the initial parameters by hierarchical related terms which results in accepting documents/candidates matching narrower/broader spatial entities but not the spatial entity of the initially specified spatial parameter itself. Referring to the computation of the $rank_{hierarchy}$ there is a database providing pre-computed indices for all possible hierarchical relationships between spatial units within the different gazetteers. Primarily

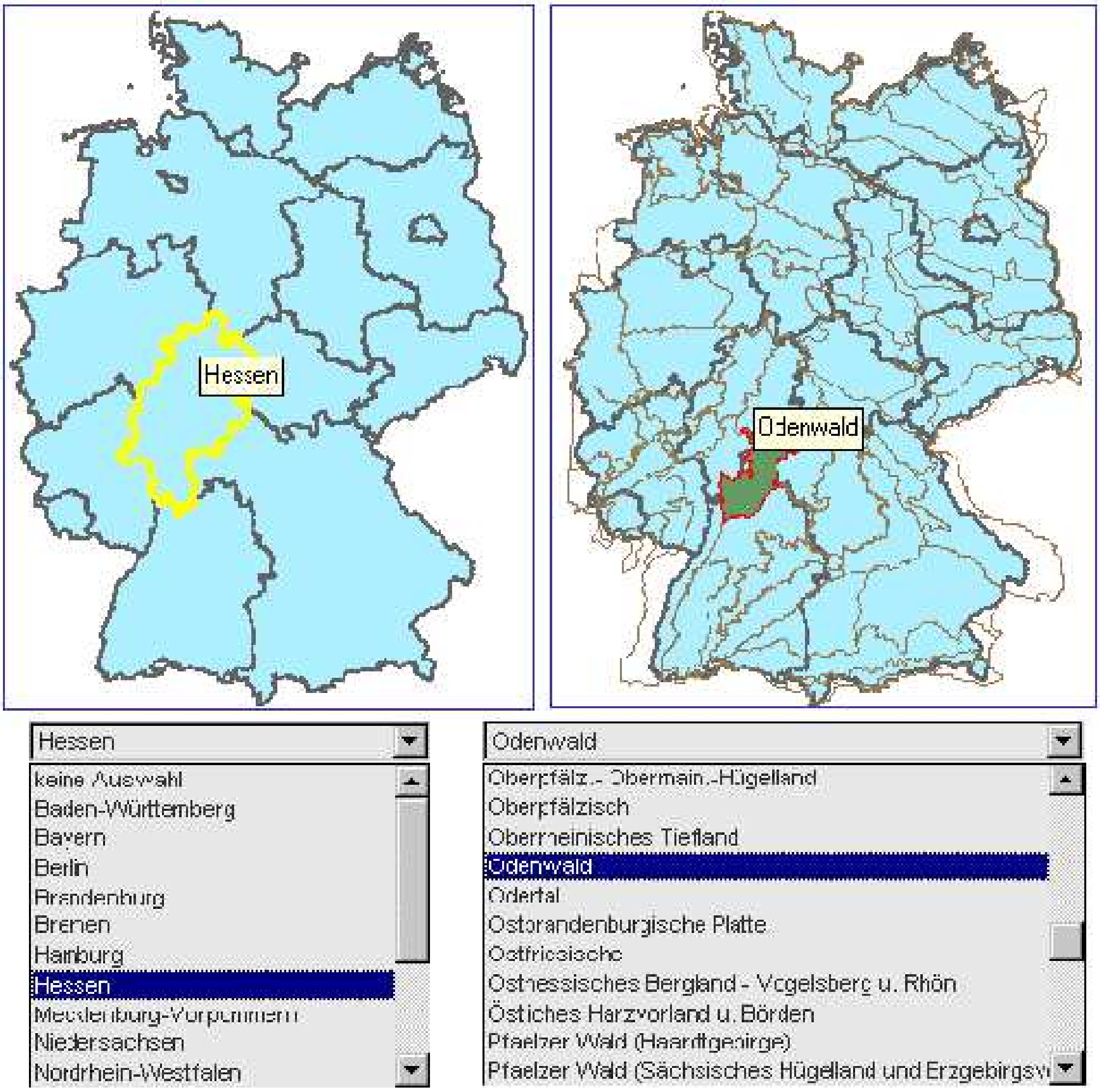


Figure 3.11: *Gazetteers Administrative Units and Natural Structuring for Germany [Göb02].*

(for easy searches) relationships of one gazetteer are taken into account; optionally gazetteer-overlapping relationships among spatial entities are used within advanced searches. Therefore it is not necessary to compute coordinate-based factors between search targets and candidates which results in a better performance of the information retrieval process.

As with the $rank_{hierarchy}$, the $rank_{neighborhood}$ also takes into account the geographic-topographic relationships between spatial units. Here, in contrast to $rank_{hierarchy}$, spatial units are located within the same level of gazetteers. Examples for this are immediate neighbors of towns or villages such as 'Mainz' and 'Wiesbaden', which are topographically related in the direct neighborhood but belong to different parts (branches) of the tree corresponding to the gazetteer (Mainz belongs to the state of 'Rheinland-Pfalz' and Wiesbaden to the state of 'Hessen').

Again, the idea is to accept not only candidates exactly matching the initial search targets, but also candidates located in the immediate neighborhood of the specified search area. This enables users who are not familiar with the targeting search area to locate maps of towns or villages, for example, where they do not know the exact/official name of the broader administrative unit. Besides, sometimes it is possible to find maps which primarily describe spatial areas aside the target search area, but also show some useful information about the intended area (in the immediate neighborhood). From the technical point of view, the computation of these spatial ranks is time-consuming and consequently are only used optionally in advanced search modes. Due to this fact and the fact of uncertainty/incorrectness, there are lower weights for $rank_{hierarchy}$ and $rank_{neighborhood}$ compared to $rank_{coordinates}$. However the use of these ranking parameters and retrieval mechanisms enormously improves the functionality of search engines for Geo-data.

3.4.4 Multidimensional Ranking Mechanism

Taken together, the following formulae show an integrated version of an algorithm to compute the *'spatialrank'* within MIS for Geo-data.

$$rank_{spatial} = w \cdot \sum rank_{coordinates} + rank_{hierachy} + rank_{neighborhood} \tag{3.7}$$

$$rank_{coordinate} = \frac{target}{candidate} \tag{3.8}$$

$$rank_{coordinate,inside} = \frac{area_{candidate}}{area_{target}} \tag{3.9}$$

$$rank_{coordinate,overlaps} = \frac{q}{r + 100} \tag{3.10}$$

$$rank_{coordinate,contains} = \frac{area_{target}}{area_{candidate}} \tag{3.11}$$

$$rank_{hierarchy} = \frac{level_{target}}{level_{candidate}} \tag{3.12}$$

$$rank_{neighborhood} = \text{target - candidate} \tag{3.13}$$

Initial user tests [Göb02] have shown the acceptance of these additional search concepts and mechanisms. With particular reference to 'extreme' result sets (no results, never-ending result list), this extension of the initial search target is quite useful. Thus taking into account not only the initially specified search target (coordinate based or geographic name) but also hierarchic and topographic related spatial units, it first increases the probability of finding matching datasets (candidates). With regard to excessively-long search-result lists, the mechanisms and concepts presented provide the possibility of offering a more detailed granularity of ordered search-result lists providing thematic, temporal and several spatial ranks. Altogether, the integration of these concepts improves functionality of the information retrieval and ranking process within MIS for Geo-data. On the other hand, this causes additional effort from the MIS provider's point of view and decreases the performance of the system. Here, comprehensive case studies and performance tests are currently being performed to evaluate the usability of these concepts, were they to be integrated [KG02].

3.4.5 Temporal Ranking

The approach for the temporal ranking is similar to that for the spatial ranking. As seen in 3.4.4 on the page before, there are formulae which describe the relations of the space of time as well as point in time. A special case is the periodic time interval, but this can be calculated using combinations of the basic formulae.

$$t = \frac{length_{candidate}}{lenght_{target}} \tag{3.14}$$

$$t = \frac{q}{r + 100} \tag{3.15}$$

$$t = 1 - \frac{d_i}{d_0} \tag{3.16}$$

Where $q :=$ percentage of overlap between target and candidate and $r :=$ percentage of non-overlap between target and candidate. Formula 3.16 is for points in time (moments) with d_i as the distance $\|target_{end} - candidate_{date}\|$ and d_0 as the target interval. Formula 3.14 calculates the 'inside' and 'contains' values of the time interval. Formula 3.15 calculates the overlap percentages of the intervals.

3.4.6 Grand Unified Formula

The ideas from 3.4.3 on page 49 to 3.4.5 on the facing page can be summarized in a unified formula for ranking geographic queries:

$$rank = \frac{n_G * G + n_V * V + n_R * R + n_T * T}{n_G + n_V + n_R + n_T} \tag{3.17}$$

The n_i represent the weightings (range from 1 to 10) chosen by the user. G is the ranking for the geographic name, V calculates the thematic (keyword-based) rank, R calculates the rank measured by the spatial-ranking based on coordinates, T represents the temporal rank.

The particular values are calculated as described in the previous sections. The user profits from this transparent ranking schema in using this information in the query refinement process.

3.4.7 Practical Aspects (ISO 19115)

With regard to the use of these information-retrieval and ranking mechanisms within MIS for Geo-data, the following list of meta-data fields of the upcoming ISO 19115 meta-data standard for geographic information was chosen:

- Referring to thematic ranks, primarily the fields *'title'* and *'keywords'* (category information) are useful for keyword-based searches. Additionally, the fields *'abstract'*, *'qualitative_narrative'* and *'lineage_statement'* could be taken into account for extended thematic (keyword) searches using full-text retrieval mechanisms. Further on, the fields *'keyword_thesaurus'* and *'theme_code'* are important with regard to the (semi-automatic) extension of initial searches using the semantic network for Geo-data, see 3.4.1 on page 46. The field *'theme_code'* is particulary useful, because it represents the center of semantic network assigning keywords to Geo-data application areas and vice versa. In this context, additional thesauri (field keyword thesaurus) are used as catalysts in order to specify well-defined terminology and extend initial searches with these well-defined search terms. Thus the probability of finding matching candidates is greatly increased.

- Concerning temporal-search aspects, the category *'temporal_extent'* provides the fields *'begin_date'* and *'end_date'*, which could be used for individual dates and time ranges (using both). This could also be used for content-aging strategies see section 3.1 on page 32

- Spatial-ranking aspects deal with the category extent-providing fields for bounding boxes, exact polygons of spatial units, geographic names and gazetteers. Here, from the MIS provider's point of view, the search engine receives input from a graphical user interface (web portal to Geo-data), extends the specified spatial parameter(s) using the semantic network and contacts meta-data servers connected to the MIS network. The retrieved information about matching datasets contains title and hyperlink to the complete meta-data set as well as a set of spatial ranks. These ranks are summed with thematic and temporal ranks and are presented in a ranked search-results list within the MIS browser (GUI) in addition to the title plus hyperlinks to the meta-data sets.

As mentioned in section 3.4.1 on page 46, the MIS server (query engine) requires additional thesauri and gazetteers to provide advanced search and retrieval mechanisms introduced within this chapter.

3.5 Summary

Meta-data plays various important roles in a visual information seeking system. The complex conditions of data, user and task should be considered from the start as equally important and tightly linked with each other. The analysis of the available (meta-)data and the contextual task analysis are an example for such a situation.

A strategy for the maintenance of meta-data should cover its whole lifecycle. Entry tools that guarantee the correct and definite acquisition belong to every meta-data toolbox. Automated test settings and evaluations are additional tools that should be considered from the start.

A well cared-for meta-data repository represents a valuable source for professional users of a visual information-seeking system. A ranking based upon meta-data is a logical supplement to classical text retrieval. The user has the possibility of optimizing queries based on his/her knowledge of the underlying information space. Section 3.4 on page 43 showed how such meta-data ranking performs. A generalized extrapolation can be made as follows. A closer look at the kind of meta-data reveals four different categories:

1. Numbers and numerical ranges (e.g. spatial and temporal meta-data)
2. Hierarchies (e.g. *'gazetteers'*)
3. Free text (e.g. *'abstract'*)
4. Categories (e.g. '$theme_{Code}$')

Rankings for numbers and ranges were presented in section 3.4.3 on page 49.

Hierarchies can be computed by depth level or membership of a part-tree. This is very context sensitive (see the example for gazetteers in the INVISIP project on page 52). What should be a good match? Neighborhood? Level? Without answering this question it is difficult to make generalizations about a ranking schema for meta-data hierarchies.

Because of the usual text sizes (very small) and the homogeneous vocabulary it is no simple matter to form a ranking mechanism for the text fields. Traditional text-retrieval algorithms must be optimized in order to deliver good results. In the INVISIP project a modified version of the *'latent semantic analysis'* [DDL$^+$90] was recommended. Early tests were very promising. A final result can be estimated in the Bachelor Thesis of Sebastian Rexhausen [Rex05].

In contrast to the rankings above, the computation of categorical meta-data is quite trivial. The only consideration is whether the meta-data contains this element or not, leaving us with a simple summation over all categorical data.

According to these considerations, a generalized meta-data ranking scheme looks like equation 3.18, where a summation over the differnt kinds of meta-data produces a ranking.

$$sim(md_n, q) = \underbrace{\omega_n \cdot \sum_{i=1}^{t} \frac{1}{|d_i - q|}}_{\text{Numbers}} + \underbrace{\omega_c \cdot \sum_{i=1}^{t} d_i}_{\text{Categories}} + \underbrace{\omega_t \cdot \sum_{i=1}^{t} LSA(i)}_{\text{Texts}} + \underbrace{\omega_h \cdot \sum_{i=1}^{t} custom(i)}_{\text{Hierarchies}} \tag{3.18}$$

Where $\omega_{n,c,t,h}$ are the individual weights; the sums count over all available meta-data fields of the appropriate kind (t). The different parts of this equation must, of course, be adapted to specific context. Figure 3.12 on the following page is an example showing a graphical user interface which offers these advanced query possibilities (the InGeo Information Center as theme-overlapping MIS for Geo-data in Germany). Here, users can specify the different search parameters within simple and advanced search forms as well as maps. On the left side one can see the different search tabs, which are categorized in thematic, temporal and spatial forms, as well as the switch between search and 'advanced search'.

The forms offer the possibilities of entering the geographic area as seen in a Gazetteer, or by coordinates, as well as by names of natural structures. The user can weight his search parameters. LVS (Limited Vocabulary Search) is used to minimize the probability of input errors by the user.

It is possible to weight each query term via a priority function (see figure 3.12 on the next page: the slider in the left bottom corner).

Such query masks offer powerful possibilities to the user, but these are usually associated with an increased cognitive load. A simple preview of the estimated result set, as well as some information on how the result set is composed, is usually missing as are other features that would reassure the user. To empower the user with such features the approach of using visualizations instead of traditional forms seems promising. The next chapter will cover visualizations that are intended for query preview and filtering of multidimensional meta-data sets.

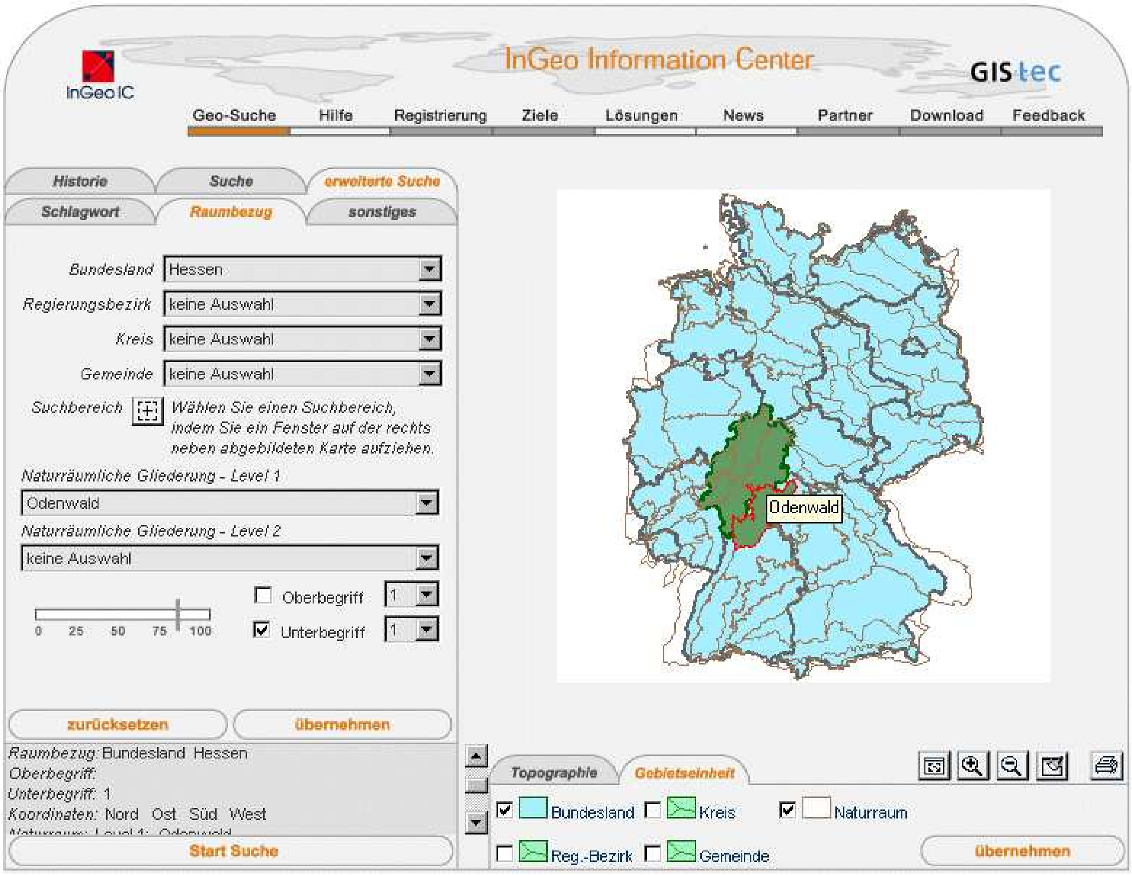


***Figure 3.12**: Graphical User Interface of the InGeo IC - Advanced Search Mode Germany. http://www.ingeoic.de*

4 VISUALIZATIONS AND HUMAN-COMPUTER INTERACTION

Contents

After considering the issues of the previous chapters, the question of *how* the user will interact with the information system arises. A classical approach can be seen in Figure 3.12 on the facing page where standard widgets are used in a powerful and complex form. This kind of a query has the advantage of offering users an interface that they expect[1]. But this point should not, of course, prevent research on advanced, optimized and alternative user interfaces. Such alternatives have to provide added value to the user in order to be accepted. This added value comes with a learning curve which varies depending on the scenario, the user and the visualization. If, for example the user is a professional Information Broker, it is likely that he/she is ready to invest in training

[1] in fact the point of *expectation* is part of ISO 9241-10: Dialog Principles for User Interfaces

time so as to increase the future efficency and effectiveness of his/her work. On the other hand an information system that is aimed at novice or casual users (web search engines or library information systems) must have a *self-explanatory*[2] system.

The research project INVISIP clearly defined professional users as the main target group. Their application domain owns a very complex and powerful meta-data standard (ISO 19115) as a foundation for the information space. Information Visualization possesses concepts and technologies to handle those situations and to literally not lose the *overview*. Information Visualization enables the user to discover interesting and useful results by the use of highlight patterns. Objectives such as retracing a known item, finding a small set of items in a large collection or discovering new patterns (similar to data analysis or visual data mining) can be improved with regard to effectiveness and efficiency when visualizations are used.

This chapter will cover the various tasks for a query and filter visualization (section 4.1). The appropriate concepts and interaction techniques will be covered in section 4.2 on the facing page. Besides these more theoretical sections, some related work will be presented in section 4.3 on page 75. A summary of the most important criteria for the development of the CircleSegmentView finishes this chapter.

4.1 Tasks

According to Shneiderman the seven basic tasks involving information manipulation that are possible for a user are: *'Overview'*, *'Zoom'*, *'Filter'*, *'Details on Demand'*, *'Relate'*, *'History'* and *'Extract'* [Shn04]. From these seven tasks only a fraction need be considered for the CircleSegmentView. These remaining tasks (Overview, Filter, Details on Demand) are examined more closely in the following subsections:

4.1.1 Overview

'Overview' is used to present a large amount of content that is too big, complex or dynamic to show in an obvious form. It is used to offer users an overall structure of the content and give them the possibility of traversing the content at their own pace, in the order of their choice.

This is a way of dealing with complexity: present a high-level view of what's going on, and let the user *'drill down'* from that view into the details as they need to, keeping both levels visible for quick iteration. *'Overview Plus Detail'* breaks up the content into comprehensible pieces, while simultaneously revealing their interrelationships to the user.

[2] this is also a demand from ISO 9241-10

Edward Tufte uses the terms *'micro and macro readings'* [Tuf90] to describe a similar concept for maps, diagrams, and other static information graphics. The users have the large structure in front of them at all times, while being able to peer into the small details at will: 'the pace of visualization is condensed, slowed, and personalized'. Similarly, users of *'Overview Plus Detail'* can page methodically through the content, jump around, compare, contrast, move quickly, move slowly, or even rearrange it.

Finally, the overview can serve as a 'You are here' sign. Users can tell at a glance where they are in the larger context. Most interfaces that use *'Overview Plus Detail'* present the overview as either a linear or a hierarchical set of selectable objects. Users generally know what to expect from lists and trees.

The concept of *'Overview'* may be optimized if it is combined with the results of Susan Dumais' and Hao Chen's [CD00], work which predicts that categorized sets are easier to access than the raw and unstructured representation.

4.1.2 Filter Task

'Take out the uninteresting items'. The goal is to give users easy controls with rapid display updates, no matter the amount of data presented. For a screen capture of the Spotfire demo see Figure 4.1 on the following page and 4.2 on page 63.

In those cases where too much information is available, there is a need to filter the information in some way. If this filtering takes the form of selecting a subset of the data along a range of numerical values of one or more dimensions, it is called *'filtering by zooming'*. Filtering and zooming work by reducing the amount of context in the display; this distinguishes them from the *'focus+context techniques'*, which attempt to retain all the contextual information even if it must be drawn so small as to make it virtually invisible. This information visualization technique of filtering through a zoom task is illustrated in Figure 4.2 on page 63.

4.1.3 Details on Demand

'Details on demand' is defined as the selection of an item or a group of documents and the extraction of more details when needed, once the entire collection has been reduced to a few items. Details on demand are usually implemented as tool-tips or pop-up windows.

4.2 Concepts

This section covers some strategies concerning interactive systems that are valuable for the development of the CircleSegmentView. An comprehensive overview of available concepts can be gained in [CM99], [Shn04] and [War00].

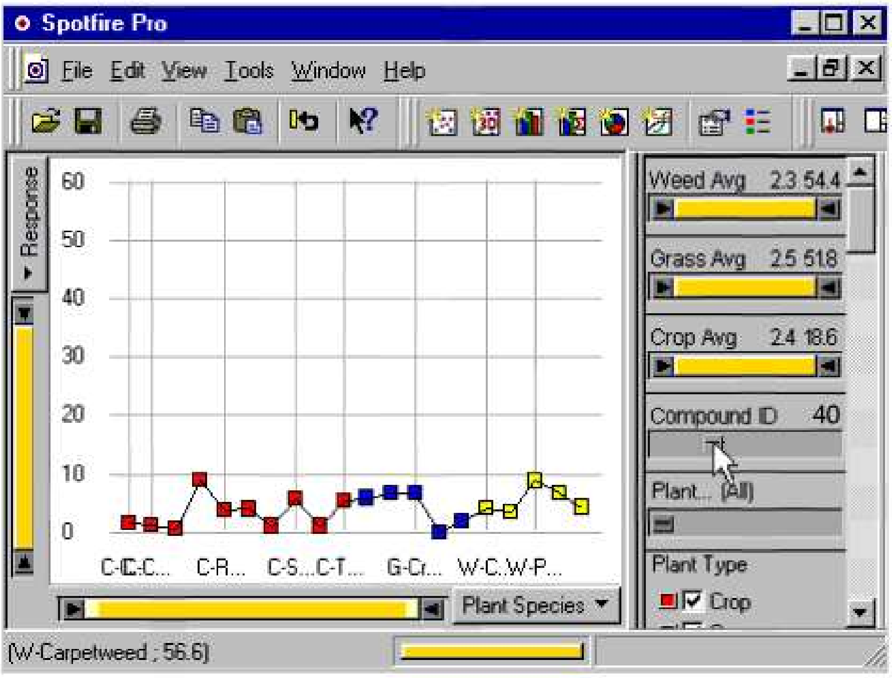


Figure 4.1: *The figure shows the scrollbar allowing the selection of one single compound for filtering (fourth slider from above on the right side leaves only one item visible). (www.spotfire.com).*

4.2.1 Multidimensional Data

Designing visualizations for *'multidimensional data'* is difficult because human perception can not distinguish more than three dimensions (let alone the time as a fourth dimension). Trying to map higher dimensional data to 3D (e.g. using 3D scattergrams) often results in disorientation and occlusion. Parallel-coordinate plots are one of the few truly compact multidimensional techniques [ID90] (see Figure 4.3 on page 64).

Other popular multidimensional visualizations are the *'TableLense'* (see Figure 4.4 on page 65), *'VisDB'* (see Figure 4.5 on page 65) and *'InfoZoom'* (see Figure 4.6 on page 66).

4.2.2 Multiple Coordinated Views

Result-set presentation with the help of visualizations is becoming a more and more common technique (see e.g. [Shn04]). [Kei02] describes the use of visualizations as follows: "The basic idea of visual data exploration is to present the data in some visual form, allowing the human to get insight into the data, draw conclusions, and directly interact with the data". *'Multiple Coordinated Views'* (MCVs) take another step further

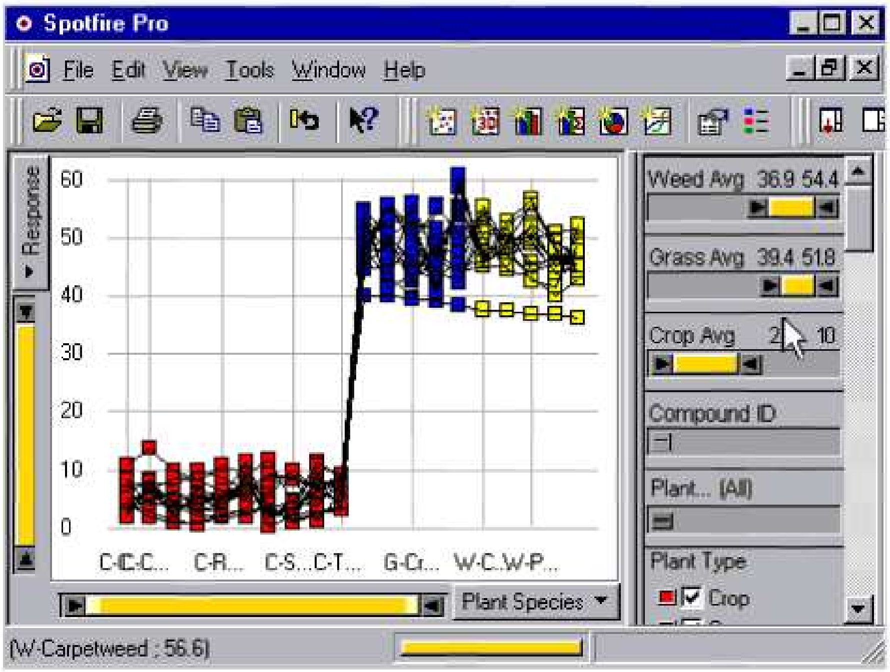


Figure 4.2: *This figure shows that by manipulating the various scrollbars on the right part of the screen related to the reaction of each type of plant to compounds it is possible to filter out unwanted data (www.spotfire.com).*

and improve the possibilities given by a single visualization. Over the past few years MCV has become more and more popular. A lot of systems use this approach to provide better access to the mass of data users are confronted with. Nevertheless, there are drawbacks that have to be weighed up. If the advantages preponderate, it can be meaningful to use MCVs: if not the idea should be abandoned. The design decision to use MCVs for search result visualization is a very important and significant one. Not only do the visualizations have to be chosen, but also the coordination between them, which has extensive consequences for the implementation of the system's design. [WWK00] presents eight guidelines for the design of multiple coordinated views. These can be used as the basis for a decision process on whether to use MCVs or not. The guidelines are organized in two main sections; the first part deals with the situation when multiple views are preferable, i.e. it supports the designers in coming to a decision. Part two deals with the usage of MCVs, i.e. costs (such as space used, cognitive attention, etc.) that arise working with MCVs should be minimized. Unfortunately, trade-offs exist among the rules - this is common in reality but has to be investigated more than ever in this case.

To avoid misunderstandings, the following definition of the phrase 'Multiple Coordinated Views' will be used in the following text.

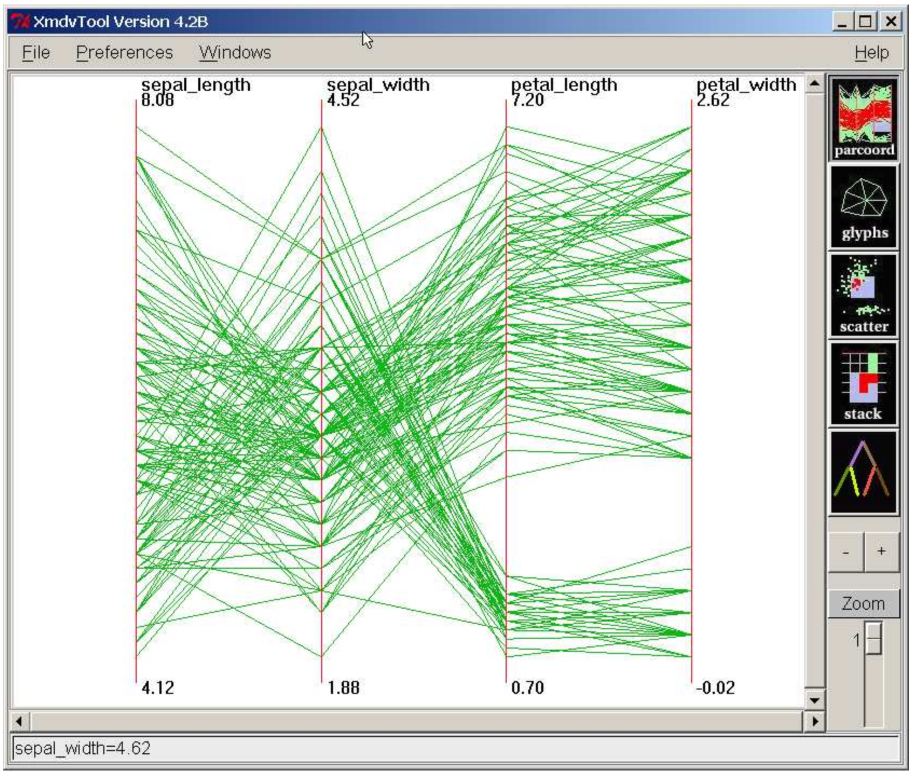


Figure 4.3: *Parallel coordinates used to visualize a four-dimensional iris dataset with 150 data items. The image was generated with XmdvTool*

Definition 4.1 *Multiple Coordinated Views (MCVs) consist of different single visualizations that are linked together by specific interactions. "Different" means that either the data itself or the visual representation of the data vary. The interactions between these different views can be realized by selecting and/or navigating.*

4.2.3 Brushing and Linking

Interaction is one of the most important features in information-visualization systems. Especially for visualizations that offer several views simulaneously, it is necessary to provide control and feedback for the user.

'Brushing', for example, is a process where the user can *highlight*, *select*, or *delete* a subset of elements by pointing to the elements with a pointing device [Wil96]. In situations where multiple views of the data are shown simultaneously, *'brushing'* is often associated with *'linking'*, where brushing in one view affects the same data in other views. In linked views, the result of a brushing interaction is shown in all involved views. This means that points which are selected by the brush in one view are highlighted in

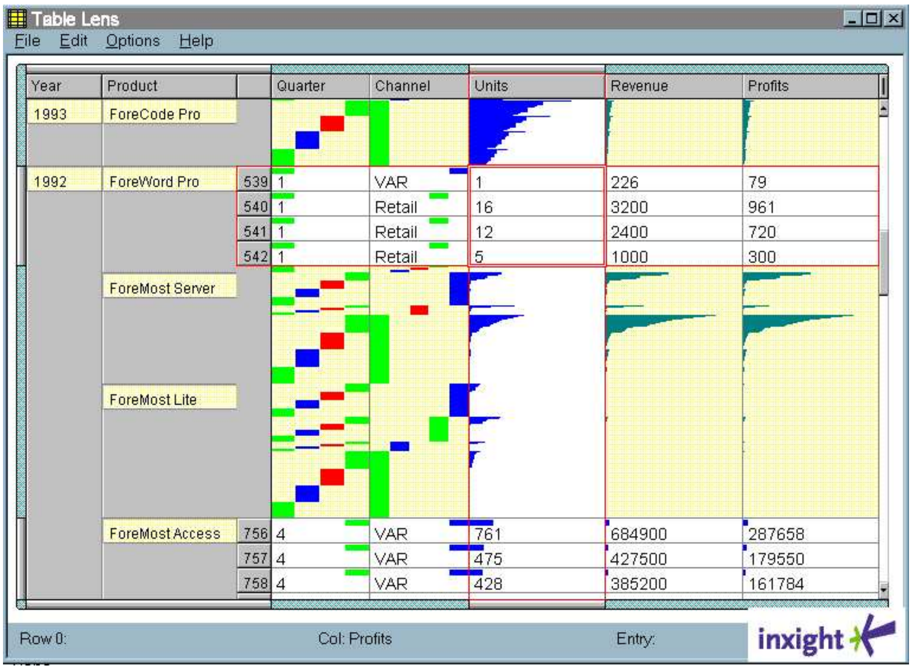


Figure 4.4: *TableLense of Xerox Parc: Some rows from a large spreadsheet are in focus while the rest of the spreadsheet is represented by histograms and charts.*

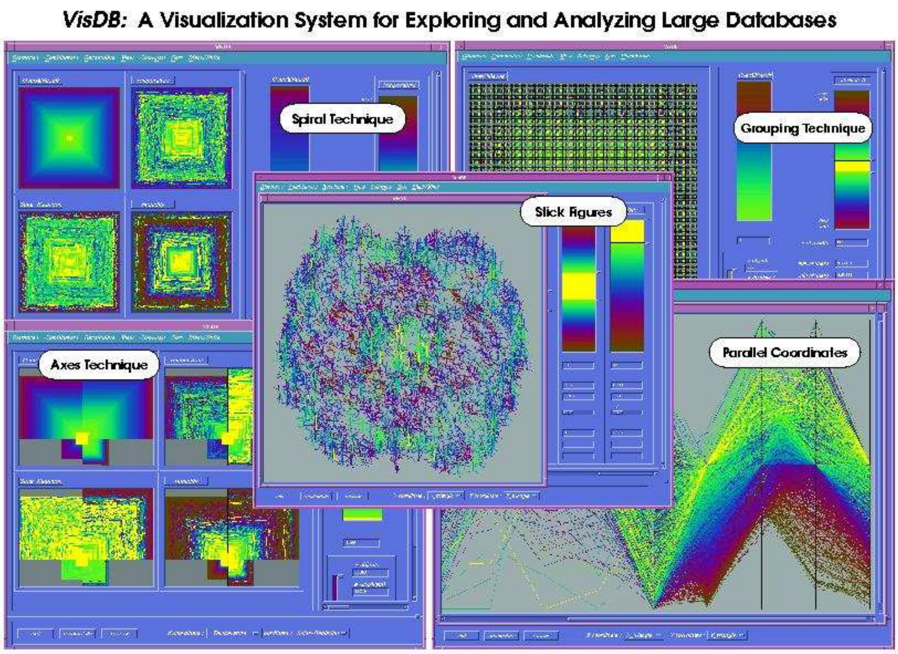


Figure 4.5: *VisDB uses spirals and color-coding to show the relevance of results to the query and gives multiple views of these results (e.g. parallel coordinates).*

***Figure 4.6**: InfoZoom (available from www.humanIT.de) showing the overview of data by categories (producer) and size coding (price).*

other views as well.

'Brushing and linking' is a basic interaction technique for selection and interest-highlighting in this work. Brushing techniques have already been applied to high-dimensional scatterplots [BC88], where the user specifies a rectangular region to select a subset of points in one of the 2D scatterplot projections, and based on the operation mode, corresponding points are highlighted, deleted, or labeled in other views.

4.2.4 Tight Coupling

'Tight coupling' is a strategy in the design of query mechanisms in direct-manipulation query systems. *'Tight coupling'* helps users navigate toward high-precision queries in a space of database queries, avoid empty query results, and quickly narrow down the number of possible and meaningful queries. *'Tight coupling'* of a query mechanism is defined as:

Definition 4.2 (Tight coupling:) *The results of user operations (querying, zooming, panning) on query devices (starfields, rangesliders, Alphasliders, and toggles) are reflected in all query devices by visual feedback and physical constraints on meaningful query settings [AW95].*

Query devices and their related query-formulation mechanisms are designed to interact with each other by restricting users to query criteria that lead to non-empty query results - which also provide users with important feedback about the state of the query mechanism. This interaction should be *rapid*, *incremental*, and *reversible* (following the principles of direct manipulation) [Shn04]. A tightly-coupled query can be regarded as a series of filters selecting a subset of a database. For each new filter that is added, users can only select filter values letting through at least one database object still existing after the last filter - see Figure 4.7 on the next page.

However, this intuitive view of the query does have disadvantages. Primarily, it imposes a unnecessarily-strong sequentiality on the query. After having selected one filter value, users cannot change their mind and select another value on the same level, as only those filter values existing in objects below are selectable after the manipulation. Tight coupling aids users through the query process in several ways:

- **Selection** of a sequence of manipulations is simpler, as the number of possible actions (leading to meaningful query results) narrows down quickly due to the tight coupling.

- **Manipulation** of the widgets, especially Alphasliders, is quicker and simpler as the number of choices is smaller.

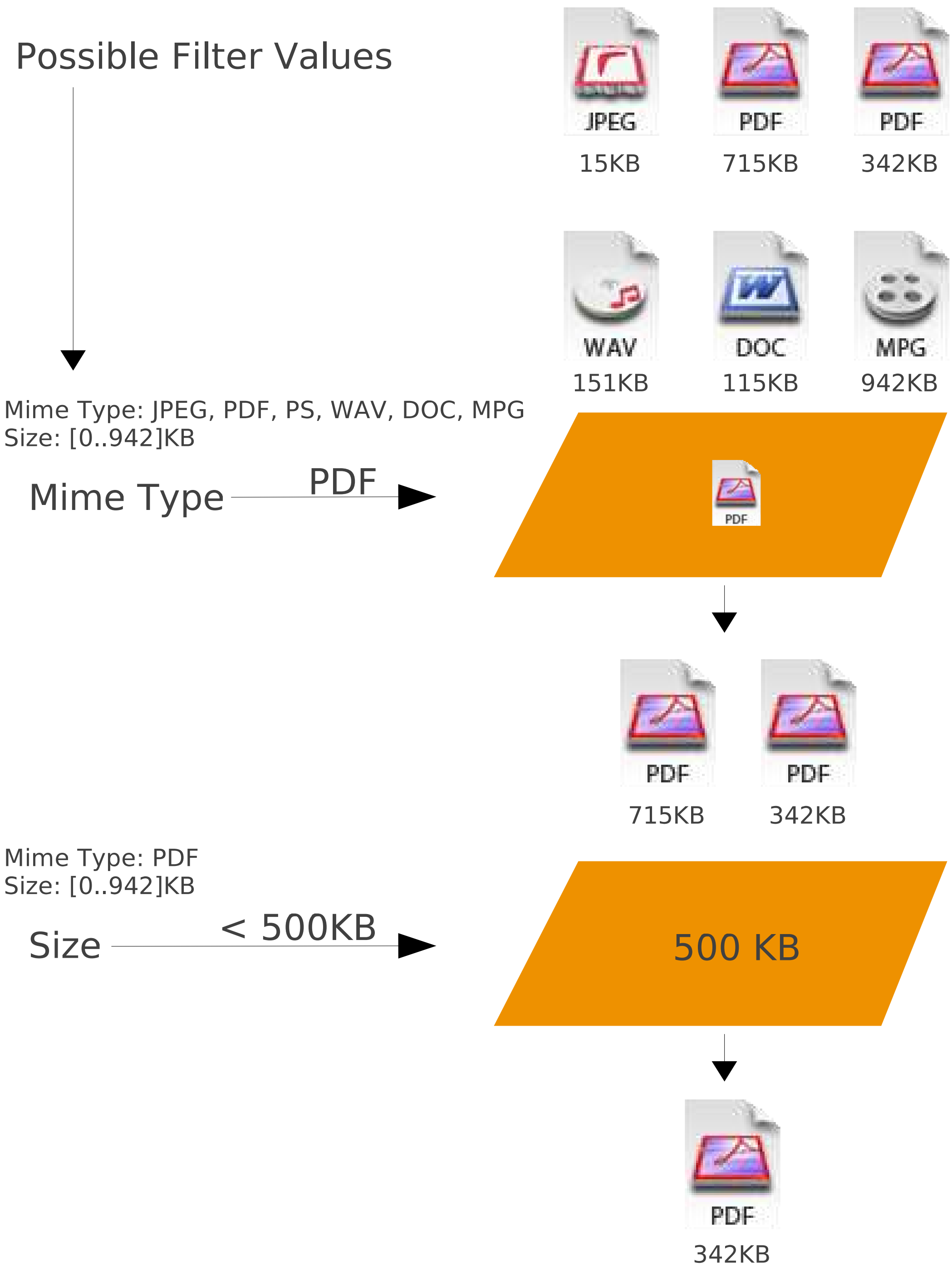


***Figure 4.7**: Intuitive view of a tightly coupled dynamic queries system with objects holding attributes of two types; mime types and size. After a filter has been applied, only those values existing in objects which passed the previous filters can be selected as filter values, thus avoiding empty query results.*

- **The visible states** of the widgets reveal important information about the state of the query mechanism and enhance the interpretation of the visualization.
- **Evaluation** of the query result is enhanced, as the information revealed about the state of the query mechanism can often be used directly to evaluate the result set for relevance.
- The more powerful interpretation and evaluation stages help users in the reformulation of their **exploration goals**.

4.2.5 Semantic Zoom

A typical example for a system realizing semantic zoom is *Pad++* by Bederson [BH94]. In this context a set of web documents can be displayed as small thumbnails or icons, showing a small set of details. At this level users can be given an overview to recognize the global connection. To get more information, the have to look deeper into the document. This can be done by zooming in, to increase the amount of information progressively up to the actual document itself.

Financial data concerning the business volume of an enterprise are another example of dividing information in different levels of detail. The first level presents the business volume for a whole year. Following the drill-down concept, one is able to view the data summarized for three months at a time by increasing the level by one. The next step displays the data monthly, and so on. This approach clarifies the idea that, from step to step, one receives more information that one was not aware of previously. Furnas laid the foundation stone for the idea of semantic zooming by introducing the *Fisheye View* [Fur81]. He addresses the fact that the amount of data grows, though the space to display the data still remains small, limited by technical restrictions (screen size) and by human visual-processing capacity. The problem arises of deciding what part of the information to show. Therefore the *'Degree of Interest (DOI)'* function is established to support the decision process. Three properties have to be defined to calculate the degree:

1. A focal point (or focus) FP.
2. The distance from the focus $D(FP, x)$, where $D(FP, FP) = 0$, and
3. The level of detail (importance, resolution) $LOD(x)$

The focal point FP describes the current point of interest, the distance D measures the semantic distance between points and has to be defined for any point x (may be a simple linear distance, or a more structurally-defined one), and the LOD measures the importance of a point x, dependent on the global structure, also known as 'a priori importance' [Pre99]. The definition of the degree of interest at a given point x can now be written as:

$$DOI(x\|FP) = LOD(x) - D(FP, x) \tag{4.1}$$

The absolute value of the DOI function is only of small interest. Nevertheless, it is a measure for comparing the importance of different objects in order to decide what should be displayed, and when. The distance D (as the static part) and the level of detail LOD (the dynamic part) have to be weighted in a convenient way. If the LOD is very small compared to the distance value, the layout is almost exclusively dependent on the latter, and vice versa. This weight has to be controlled by the context. The *Fisheye view* that is introduced is implemented as a focus and context technique that makes it possible to unify overview and detail in a single view. A classification of *Fisheye views* (see [Noi94]) can be done in the following way:

- **Distorted presentation**: adaptation of size, position, or shape of objects onto the DOI
- **Filtered presentation**: comparison of DOI with threshold value - the result determines if an object is presented or not
- **Decorated presentation**: adaptation to the DOI in respect of specific presentation variables, like color, transparency, font, etc.

Usually, the implementation is a combination of more than one presentation style. Preim introduces a zoom technique called *'Zoom Navigation'* [Pre99]. Additionally to the DOI he defines an *'Aspect of Interest'* (*AOI*). The idea of the AOI is to analyze user interactions and to draw conclusions for the desired information. These two approaches must be applied to define a 'representation matrix', where the DOI determines the matrix row whereas the AOI is responsible for the column in this row. This implies that different aspects share the same DOI. The AOI is defined as

$$AOI(aspect_k) = f(N, t_1, t_2) \tag{4.2}$$

where N is the number of visits for $aspect_k$, t_1 stands for the duration of visits and t_2 defines the last visit. As an example imagine the following scenario:

Example 4.1 *A student is interested in a film of Charlie Chaplin. The first information he gets from the library information system is the title. Zooming in one step provides a short abstract about the content. Up to now there is always one AOI in each level. Further zooming in can now display different data, e.g. year of origin, name of actors, available language. The corresponding representation matrix would look like this (table 4.1 on the facing page):*

***Table 4.1**: Representation Matrix for Library Scenario*

<table>
<tr><td></td><td colspan="3">Aspects</td></tr>
<tr><td rowspan="3">DOI</td><td>Title</td><td></td><td></td></tr>
<tr><td>Abstract</td><td></td><td></td></tr>
<tr><td>Year of origin</td><td>Name of actors</td><td>Poster</td></tr>
</table>

4.2.6 Dynamic Queries and Query Preview

Definition 4.3 (Dynamic Queries:) *This technique allows the user to change the query parameters and see the display update in real time. Dynamic queries continuously update search results' within 100 milliseconds; users adjust sliders or select buttons to ask simple questions of fact or to find patterns or exceptions. [WS92]*

Dynamic Queries are usually characterized by the following attributes:

- Interactive control
- Visual query parameters
- Rapid display update
- Animated display
- Visual presentation of query components
- Visual presentation of results
- Rapid, incremental and reversible control of the query
- Selection by pointing, not typing
- Immediate and continuous feedback

Typical benefits of Dynamic Queries: A user can quickly discover...

- Which areas are densely populated vs. sparsely populated
- Clusters
- Exceptions
- Gaps
- Outliers
- Trends
- Patterns

Visual overviews of an entire database by *starfields* (zoom-able scattergram of color points), *tree diagrams*, *treemaps* (nested rectangles that show hierarchies), *parallel coordinates*, *network diagrams*, and other strategies are making visual browsing and dynamic filtering viable. As users select widgets such as sliders, buttons, and maps, the result list is changed, often within 100 milliseconds, thereby enabling rapid exploration [AS94b]. The Visual Information Seeking strategy is: Overview first, zoom and filter, then details-on-demand [Shn04].

For large collections, especially when searching across the Web, search actions can be split into two phases. First, a rapid rough search that previews only the number of items in the result set, and then a query refinement phase that allows users to narrow their search and retrieve the result set [DPSB97].

Query previews require database maintainers to provide an updated table of contents that users can download from the server. Then users can perform rapid searches on their client machines. The table of contents contains the number of items satisfying combinations of attributes, but the size of the table is only the product of the cardinality of the attributes, which is likely to be much smaller than the number of items in the database. As the example of the *RestaurantFinder* (see page 75 and Figure 4.10 on page 76) with twelve kinds of restaurants, eight regions, three kinds of charge cards illustrates, a simple table of contents would contain only 288 entries. Storing the table of contents burdens users who may have to keep tables of contents (1000 to 100,000 bytes) for each database that they search. Of course, the size of the table of contents can be cut down dramatically by simply having fewer attributes or fewer values per attribute. These burdens seem moderate when weighed against the benefits, especially if users search a database repeatedly. The table of contents is only as big as a typical image in a website and it can be automatically downloaded for use when Java applets are used.

The method of dynamic querying mapped to networked querying environments appears to be useful. However, high system-resource needs make dynamic querying less applicable, at least at first glance, to huge networked information depositories. A solution to this problem might be found in the division of the bigger problem into smaller problems. This idea resulted in the design of Query Previews [Tan01]. The goal is to give an overview of the database to the user before the details are presented.

A good preview should enable users to see sufficient detail about the database in order to better understand the data distribution and then make an informed query. Furthermore, the querying process should be divided into steps to reduce the amount of resources needed to form the queries that are submitted. So a multi-phase incremental querying process will hopefully lead to a set of desired results using less resources and time. In a two-phase approach the designer chooses a few of the most discriminating attributes of the database as the first phase (possibly those that form a primary key for that relation of the database). Then, by using the direct manipulation approach, a user interface should be designed using this small number of discriminating attributes. The rest of the attributes should be kept for a second phase that will also include the discriminating attributes [TPS00].

The first phase, when concatenated with the second one, should form an interface for querying on the related database. When the querying environment is activated, the first interface should show up immediately. The user should make some decisions using this first interface (*Query Preview Panel*) and then move to the second one (*Query refinement Panel*) [TPS00]. The second interface should be used to complete the query.

The Query Preview Panel is a powerful tool for defining approximate ranges on the data-set that is being manipulated. The reason for this is that it contains the most discriminating attributes in the database, so that any range constraint will lead to a small subset of the database. In addition, it does not consume substantial system resources because only a small number of attributes from the database are used at this phase.

To guide users in the query formulation process the preview panel presents aggregate information about the database. The data distribution on an attribute of the data-set is shown before the actual query formulation. The distribution of data over an attribute can be shown as a pie or a bar chart. When users make a selection on any of the attributes of the preview panel, the rest of the user interface should be updated accordingly. Therefore, for each action that users take, feedback is given. As users can see the possible size of their query before refining the rough ranges, there is little chance that they will get zero or mega hits for the final query itself. The system load will drop drastically for downloading the necessary hit-set due to the limited size of hits for the rough query. Perhaps the most critical advantage of the previewing idea is that the administrator of the system needs only the aggregate information about the system at this phase. So whatever the size of the database is, we only need the distribution information about the data to form a preview panel (this again decreases the system resource needs). An example Query Preview Panel is given in Figure 4.8 on the following page.

In most of the Query Preview implementations, the size of this array, which contains the necessary aggregate information, is small with respect to the data-set size. For example, the preview in Figure 4.8 on the next page can be represented with only 1840 integers and in most of the current systems this is less than 10 Kbytes of information. The Internet is fast enough to accommodate the transfer of 10 Kbytes very easily over large distances and even at high levels of congestion (this is especially important in relation to the total size of the database - in the case of NASA, this is generally more than a terabyte of information). In addition, most of the data structures presented by Tanin [TPS00] use a similar approach to represent the data internally for DQIs.

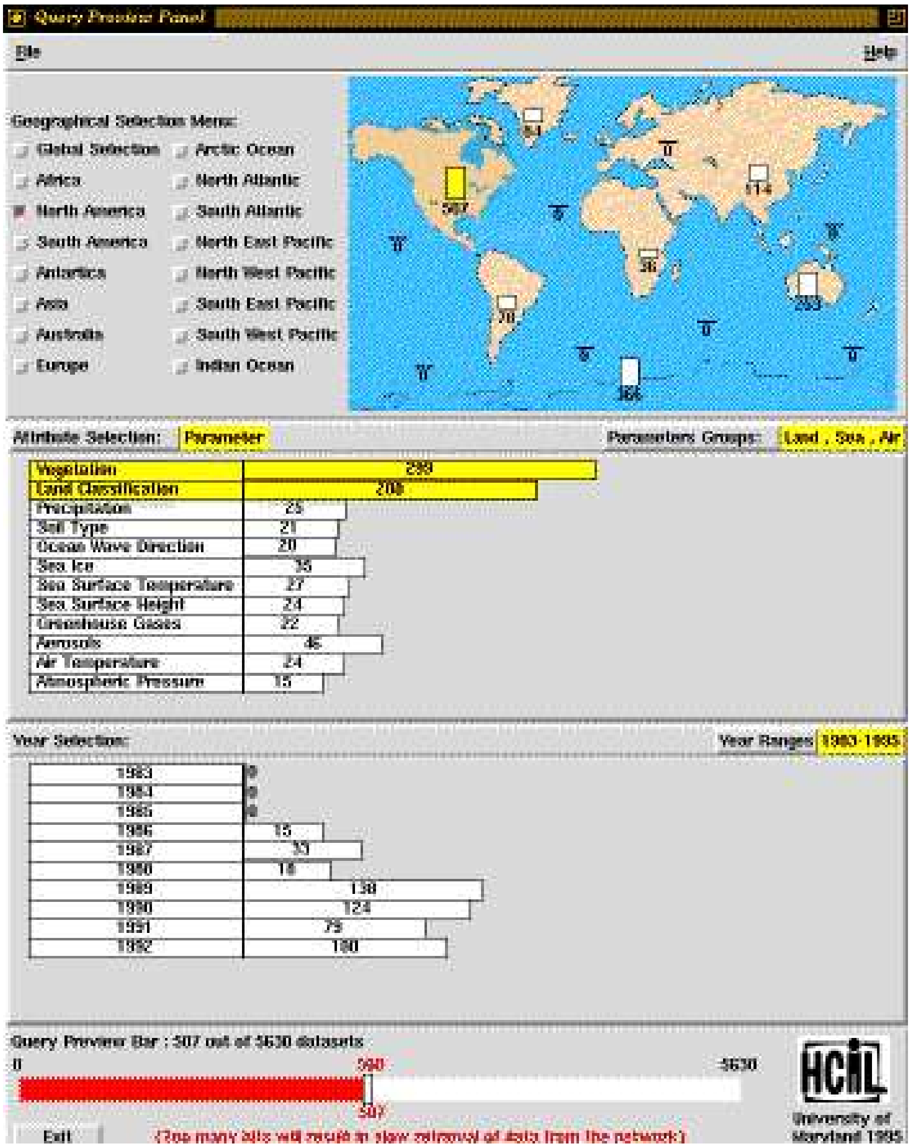


Figure 4.8: An example Query Preview Panel developed at the Human-Computer Interaction Laboratory at the University of Maryland. Year and location on earth are examples of discriminating attributes for many scientific data sets obtained from the NASA archives. The bars show the distribution of data [Tan01].

This internal representation is powerful with many of the data types. On the other hand, some of the data types can not be handled easily with the n-dimensional array approach. For example, the data-set used for the application in Figure 4.8 might be a multi-valued data set (i.e., any record in the data may cover more than one year, location, or parameter, e.g., a data-set in North America could be taken for both of the years, 1990 and 1991). A dramatic problem arises if ranges of values are used in a database (e.g., years 1983 to 1991). This will result in large numbers of duplications in the counts of the cube that will be used by the Query Preview Panel.

4.3 Related Systems

The first dynamic query visualizations emerged in the early 1990's. The first approach was developed by Williamson and Shneiderman in 1992 [WS92]. Their *Dynamic Home Finder* was designed to explore some real estate data-sets. Users were able to adjust attributes like 'number of bedrooms' or 'price' by simple widgets (see Figure 4.9). Later this idea evolved and triggered a product called *Spotfire* (see Figure 4.1 on page 62).

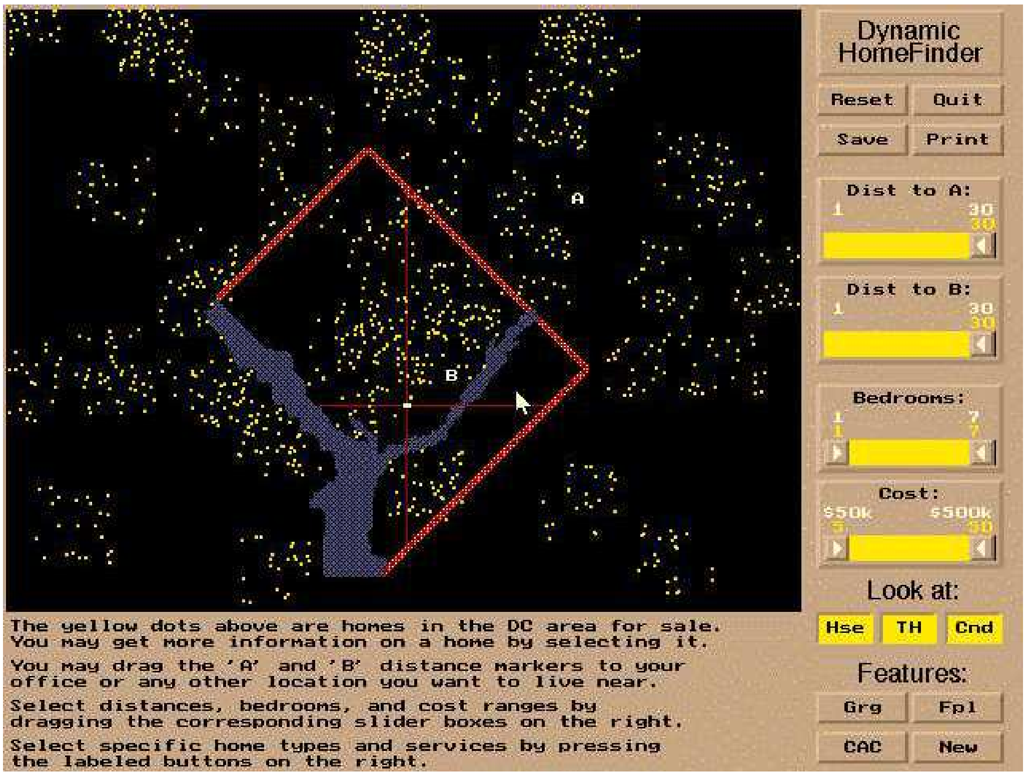


***Figure 4.9**: The first dynamic query example, Home Finder, using a real estate data-set from Washington, D.C. Users can adjust the widgets on the right to manipulate the six different dimensions of the data; updates are immediate on the map.*

Another example for a query preview system is the *'Restaurant Finder'*. In searching for a restaurant (see Figure 4.10 on the next page), the query preview screen gives users limited choices by means of buttons for the type of food (e.g. Chinese, French, Indian), double-boxed range sliders to specify average price of a main course and the times that the restaurant is open, and maybe a map to specify approximate regions. As users make selections among these attributes, the query preview bar at the bottom of the screen is updated immediately to indicate the number of items in the result-set. Users can quickly discover that there are no cheap French restaurants in downtown New York, or that there are many Caribbean restaurants open after midnight. When the result-set is too large, users can restrict their criteria and when the result-set is too small they can relax the constraints.

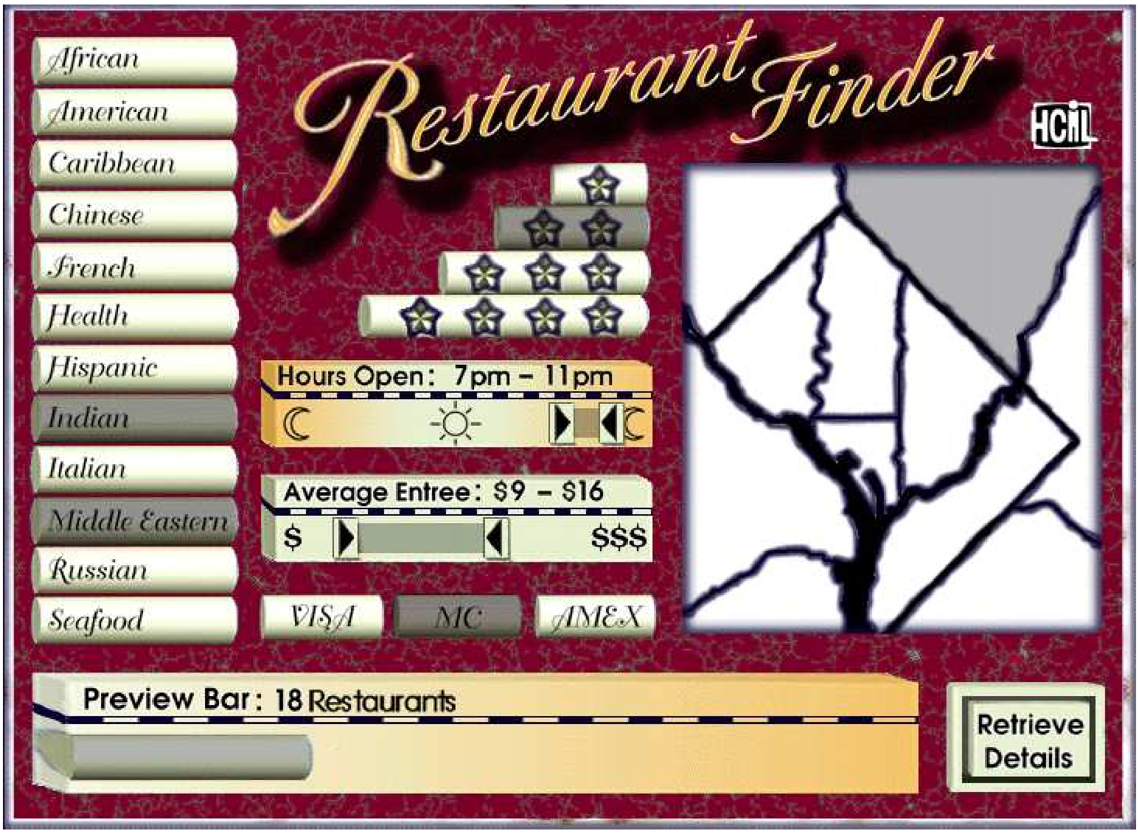


Figure 4.10: *Restaurant finder demonstrates the query preview idea. Users can quickly adjust the parameters and see the effect on the size of the preview bar at the bottom. Zero-hit or mega-hit results are immediately visible and users can always be sure that their search will provide an appropriate number of results (Graphic design by Teresa Cronell) [DPSB97].*

A similar approach was taken in the *FilmFinder* project [AS94c] (see Figure 4.11 on the facing page). This system tries to overcome search problems by applying dynamic queries, a starfield display and tight coupling among components. Dynamic queries were applied by having a double-box range selector to specify film length in minutes, by having buttons for ratings (G, PG, PG-13, R), large color-coded buttons for film categories (drama, action, comedy, etc.), and Alphasiders for film titles, actors, actresses and directors. The query result in the *FilmFinder* is continuously represented in a starfield display. The X-axis represents date and the Y-axis a measure of popularity. The system allows users to zoom into a particular part of the date-popularity space. As users zoom in, the colored spots representing films grow larger, giving the impression of flying in closer to the films. The labels on the axes are also automatically updated as zooming occurs. When fewer than 25 films occupy the screen, their titles are displayed [AS94c]. To obtain more information about a particular element of the query results, users click on that element, getting desired details on demand.

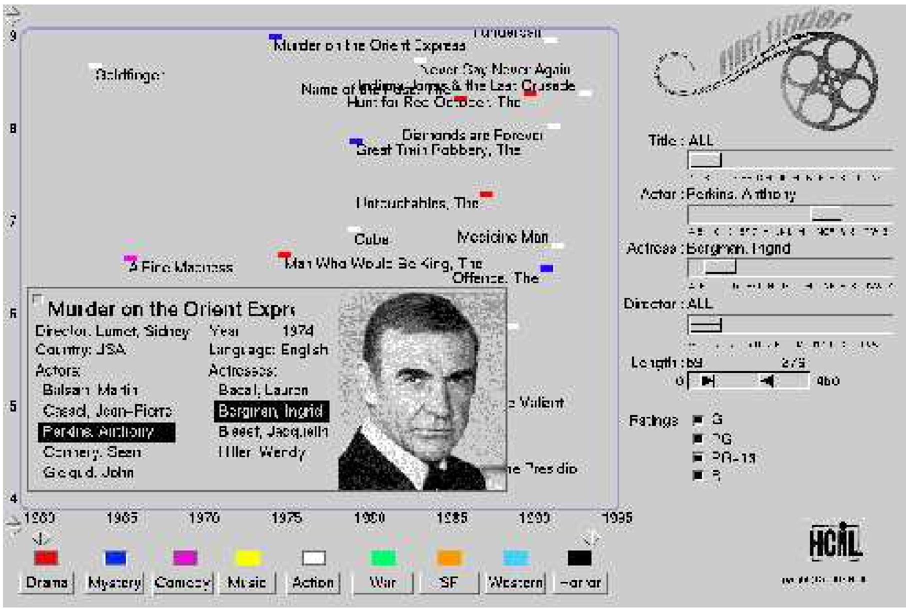


Figure 4.11: *The FilmFinder prototype presented at CHI'94 [AS94a]. Films are presented in a starfield (interactive scatterplot) with the year of production and the popularity of the films as axes. Users perform queries by manipulating query devices such as range-sliders, Alphasliders, and toggles.*

Another example is the query preview for a complex search on NASA environmental databases [PVN+99]. Users of the existing system had to understand the numerous and complex attributes of the database, which is distributed across eight archival centers. Many searches result in zero hits because users are uncertain about what data is available, and broad searches take many minutes while yielding huge and unwieldy result displays. The query preview uses only three parameters: dates (clustered into 20 one-year groups), locations (clustered into eight geographic regions), and 171 scientific parameters (cloud cover, ocean temperature, ozone, etc.) (Figure 4.12 on page 79). This comes to a total of 20 * 8 * 171 = 27360 data values in the table of contents. In the prototype, users can quickly discover that the archive held no ozone measurements in Antarctica before 1979, for example. Once a reasonable size of result-set is identified, users can download the details about these data-sets for the query refinement phase.

4.4 Summary

As humans we have the ability to recognize the spatial configuration of elements in a picture and notice the relationships between elements quickly [Kam98]. Information visualizations try to take advantage of this ability [CM99]. The foundation for powerful, interactive visual search systems is laid by various task or data-related visualization and interaction techniques. The most important ones for a dynamic query, filter or query preview techniques were shown. A major role belongs to the task of *Overview*. Not only should it provide a view on all documents but also provide some structure. This structure can come as clusters, hierarchies or categories.

Displaying the information space is one thing, working with it another. A visualization should enable the user to rapidly filter out irrelevant data. For this purpose the visualization should be tightly coupled with its controls (widgets) as well as with other visualizations (*'brushing and linking'*) when working as a filter in a *'Multiple Coordinated View'* (MCV).

The visual control of the query parameters that characterize dynamic queries should improve the user's understanding of the underlying data space. Furthermore, his/her knowledge on the subject should increase retrieval efficiency. The immediate and continuous feedback combined with an animated display (with the appropriate display refresh rate) prevents the user from becoming lost in the data space. Additionally, the fast reversal of any action increases the user's confidence in the system.

As an additional benefit of dynamic queries, the user should be able to quickly discover clusters (similar documents sharing a small space, e.g. starfield displays), exceptions, gaps (missing documents in a certain time interval), outliers, trends or patterns.

The next chapter will provide some details about the framework in which this work is embedded. Readers who are familiar with the VisMeB system may continue with the core chapter of this thesis on page 99.

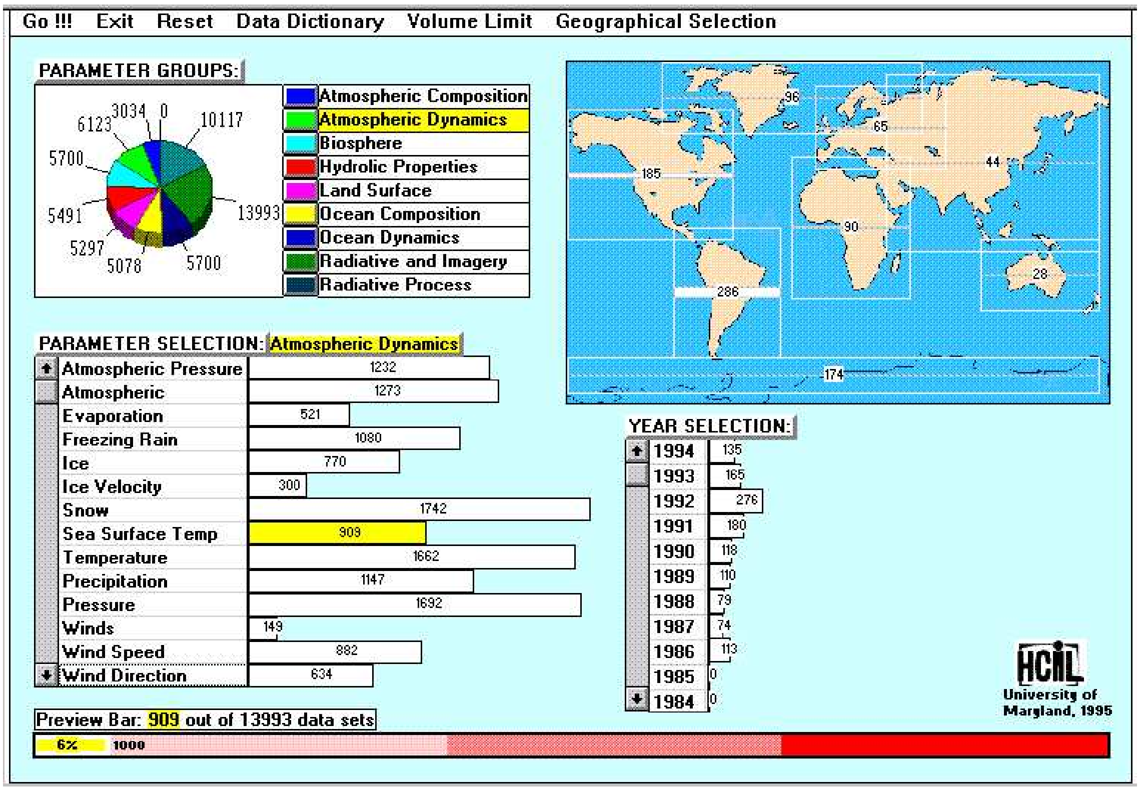


Figure 4.12: NASA query preview applies this technique to a complex search for professional scientists. The set of more than 20 parameters is distilled down to three, thus helping speed search and reduce wasted efforts. Users select values for the parameters and immediately see the size of the result bar on the bottom, thus avoiding zero-hit and mega-hit queries.

5 THE VISMEB FRAMEWORK

Contents

The previous chapters laid the foundation for the main chapter about the CircleSegmentView. But before presenting the main research, a few aspects of the framework in which this work is embedded are added.

As mentioned before, VisMeB was developed within the INVISIP project. Nevertheless we chose our approach to be as flexible as possible with regard to the application domain. As a result we are able to adapt rapidly to any (meta)database which provides a JDBC driver.

The main concepts and the first mockups were based upon a project called INSYDER[1] [Muß02], [Man02] - a web search engine with enhanced visual result-set

[1] The project was funded by the European Commission under the Fourth Framework of the ESPRIT Program, Project No. 29232. www.insyder.com

presentation and visual-query capabilities. A short review of the main results that initiated VisMeB can be read in section 5.1.

During the development of the VisMeB system I was mainly responsible for the retrieval parts as well as the development of effective and efficient query- and filter designs. The rest of the chapter should provide a rough overview of the design and concepts of VisMeB; for an in-depth discussion of the work see the thesis of Frank Müller [Mül05]. The sections that will follow cover the architecture of VisMeB (section 5.2 on the next page), the different visualizations and interaction techniques (sections 5.3 on page 85 to 5.5 on page 91) and a summary (section 5.6 on page 96).

5.1 The path to VisMeB

The first implementation of the INSYDER system includes five visualizations for the presentation of search results [MR99] [RMMH00] [RMM01]: a traditional list (mainly for evaluation purposes), a ResultTable, a ScatterPlot, a BarGraph, and a SegmentView with two modes: TileBars and StackedColumns (see Figure 5.1 on the next page). The primary intention for the use of different visualizations was to present additional information (meta-data) about the retrieved documents to the user in a way that is intuitive, may be quickly interpreted, and can scale to large document sets.

An extensive evaluation done with 40 users by Mann [Man02] has been focused on the different visualizations used to present the search results in the result phase of the search process. The primary goal of this summative evaluation was to determine the usability of the visualizations. A second goal was to detect problems with the visualizations used in the INSYDER system, and to collect suggestions for improvements. The usability evaluation was focused on the added value of the visualizations (ScatterPlot, BarGraph, TileBar, StackedColumn) in terms of their effectiveness (accuracy and completeness with which users achieve task goals), efficiency (the task time users took to achieve task goals), and subjective satisfaction (positive attitudes towards the use of the visualization) for reviewing Web search results.

The evaluation results indicated some difficulties in the area of user interaction with the system, e.g., more than 50% of the users voted for the ResultTable when asked with which visualization they performed best. Other visualizations were helpful as an addition to the ResultTable, but not as primary tools. When studying the expected value of the visualizations, it can be said that in the visualization (e.g. Scatterplot, BarGraph, SegmentView) plus ResultTable conditions where the user had the possibility of deciding which component to use, both components were used in the majority of cases. When analyzing usage times in these conditions, the ResultTable was the favorite component of the users. It was used in all three user interface test conditions with ScatterPlot, BarGraph, and SegmentView for more than 50% of the overall task time. Interpreting usage time as

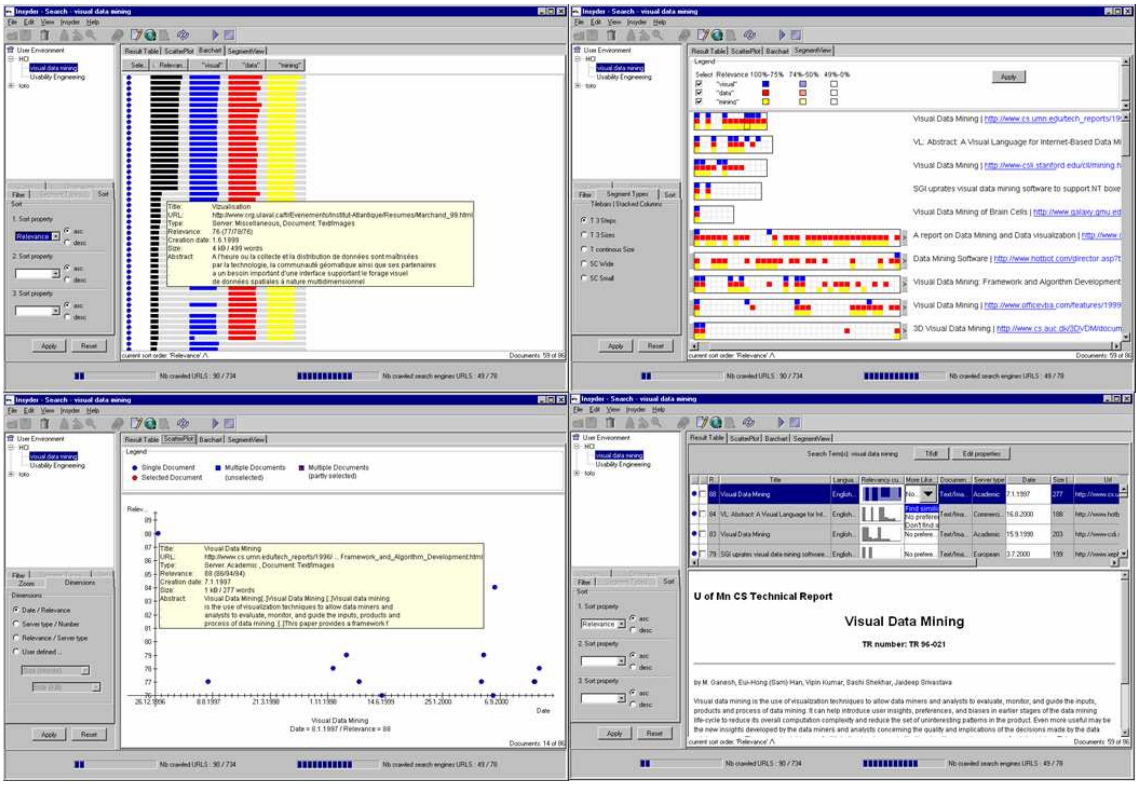


***Figure 5.1**: Various Visualizations used in Insyder: BarCharts, TileBars, ResultTable, ScatterPlot (clockwise from upper left).*

an indicator of expected value, the expected value of the ResultTable seemed to be higher than that of the other components for the users. Switching between completely different visualizations confused the users. Therefore, an attempt was made to combine / merge the regular table view with other views like the BarGraph or the SegmentView.

5.2 Architecture

VisMeB is based on Client/Server architecture [KMRL03] (see Figure 5.2 on the following page). A server stores and manages the different sessions, configurations and assignments of the visualizations. Thus it shares all the advantages of a classic terminal application (e.g. stop your work on your laptop and continue with the same session at your workstation in the office). With particular reference to the site planning process, which is probably done not only in an office but also partly at the specific location (e.g. with mobile devices), this aspect seemed important to us. The client will be available on different devices like workstations, TabletPC or PDA. On these platforms, we planned to support device-specific interaction techniques, like pen gestures or speech recognition. The VisMeB framework allows programmers, administrators and users to easily adopt the

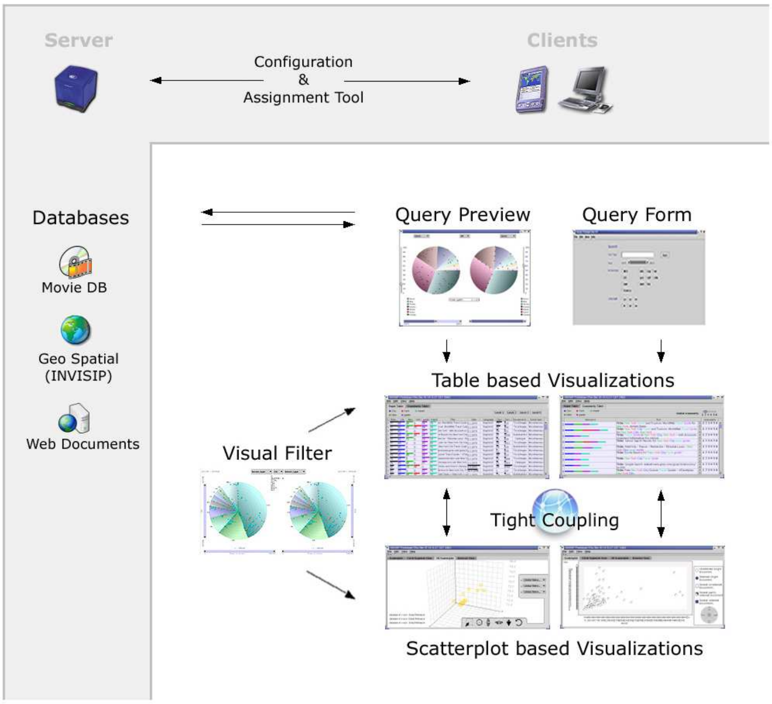


***Figure 5.2**: VisMeB's Architecture.*

visualizations' look & feel, select the appropriate database and change the assignments of metadata to different visualizations. This is achieved through a visual configuration and assignment tool (part of the meta-data toolbox, details in section 5.5.1 on page 92).

The client consists of a query preview or alternatively a form based query and the set of result visualizations. The result visualizations are either table-based or scatter plot-based (additionally, there is a browser view that displays various metadata-sets as a whole). The different views of the result-set are tightly coupled. The possibility of filtering the result-set is provided by a visual filter (or standard form-based ones). The filtering is tightly coupled to the result visualizations. A typical use scenario of VisMeB from the user's point of view starts with the information need. Let us assume that all of the data in which a user is interested are stored on a database server. The original data are characterized by metadata attributes like title, author, date based on the Dublin Core metadata standard for Web documents (see dublincore.org), or on the ISO 19115 standard for geographic information/metadata. To start a search the user can use a query form (the look & feel depends on the application domain and can be manipulated with the visual configuration and assignment tool) or the query preview functionality to restrict the data result-set, which will be analyzed to an agreeable size. The SuperTable or the ScatterPlot in combination with the BrowserView, however requested, will be used to explore the result-set. In this stage, the user can use the functionality of a visual filter to further focus on the result-set.

5.3 SuperTable

Based on the empirical findings of the evaluation mentioned in section 5.1 on page 82, we have decided to integrate the ResultTable, BarGraph, and SegmentView into one visualization called 'SuperTable' and to improve the ScatterPlot.

In the user test [Man02] the users requested a number of features for the BarGraph and the SegmentView already implemented in the ResultTable. All of this could also be implemented in the ResultTable. Therefore, the proposed SuperTable integrates the concept of a distortion-based table, the BarGraph and the SegmentView (with TileBars and StackedColumns) in a way that allows easy manipulation of the table.

The redesign of the INSYDER visualizations combines the SuperTable + ScatterPlot into one single window offering different brushing techniques between them. Therefore the ScatterPlot will supplement the SuperTable by giving the user a quick overview of all search results, and offering the user a variety of controls (e.g. defining one's own views, zooming, selecting, filtering) to reduce the number of hits to a smaller group of interesting documents. These documents can then be selected by the user and analyzed in more detail in the SuperTable.

The enhanced ScatterPlot with additional lens mechanisms (e.g., a magic lens for filtering operations [FS95]) and the radial *'Multiple Data Point'* (MDP) visualization tightly coupled with the SuperTable and a document browser (showing the detailed document with keyword highlighting) were our main redesign ideas.

Thus, two visualizations dominate the global appearance of the result presentation: The SuperTable and the ScatterPlot. They unify the typically used result list with unique combinations of visualizations. The user has the opportunity of obtaining a quick overview of the result-set as a whole and then explores relevant objects step by step. Through brushing and linking we can achieve synchronized visualizations. The SuperTable itself consists of a combination of different visualizations. BarCharts, TileBars, and highlighted texts are examples of such visualizations.

The underlying idea is a granularity concept, which enables users to change the depth of information in which they are interested. For the redesign three decisions were made:

- The visualizations are to be integrated in a way that clearly indicates the nature of the visualizations as several views on the same data. The visualizations should range from a very low detail version, which allows the display of a large data-set, to a very high detail version, which allows thorough examination of single data. We called this granularity concept. Granularity is a term used in photography to describe the accuracy of pictorial representations on film. The higher the granularity, the more detail can be seen on a picture. Exposures of high granularity films can be very large and still show a smooth and detailed picture. Exposures of low granularity films appear grainy even of small sizes.

- The result-set is aligned in a tabular form where each row presents visual and textual information for one single element. Users have the possibility of changing the degree of detail for the whole table or for single rows. If they choose to change a single row, the table becomes distorted, embedding the focused row into the context of the unchanged rows. Changing several non-neighbouring rows even leads to multiple distorted views.

- Even if the user accepts the concept of the different visualizations being only different views of the same data, one major problem remains: how to interact with these different views? How can the user switch from one view to another? There are many possibilities that can be thought of in order to solve this problem, but no single possibility is obviously the best one. We therefore decided to implement two different design variants of the SuperTable (LevelTable and GranularityTable) and await the user tests before deciding.

5.3.1 LevelTable

The first of the design variants is called LevelTable (see Figure 5.3 on the next page). Because the change from one level of detail to another is only possible for the whole result-set shown in the table, we called it LevelTable.

The first level offers an overview of all documents; the last level shows the document itself. In between you find different levels increasing their amount of information from the first to the last level. This drill-down functionality is named 'Focus of Interest'.

On Level 2 (Figure 5.4 on the facing page), more information will be visible in the form of text completing the visual representation of the multicolored bars. Now numeric values add detailed information about the bar displays from the initial, graphics-only display. Other meta-data are now readable, but only up to the width of the respective column. All wider texts become truncated.

Every meta-data has its own column, but not all meta-data are visible in every level. One special example is the 'Relevance Curve' (Figure 5.5 on the next page, third column from the left, devised as a curve). It represents the document as a whole, whereas the horizontal extent of the visualization reflects the document's length. Important text passages are marked by vertical deflection (level 3) or color-coded bars (level 4, see Figure 5.6 on page 88) whose height illustrates the factor of importance. This meta-data can only be seen in levels 3 and 4. In return, some columns (which were visible in the former levels) disappear in this stage. In the LevelTable, buttons are used to change levels on the whole

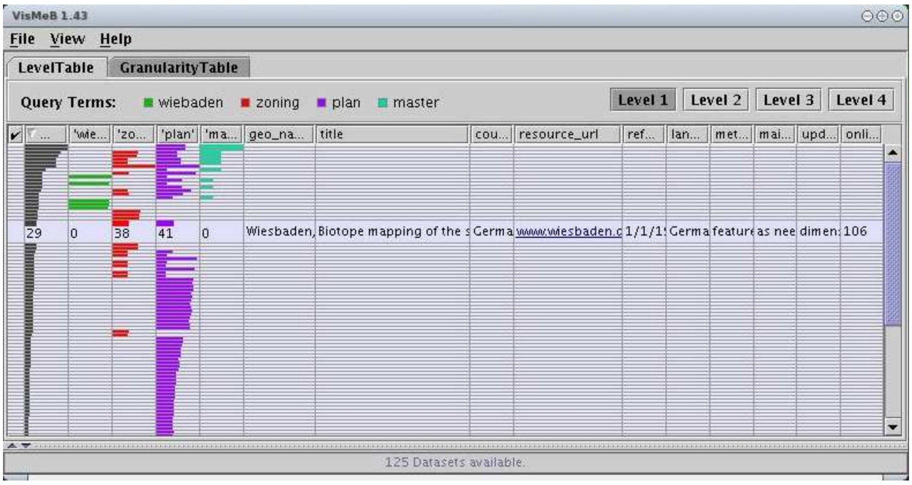


***Figure 5.3**: LevelTable in 'Level 1' with BarCharts*

document corpus (see Figure 5.3 in the upper right corner). Pressing a button moves the documents in a body to the corresponding level.

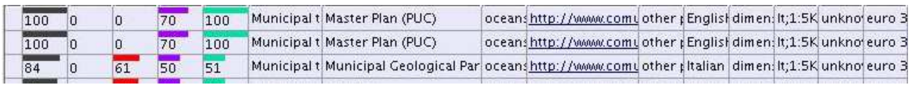


***Figure 5.4**: LevelTable in 'Level 2' with BarCharts that symbolize the overall relevance of the document (the black BarChart on the left) and the relevance by query term (color coded).*

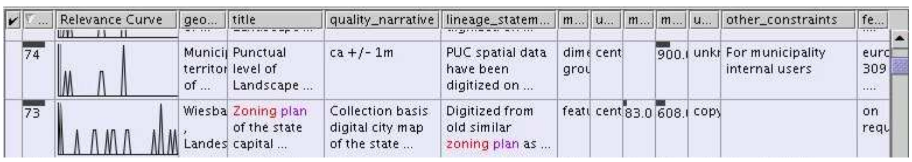


***Figure 5.5**: LevelTable in 'Level 3' with RelevanceCurve.*

5.3.2 GranularityTable

The second design variant, named GranularityTable (see Figure 5.7 on page 89), differs slightly from the LevelTable. Instead of buttons for changing level, sliders are used to change from one level to another. The number of levels differs (now you can choose

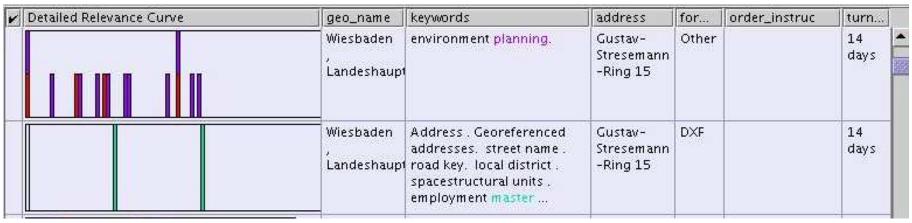


Figure 5.6: *LevelTable in 'Level 4' with StackedColumns as a detailed RelevanceCurve.*

between six). Only four columns are used to show all the information: selection, visualization, text, and granularity. The visualization as well as the text columns change their display from level to level, always giving more information than the previous level. In this version it is possible to move single rows to another stage, and not just the documents as a body.

5.4 ScatterPlots

The ScatterPlot (see Figure 5.8 on page 90) is a two-dimensional coordinate system enhanced by the possibility of assigning every kind of meta-data used in the current context to the x- and y-axis. It simplifies a comparison of document properties, for example document date, size or relevance. Using different colors for the data points adds another dimension that allows a faster perception of important facts. A standard technique for providing additional information about visualizations is the use of tool tips. Moving the mouse over an object shows the characterizing properties. All available information can be retrieved without using the table.

Using a combination of SuperTable+ScatterPlot enables the user to obtain a general idea of the entire result-set as well as the possibility of exploring interesting documents in detail. To reach this goal, both visualizations are synchronized using brushing and linking. For example, selecting objects in the ScatterPlot leads to the highlighting of only the corresponding objects in the SuperTable, or marking rows in the SuperTable marks the respective data points in the ScatterPlot.

The Magic Lens Filter, influenced by Fishkin's movable filters [FS95], which is available in the ScatterPlot, affects the SuperTable as well. Moreover, it is possible to use different lenses simultaneously, which makes it necessary to add half-transparent lens colors. If the lens filters out objects, the background of the corresponding objects in the table changes to the lens color (see Figure 5.8 on page 90). We decided to use this technique to achieve the interaction because a permanent movement in the table, caused by removing objects, would obviously confuse the user. Moreover, the possibility of

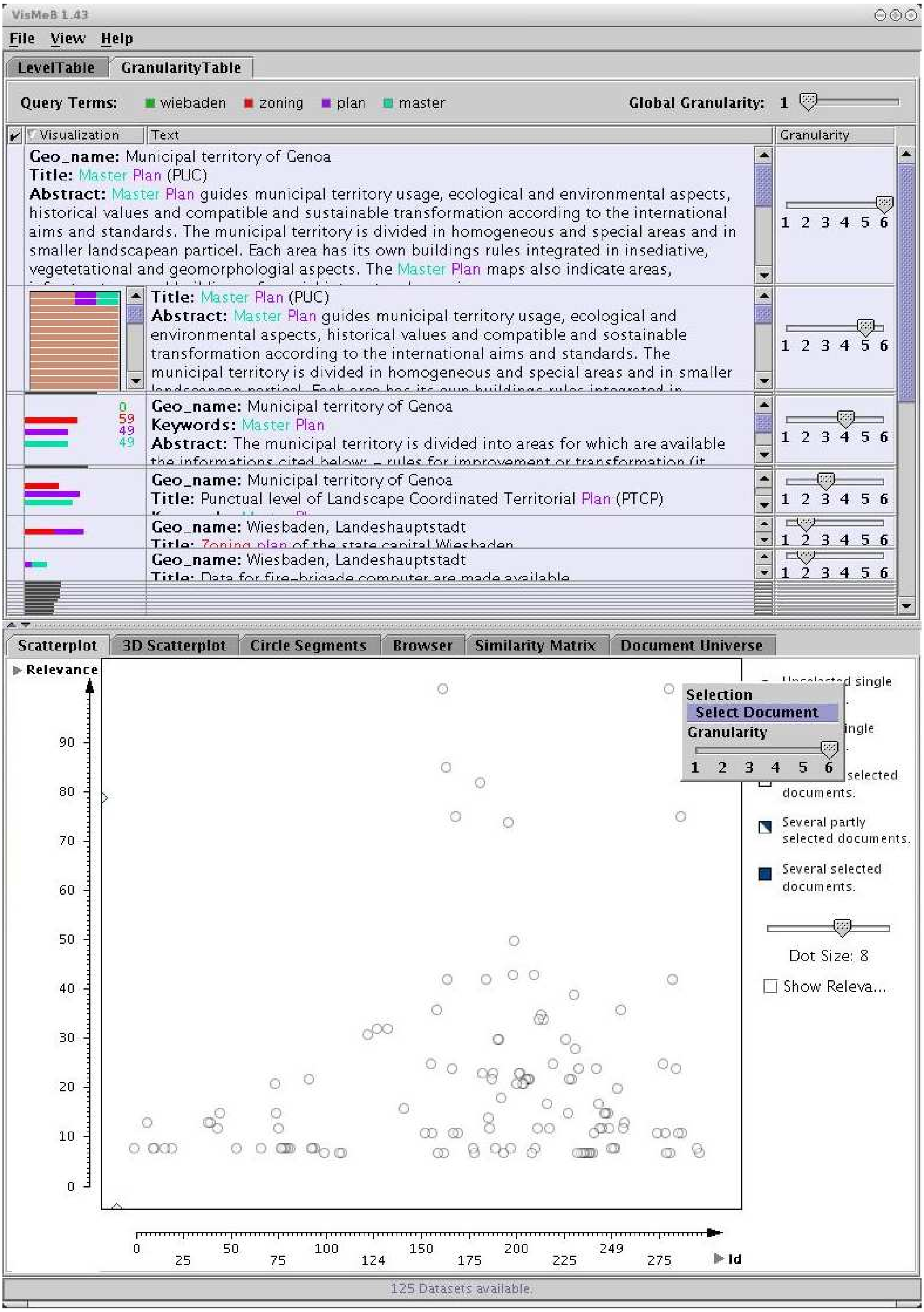


Figure 5.7: GranularityTable showing various documents in different granularity levels (1 to 6). The interaction with the ScatterPlot can also be seen.

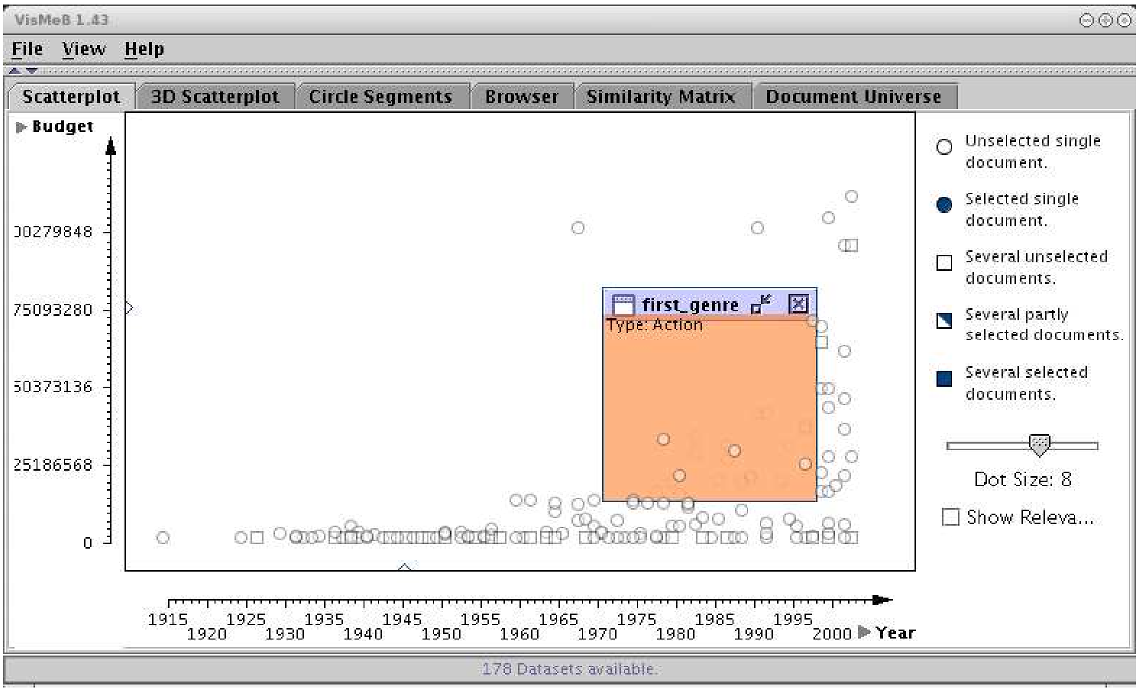


***Figure 5.8**: The 2D ScatterPlot used in VisMeB. A Magic-Lens-Filter is used in the ScatterPlot, the corresponding objects in the SuperTable are highlighted with the appropriate colors.*

exploring the filtered documents would be taken away.

As well as the 2D-ScatterPlot, a 3D-ScatterPlot was created (see Figure 5.9 on the next page). Here, data points are visualized as 3-dimensional cubes. Using a light grid in the background for limitation and better orientation emphasizes the 3D effect. Labels are set to the grids edge to achieve better clarity. Free rotation providing an illumination from all directions, a zoom function, and different selection mechanisms complete the equipment of the 3D-ScatterPlot.

Special attention was directed to the problem of data-point overlapping in the ScatterPlot. Objects frequently own the same meta-data for specific characteristics leading to the same position in the drawing area. We therefore introduced a new glyph, known as the Multi Data Point or MDP to point this fact out to the user. Our solution approach is based upon the RSVP (Rapid Serial Visualization Presentation) idea by Spence [Spe01]. We use a rotating display of meta-data index cards (see Figure 5.10 on the facing page).

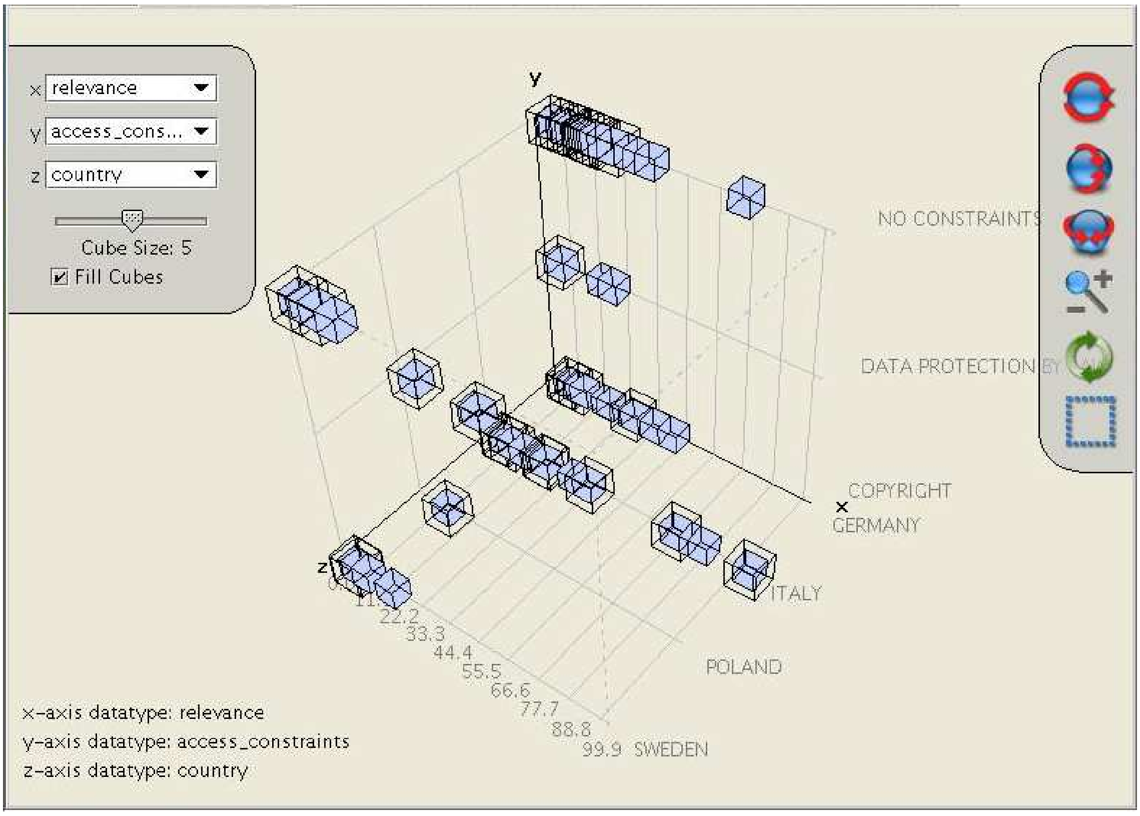


Figure 5.9: *The 3D ScatterPlot used in VisMeB.*

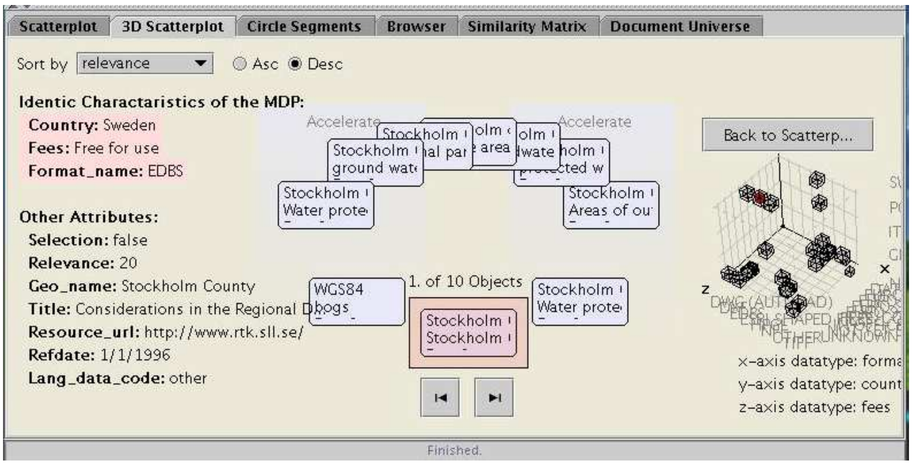


Figure 5.10: *Multi-data-dots solved as a RSVP (Rapid Serial Visual Presentation).*

5.5 Additional Visualizations

Despite the system's philosophy of SuperTable + ScatterPlot, some more visualizations evolved. This is due to the fact that as a research prototype, VisMeB attracts scientists and students, whose research contributes to the system. Some of the more powerful concepts are presented in this section.

5.5.1 Meta-data Configuration and Assignment Tool

To adapt VisMeB to an application domain it is necessary to connect to a database, explore the tables and views, inspect the data types used and finally, after collecting all this information, to configure an exclusive visualization of VisMeB. This makes it impossible for a broad user spectrum (with or without programming experience) to fundamentally alter VisMeB for their specific needs. Our visual configuration and assignment tool takes care of that topic. It is able to connect to a variety of different database systems (DB2, Oracle, Postgres, mySQL just to name a few) and to visualize their table structures.

An expressive visualization depends on the transformed raw data to encode all the data relations intended and no other data relation. This transformation step usually takes place at the DBMS level. This process is usually very intellectually challenging and is normally done by individuals rather than by software. Simple counting and calculating of distribution are preferred for software-generated meta-data as well as for calculating relevances or rankings. The next step is to map these generated meta-data to visualizations. Usually it helps if the data is collected in a separate view instead of offering the user many database tables.

At this stage VisMeB's configuration and assignment tool (as part of the meta-data toolbox) offers the possibility of mapping meta-data to visualizations (see figure 5.11 on the next page). In the first step, the general information about the database and its connection is requested. After accessing the database we are presented with a split view: a frame on the left displays the tables in a tree hierarchy. In the right frame the visualizations are also rendered in a tree. The root on the right is a label called 'Visualizations'; the first children include the main visualizations like 'LevelTable' or 'Scatter-plot', but also a node for the 'Initial Values'. The last level contains the user interface widgets; for example, the tables have their 'columns', the scatter-plots have their 'axes' and the CSV has the allocation for the combo boxes and for the two alpha sliders. The user can now connect the database tables to the visualizations by simply dragging a column to the desired node on the right. These mappings are restricted to avoid critical mappings that a visualization could not handle. For example it is not possible to connect a text array to a CSV combo-box (nominal data would not make any sense in this case). Additionally, it is possible to select meta-data that occur in the form fill-in filter menu or in the form fill-in query interface. The user can also mark table columns that should be included for the keyword ranking mechanism[2].

[2] A simple tf*idf algorithm

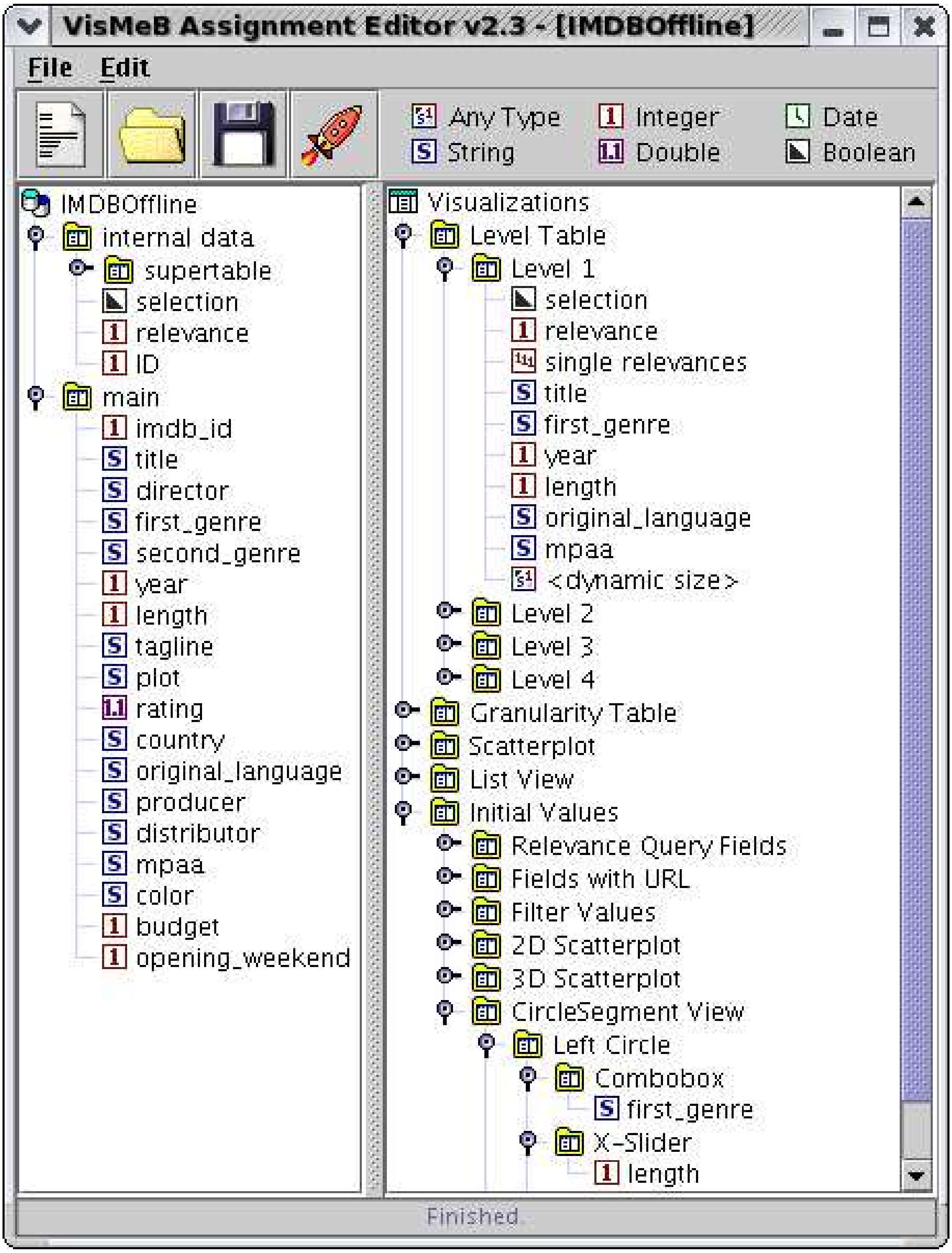


***Figure 5.11**: The visual configuration and assignment tool.*

After all assignments are finished, one can now save this configuration and test the settings. Preferably this is done with the involvement of the target group. Leaving the usability engineering life-cycle at this point would be critical. The final product is stored and can be shared with other members/work groups.

The benefits this tool offers are: The time spent on personalizing the system is dramatically reduced. It is much easier to construct two or more competing visualizations and then (for example) test them in a usability lab. Users such as information brokers can instantly share improved views on their databases. For system administrators it is much easier to access new databases. The only adaptation to a template database-class is the kind of database (e.g. Oracle on host data.base.com), the appropriate ODBC or JDBC driver, the user-name and the password. The assignments can be saved as plain ASCII text files. This makes it much easier to adjust settings during the development or the debugging phase.

5.5.2 Filter

The possibility of filtering the result-set is one main advantage of interactive systems. Visualizations can support this process by providing appropriate interaction techniques to fulfill the task. In our implementation it is possible to filter the data-set by restricting every kind of meta-data. Depending on its character (e.g. nominal or ordinal) the interaction widgets to be used vary slightly. Categories (e.g. languages) can be filtered in or out by selecting a check-box whereas an interval (e.g. from 0 to 100) will be adjusted by a two-sided slider, also known as the Alphaslider [AS94a]. The use of a map to let the user directly choose the domain attributes (e.g. .de, .us, or .co.uk) is due to the application area of geographical information systems.

Filters can easily be activated by a check-box connected to the responding meta-data, which eases the later use of a filter that was set some steps before. Additionally it is possible to activate all filters or invert the current selection to enhance the usability of the selection activity. This kind of filter has an affect on all the visualizations i.e. on the complete set of data.

5.5.3 Document Universe

The DocumentUniverse was an attempt to provide a clustered view of the information space. It is a type of starfield display. The placements of the data dots is computed through a modified LSA (latent semantic analysis) algorithm [Rex05]. As an alternative to the Scatterplot, the DocumentUniverse behaves very similarly. Focus and selection in the table or the Universe highlights the corresponding data point in the coupled view. Using the context menu in the same way as described above to change the levels of detail for elements in the table is planned, but not yet implemented. A zoom function enables the user to get more details for a specific area. Additionally, panning is possible and

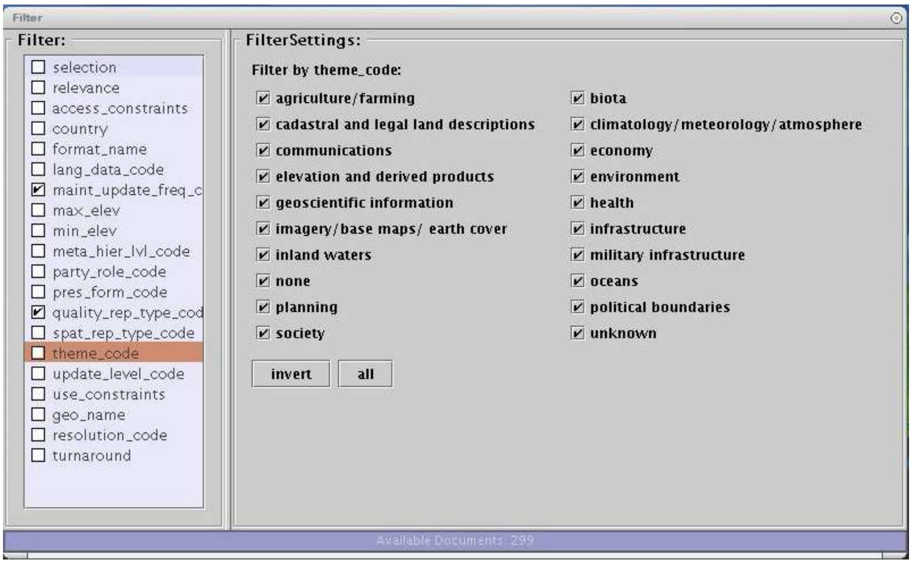


Figure 5.12*: Standard Filter Dialog in VisMeB.*

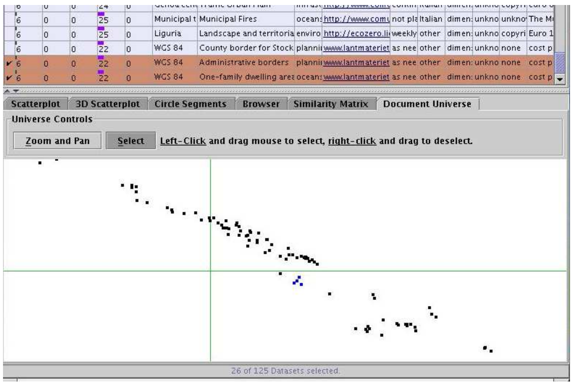


Figure 5.13*: Document Universe clusters the available documents according to a LSA-variant.*

helps to move around the complete data space.

Evaluations of this visualization indicated exceptional performance in matters of similarity exploration. The biggest deficiency is the lack of additional features.

5.5.4 BrowserView

The BrowserView (see Figure 5.14 on the following page) is used to display large text passages. It is automatically activated and replaces the ScatterPlot in level three and four in the LevelTable. In the different granularity implementations, the necessity of using this visualization varies. If there is enough space to show all the required data in another visualization, e.g. in the SuperTable, this view can be neglected. The BrowserView provides another interaction, based on 'Navigate - Navigate': The different RelevanceCurves can act as a navigational slider when used in conjunction with the BrowserView. Moving the mouse over segments in the RelevanceCurve highlights the corresponding segments in the BrowserView.

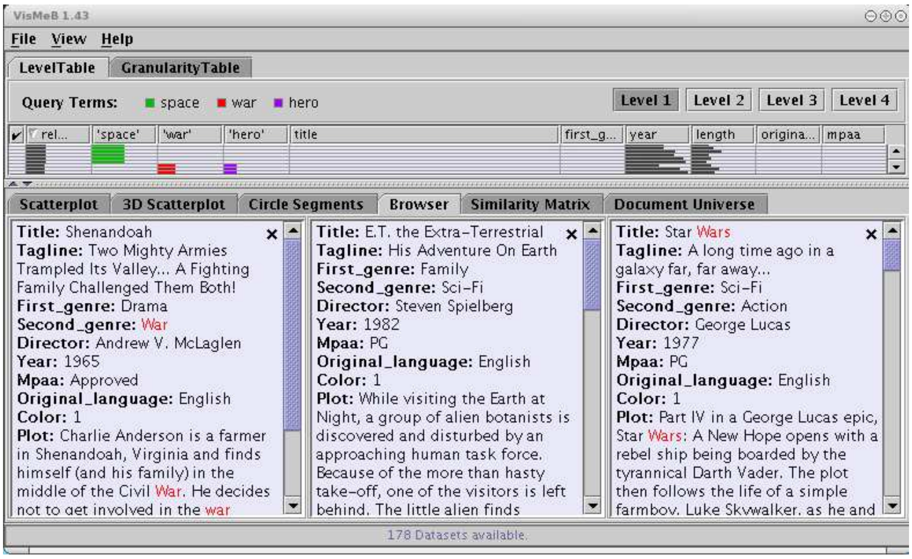


***Figure 5.14**: BrowserView offers the possibility of comparing whole meta-data-sets.*

It is possible to load more than one meta-data-set to the BrowserView. In this mode it is possible for the user to compare one or more documents at once. The layout can be changed by the user: instead of columns one can chose rows or matrix layouts.

5.6 Summary

The advantage of the original INSYDER system was the variety of visualizations used to support users in their web search. Our redesign combines these visualizations with the widely adopted spread sheet-like layout of the SuperTable. So new possibilities are offered for finding the most appropriate document for the current task in an environment to which users are accustomed. Both design approaches presented in this chapter give us the possibility to improve the advantages of this combination in a more detailed way.

Additionally a highly sophisticated data model will enable us to adapt VisMeB to a wide range of meta-data rich fields like stock market, medical data mining or geographic information systems.

Especially in the domain of information appliances the 'traditional' result list approach seems cognitively demanding and unsatisfactory. The granular SuperTable seems promising enough to stimulate some serious thinking about adapting to a wide variety of different use scenarios. Some interesting ideas about various granularity concepts (table, row, cell-granularity) can be investigated in [Grü04]. The granularity concept, together with the Multiple Coordinated Views approach, offers powerful interaction possibilities to the user.

The next chapter will address the remaining visualization and examine efficient filtering and query preview in the VisMeB framework.

6 THE CIRCLESEGMENTVIEW

Contents

In the previous chapters we saw that, to build a VISS that supports human search behavior, it is necessary to:

1. Use meta-data that describe the raw data as stated in section 3.1 on page 32. Visualizations usually visualize only the 'meta'-part of the information (data); e.g. the placement of a data dot in a scatter-plot depends on two (three if it is a 3D scatter-plot) meta-data values (usually numeric ones). Thus the user can only act on the data knowing that data D has value $D_X = x$ and $D_Y = y$.

2. Follow a user-centered design process. Ideal case: synchronized usability and meta-data lifecycles (section 3.3 on page 38).

The time has now come to concentrate on the user interface. In the field of Information Visualization and Visual Data Mining much effort has been spent on inventing new visualizations that utilize human visual capabilities [Spe01]. Functions of Information Visualization and Retrieval Systems include browsing, searching, refinement and presentation as mentioned in section 2 on page 9. However, increasingly users demand more features and tools to help them to mine data interactively, to generate patterns and to conduct analysis on data [Mos03].

The meta-data generation and analysis is already concluded (as described in the information visualization reference model [CM99]) and the focus now rests on the mapping of the meta-data to the visualization. For this meta-data driven, user-centered approach the following design focus was adopted (this scenario was part of the INVISP project):

1. Build a visualization for expert users that is capable of mapping a lot of different meta-data and supports the user's way of searching. The expert user is defined as a frequent user, trained on the interface and who performs information requests on a regular basis (such as, for example an information broker).
2. The visualization can compete with traditional form-based queries.
3. The visualization can compete with other visualizations such as ScatterPlots as a visual filter and supports data-mining qualities (pattern recognition).
4. The visualization should be aesthetically pleasing.

6.1 Development

The idea of the CircleSegmentView arose from the idea that pie charts (the first pie chart is thought to have been drawn by William Playfair in 1801) could offer optimal visualizations of categories [DDM86]. Pie charts are a well-established visualization in the field of business graphics [Mey96]. They should be easily understandable or easy to learn [Tho33][Fei03]. Pie charts, in their simplest form, are circles subdivided into differently-coloured regions. The greater the segments area, the greater the categories' value. Pie charts are typically used to summarize categorical data or, even more often, percentile data. The components have to add up to make a 'whole' of sorts or else the chart becomes meaningless (e.g. student population, market segment, etc...). A segment may be seperated from the rest of the pie to indicate its significance. Typical advantages of pie charts are:

- Pie charts provide an excellent visual concept of a whole.
- Clear comparison of different components.
- Highlighting of information by visual separation of a segment.

Commonly, pie charts are associated with the following disadvantages:

- Comparing pie charts is very difficult, as pie charts indicate components' sizes relative to each other, not to some absolute value.

- Too many segments are difficult to read and hard to label; better to use a bar graph.
- Difficult to understand without labels (especially with similarly sized segments).
- It is hard to illustrate error values.

A pie chart visualization with some more features to visualize so-called *'use packages'* was developed in our working group [Pre00] - see Figure 6.1. A focused object is placed in the center and all the other objects are placed according to their relation to the center-object. The segments of the pie characterise certain semantic relations. The distance to the center is an indicator for the release date or the relevance (when available). The interaction is limited to changing the center-object and change of dimensions.

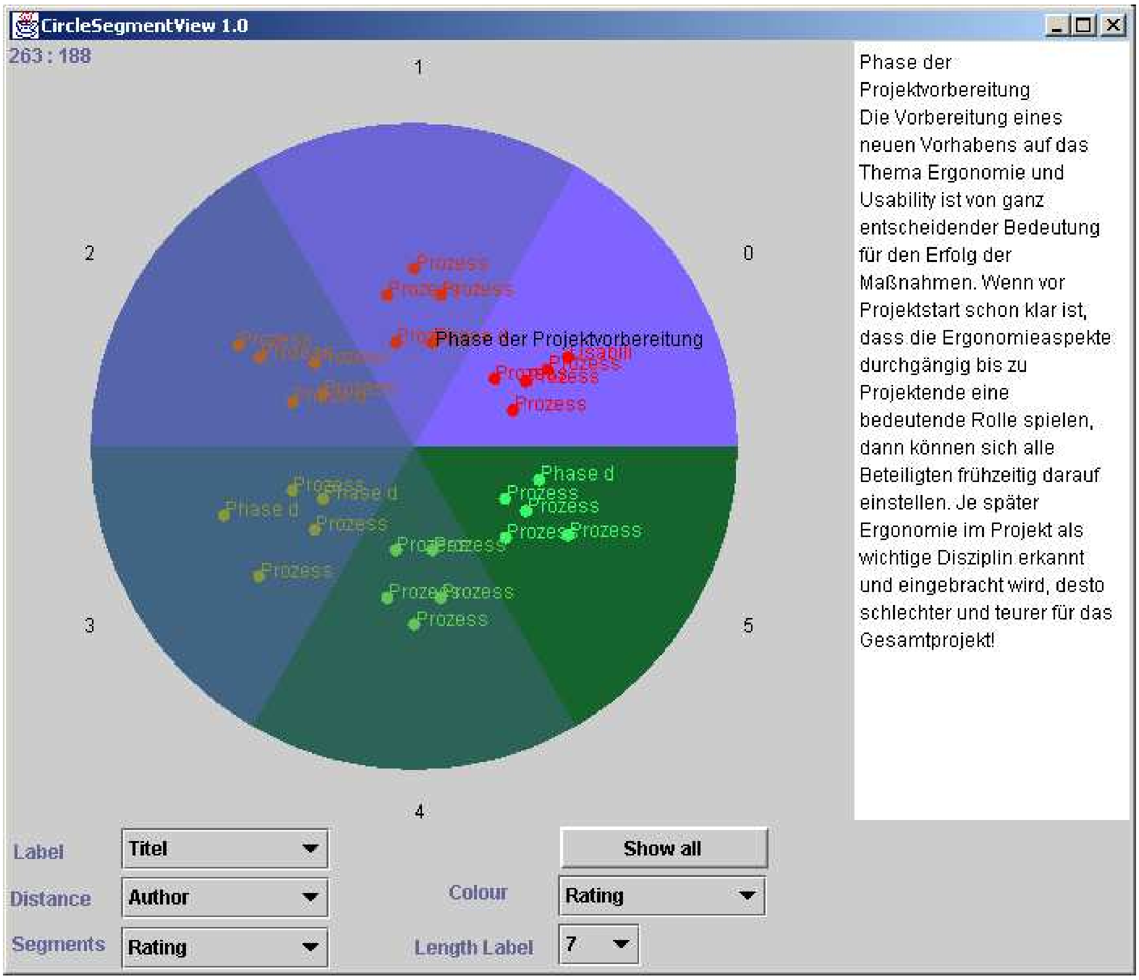


Figure 6.1*: PieChart visualization as part of [Pre00].*

The work of [Pre00] integrated the pie chart concept with scatterplot / scatterfield characteristics. Merging scatter-plot elements (displaying single data or document points) with the strengths of pie chart visualization seemed like a promising attempt to advance from this basic idea to a powerful *'overview'*. A first low-fidelity mockup (one of the first computer drawn sketches can be seen in Figure 6.2 on the next page and 6.3 on page 104) was created. These *'lofi mockups'* were subjected to several heuristic evaluations inside and outside the HCI Lab. Based on the insights gained from these discussions, a first JAVA prototype evolved.

This application was designed after the MVC principle (Model-View-Controller [GHJV93]) - see Figure 6.4 on page 105 for a detailed UML diagram. All datastructures were inherited from the accompanying SuperTable + ScatterPlot Framework that was later renamed to *VisMeB*. The API was designed to handle meta-data exchange with other visualizations that were being worked on in parallel.

The CircleSegmentView was on a constant iterative development program as was the whole VisMeB (see section 3.3 on page 38 for the choosen procedure). Testing and evaluation was done using different methods. Firstly there is the *'cognitive walkthrough'*.

> Cognitive walkthroughs involve simulating a user's problem-solving process at each step in the human-computer dialog, checking to see if the user's goals and memory for actions can be assumed to lead to the next action. [NM94]

This is usually a method followed by the programmer or developer together with a usability expert. The evaluators walk through the action sequences for each task, placing it within the context of a typical scenario, and meanwhile they try to answer the following questions [RC02]:

- Will the correct action be sufficiently evident to the user?
- Will the user notice that the correct action is available?
- Will the user associate and interpret the response from the action correctly?

This method clearly benefits the situation where the programmer and usability experts work in the same group. Another method that was applied during the development is the *'interview'* along with *'focus groups'* and *'workshops'*.

This usually happened when members from the INVISIP project came to visit[1] or when we met at other members' locations. Because we were responsible for accomplishing of the feasability and usability studies, most of the tests took place in Konstanz. Other opportunities arose, when researchers and experts in the field of information visualization came to visit (thanks to Anselm Spoerri, Maximilian Eibl, Alfred Kobsa,

[1] Project members came from: Sweden. Poland, Italy and two other locations in Germany: Darmstadt and Rostock.

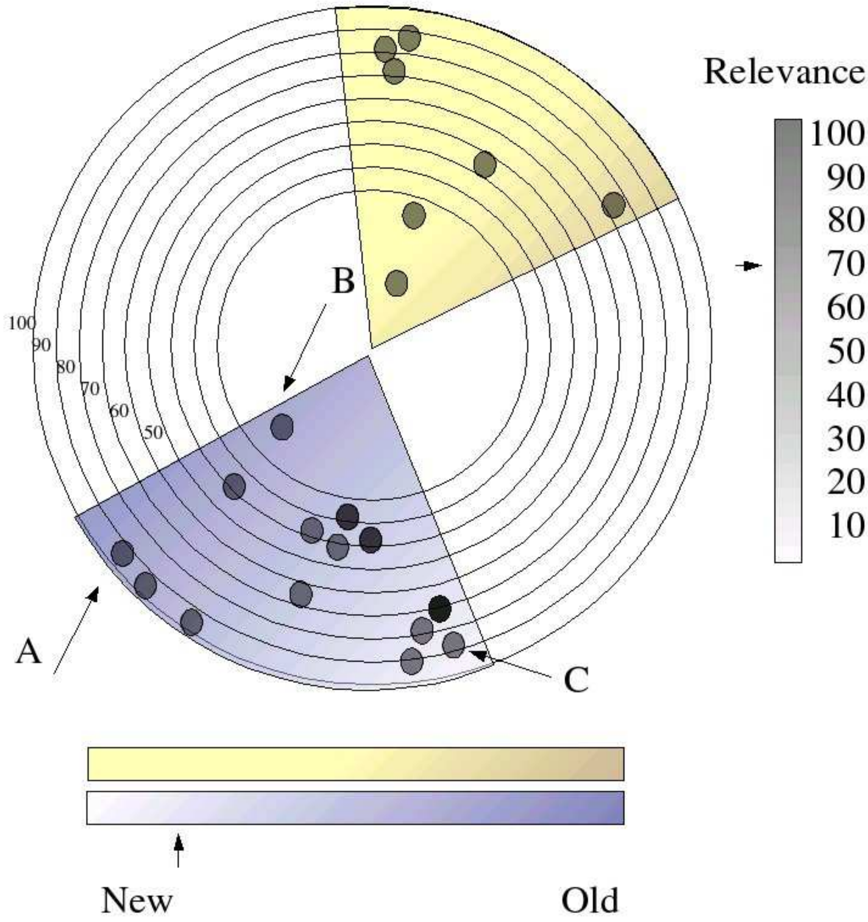


***Figure 6.2**: Early mockup for the placement idea.*

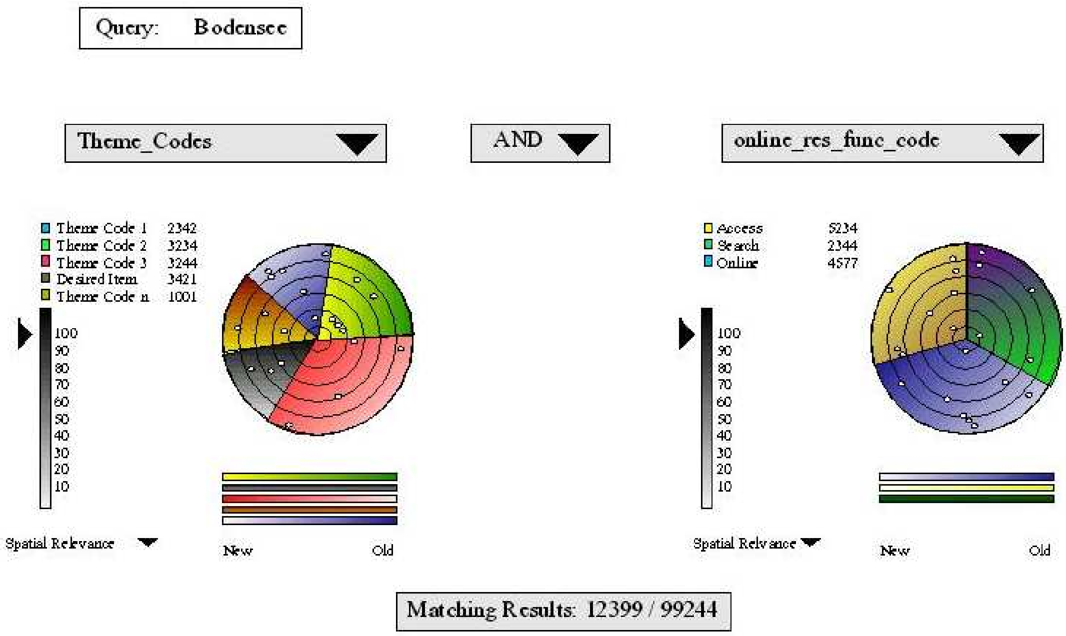


***Figure 6.3**: Early mockup for the User Interface.*

Philip J. Langley...), or met at conferences. All these encounters were usually a rich source of information on flaws that had been detected, and hints on how to improve the design.

The provisional end of these evaluations was a usability test with 20 persons. The result of this study can be read in section 6.5 on page 120. The evaluation continues even now as a new cooperation with the Fachhochschule Konstanz (School of Design) opened the possibility that design experts would analyze the CircleSegmentView. Currently these studies are completed and their findings are included in the conclusion and outlook section on page 127.

6.2 Design

The CSV was planned to introduce the user to the visual information-seeking process. It should offer an overview of the information space and also some general features of the collection before the user can specify a query. Besides this information, the visualization should make it possible to limit and filter the result-set to a manageable size. The evaluation results [LKMR03][KMLR03c][KMLR03a] on search behavior for the INVISIP domain showed a clear tendency towards filtering out irrelevant data as early as possible, based on some meta-data attributes. These findings can be backed up

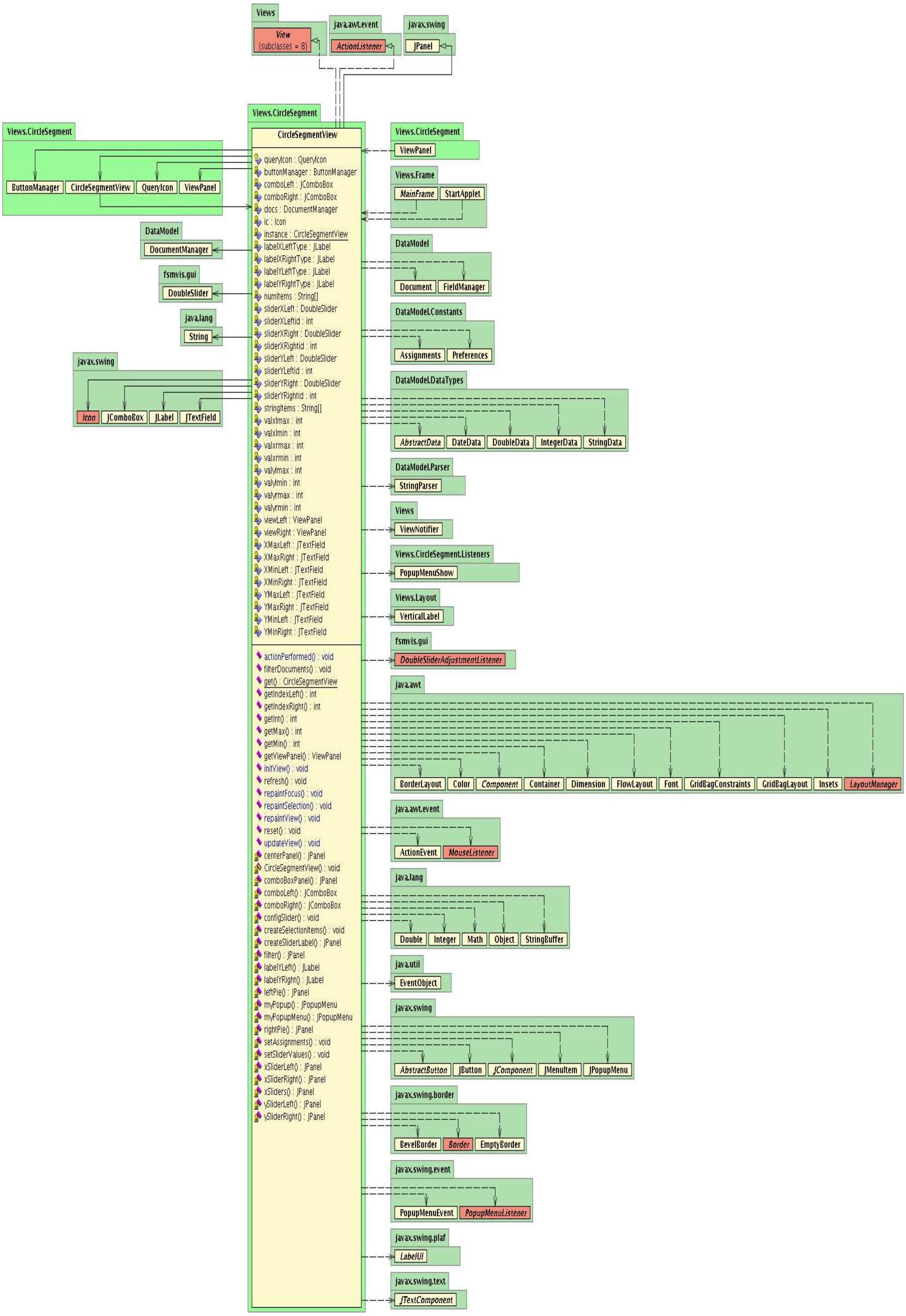


***Figure 6.4**: UML diagram of the CircleSegmentView - a zoomable version can be found on the accompanying CD-ROM.*

by the research results of [CD00], who found that users can handle search results much more efficiently when presented in a categorized, instead of a (common) list-based, view. The categorized visualization was a simple summarization of result items under the appropriate superordinate concept (see Figure 2.2 on page 25).

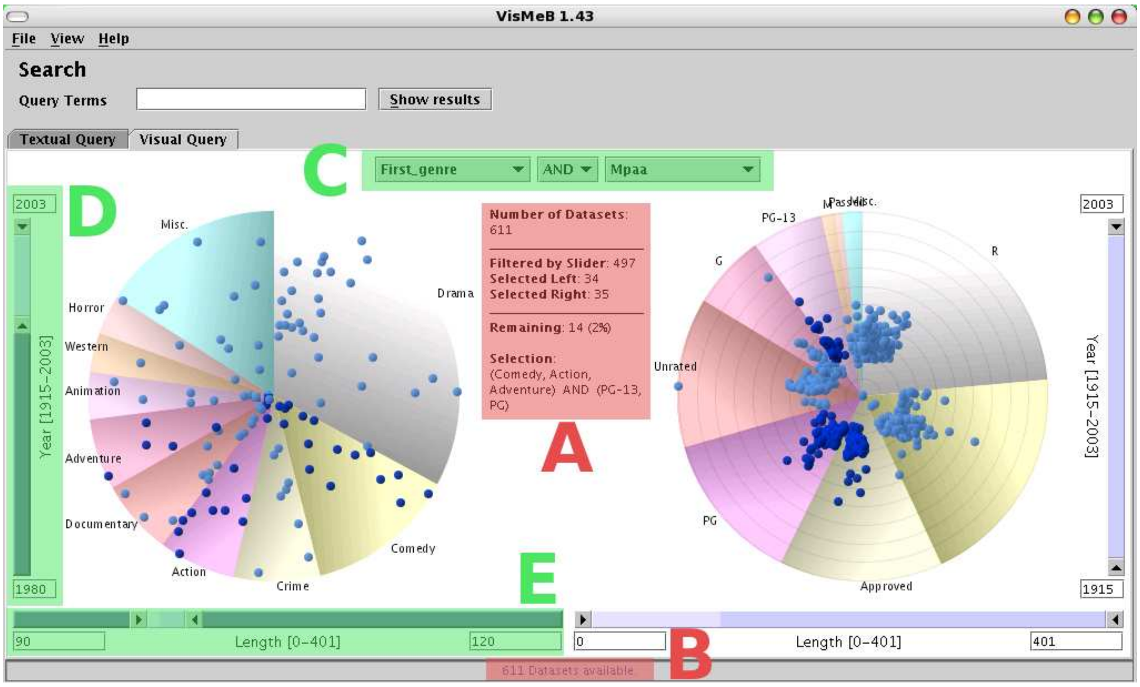


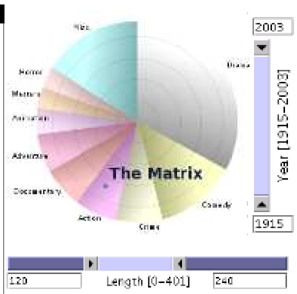
Figure 6.5: CSV - user interface components: A and B - preview areas with information about the selected and filtered data; C - drop down boxes which contain the different categories for each pie menu and the boolean operator to connect them; D and E - the Alphasliders to manipulate the ranges of the numeric meta-data.

The idea of Query Previews [TPS00], where the user has a visualization of the data and a set of controls (such as sliders) by which subsets of the data table can be selected, was adopted. The CSV (see figure 6.5 for the actual user interface), which emphasizes the distribution of the data, also gives continuous feedback about the size of the result-set. Additionally, the users benefit from this because of the prevention of zero-hit or mega-hit results.

For categorical meta-data, the effectiveness of control depends on the data type and on the cardinality of the attributes. So careful consideration has to be given to the question of which data from our databases matches the requirements of the query preview.

The visual representation of the categorized information space is solved by pie charts. The whole circle represents the information space. The different segments show the

MD Attribute	Content	Data Type	Mapping
title	Matrix, The	Nominal	X
director	Andy Wachowski	Nominal	X
first genre	Action	Categorical	Segments
second genre	Thriller	Categorical	Segments
year	1999	Quantitative	Sliders
length	136	Quantitative	Sliders
tagline	Be afraid of the future	Nominal	X
plot	...	Nominal	X
rating	8.5	Ordinal	Sliders
country	USA	Categorical	Segments
orig. language	english	Categorical	Segments
producer	Bruce Berman	Ordinal	X
mpaa	R	Categorical	Segments
color	1	Categorical	Segments
budget	63000000	Quantitative	Sliders
opening weekend	27788331	Quantitative	Sliders



Raw Data → Generated Meta-Data Tables → Visual Mapping

***Figure 6.6**: Meta-data extraction/generation and visual mapping.*

distribution of the data with regard to one type of meta-data. The extent of each segment hints at the size of this categorical attribute. The segments 'start' at the top (twelve o'clock position) with the largest segment and continue clockwise in descending order. The available categories can be selected through a drop-down list which is placed above the circle (see figure 6.5 on the facing page C). For clarity, the number of segments is limited to ten (nine plus the remaining segments, which are summarized in a tenth segment labeled 'Misc'). The dots that represent documents are placed inside the segment using two other numeric meta-data. One meta-data is mapped on the radius (e.g. rank or popularity) and another one on the angle (e.g. release date, or length). The position is rendered, so that greater values lie on the border (for the radius) and at the highest angle (counter-clockwise for the angle). This setting was chosen to offer more space for relevant data (high values were considered more important in our scenarios). The area representing the intersection of both high values is supported by a colour gradient to gain easy access to that information (e.g. the 'newer' and 'larger' documents would lie in the more brightly-drawn area if the chosen meta-data were 'release year' and 'size in bytes'). The selected sets of documents in both pie charts can be linked by a boolean operator. At the moment the system supports the boolean 'AND' and 'OR' operators.

To influence the ranges of the numeric meta-data needed to place the data-dots, we use two Alphasliders [AS94a]; see figure 6.5 on the preceding page, D and E. The first one (vertically orientated) specifies a range for the angle (e.g. 'year' from 1980..2003 as can be seen in figure 6.5 on the facing page), the second (horizontally orientated) a range for the radius (e.g. 'length' from 90..120). The selected range is rendered in a darker shade in order to be easily interpreted. By clicking on the Alphasliders, a pop-up window appears and lets us change the assignment for the radius or the angle. The minimum and maximum values for the sliders are derived directly from the database. Instead of using the sliders, the user also has the possibility of entering the border values in two text fields placed at the end of the sliders. Their use may be the fastest choice if there are nearly as many values as pixels available. If there are more values than pixels available, it may not be possible to use the sliders - so the additional text fields make sense. These text fields always contain the actual border values.

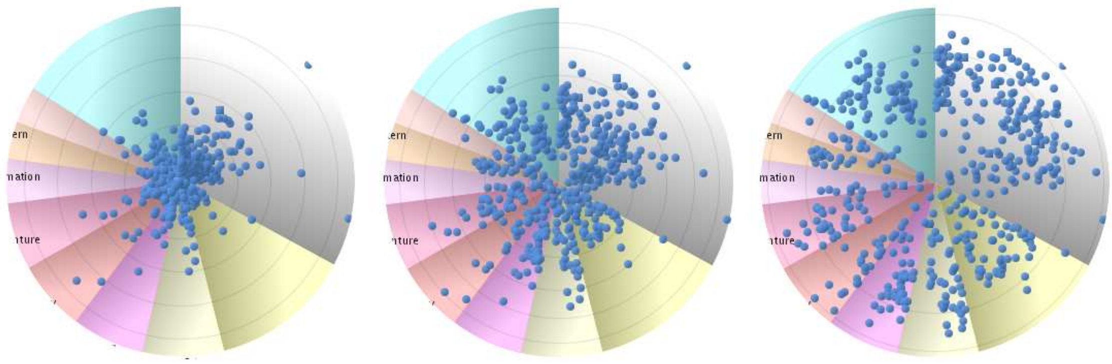


***Figure 6.7**: Different placements for the dots. From left to right: linear, square root and logarithmic placement.*

The CircleSegmentView's merging of a pie chart and a scatter-type visualization has the drawback of wasting screen size compared to other scatter-based visualizations (scatterplot / starfield display). Depending on the distribution of the the data, the linear scaling of the numeric meta-data tends to waste space. This scaling produces dense fields, especially when confronted with outliers. To optimize the placement in the available space, the context menu of the Alphaslider offers different placement algorithms:

- Linear placement
- Square root placement
- Logarithmic placement

The effects can be seen in Figure 6.7; all three pie menus show the same information space. Here it is clear that the linear placement is not the optimal scaling when it comes to avoiding overlapping effects. On the other hand it can be difficult for the user to understand the scaled version, especially the scaling on the angle. The code for the placement of the dots (called button in the code) is outlined in the following source-code snippet:

```
private void setButtonLocations() {

  updateButtons();
 DataButton[] buttons = buttonManager.getButtons(ID);

 for(int i=0; i<buttons.length; i++)
  {
  buttons[i].resetButton();
```

```
  if(! buttons[i].getDocument().isAvailable()) {
  buttons[i].setVisible(false);
  continue;
  }

double xData = CircleSegmentView.get().
getDouble(xIndex, buttons[i].getData(xIndex).getString());
double yData = CircleSegmentView.get().
getDouble(yIndex, buttons[i].getData(yIndex).getString());
double xTotal = xMax - xMin;
double yTotal = yMax - yMin;

// Linear Placement

  if(yAlignment == 1 && xMax != xMin) {

  // Square Root Placement

  xData = xTotal * (Math.sqrt(xData - xMin + 1) - 1) /
        (Math.sqrt(xTotal + 1) - 1) + xMin;
  } else if(yAlignment == 2 && xMax != xMin) {

  // LogarithmicPlacement

  xData = xTotal * Math.log(xData - xMin + 1) /
        Math.log(xTotal + 1) + xMin;
      }

  if(xAlignment == 1 && yMax != yMin) {

  // Square Root Placement

  yData = yTotal * (Math.sqrt(yData - yMin + 1) - 1) /
        (Math.sqrt(yTotal + 1) - 1) + yMin;
  } else if(xAlignment == 2 && yMax != yMin) {

  // LogarithmicPlacement

  yData = yTotal * Math.log(yData - yMin + 1) /
        Math.log(yTotal + 1) + yMin;
      }
  if(FieldManager.getType(xIndex) instanceof IntegerData) {
  xData = (int) xData;
  }
```

```
if(FieldManager.getType(yIndex) instanceof IntegerData) {
yData = (int) yData;
}

boolean act = yData >= yMin && yData <= yMax &&
xData >= xMin && xData <= xMax;
buttons[i].setActive(act);

// if button lies within the circle...

if(act) {
 xData = Math.min(xMax, Math.max(xMin, xData));
 yData = Math.min(yMax, Math.max(yMin, yData));

 int s = buttons[i].getSegmentIndex();

 double angleStart = segments[s].getAngleStart();
 double angleExtent = segments[s].getAngleExtent();
 double angleProp = angleExtent / (Math.max(0.001, xMax-xMin));
 double yProp = (double) radius / (Math.max(0.001, yMax-yMin));

 double yZoom = Math.max(0.001, yMax - yData);
 double xZoom = Math.max(0.001, xData - xMin);
 double buttonRadius = (radius - yZoom * yProp) * 0.7;
 double clockwise = - xZoom * angleProp - angleStart;
 double angle = Math.toRadians(315 + clockwise);

 int xp = (int) (Math.cos(angle) * buttonRadius -
        Math.sin(angle) * buttonRadius + width / 2);
 int yp = (int) (Math.sin(angle) * buttonRadius +
        Math.cos(angle) * buttonRadius + height / 2);

 buttons[i].setLocation(xp, yp);

 // check if another point was already set on this location

 int j;
 for(j=0; j<i; j++)
 {
  if(buttons[j].getLocation().equals(buttons[i].getLocation()))
  {
   buttons[j].addButton(buttons[i]);
   buttons[i].setMultiButton(buttons[j]);
   break;}
  }
```

```
    // no other button has the same location...

   buttons[i].setVisible(j == i);
   if(j == i) buttons[i].calcSize(radius);
   } else {
   buttons[i].setVisible(false);
   }
 }
 newButtons = false;
}
```

6.3 Usage Scenarios and Interaction Techniques

Typical tasks for the CSV are reducing the search space, finding patterns/outliers in a result-set or simply providing an overview of the distribution. The following example gives an outline of the intended usage scenario. This scenario is based on a movie database which provides information (read: meta-data) about films. It is motivated by the fact that movie databases are a common playground for information visualizations and do not require special knowledge of the data/meta-data standards. The use of the system does not differ if a meta-data standard like Dublin Core, ISO19115 or MAB2 is used.

After extraction and generation of meta-data (see figure 6.6 on page 107), the underlying database table looks like table 6.1 on the next page.

6.3.1 Visual Query

A characteristic scenario for a user of a movie database is the selection of a film for her evening entertainment. The user must consider some parameters like 'how long should the running time of the film be' (had a rough day in the office?), 'is the choice acceptable to my partner', and so on. The CSV provides an overview, the totality of all the movies and also some general features of the collection. The initial view of the CSV user-interface is preconfigured so that the circles are segmented by the attributes 'first genre' and 'MPAA'[2] and the numeric meta-data are set to 'year' (mapped to the angle) and 'running time' (mapped to the radius). The viewer has now the possibility of 'zooming' into the collection using the Alphasliders (thus using the zoom function as a filter tool). The user restricts the running time of the movie between 90 and 120 minutes because she wants a full length movie that does not last too long. She does this by setting one end of the two-sided-slider to 90 min and the other end to 120 min; only those cases of the data table whose 'length' variable lies between these limits will

[2] voluntary movie rating system from the Motion Picture Association of America.

Meta-data	Data Type	Movie ID
title	Nominal	The Matrix
director	Nominal	Andy Wachowski
first genre	Categorical	Action
second genre	Categorical	Thriller
year	Quantitative	1999
length	Quantitative	136
rating	Ordinal	8.5
country	Categorical	USA
orig. language	Categorical	English
producer	Nominal	Bruce Berman
distributor	Categorical	AVH
mpaa	Categorical	R
color	Categorical	1
budget	Quantitative	USD 63.000.000
opening weekend	Quantitative	USD 27.788.331
plot	Nominal	...
tag line	Nominal	...

***Table 6.1**: Example of meta-data provided by our movie database and used in VisMeB (extracts).*

be displayed in the results. The preview area tells her that over 300 movies are still available. She continues to reduce the result-set by limiting the year of release to the period from 1980 to 2003 by using the vertical slider. There are still over hundred movies available. It is visually apparent, for example, that a good proportion of the movies in the collection are 'dramas'. But in order to obtain some 'easy viewing' (and going along with her partner) she selects the genres 'comedy', 'action' and 'adventure'. Only 34 movies remain. This set is further reduced by the assumption, that her 13 year old niece could possibly visit her and so, she selects only those movies that are approved for children ('PG-13', 'PG', 'Approved'). The boolean 'AND' operator that binds the two pie charts results in fourteen movies remaining (see figure 6.5 on page 106). If the user were to relax one of the criteria just a little (say reducing the lower boundary of 'year' to 1970), she would get more movies to choose from (in this case: 21).

6.3.2 Visual Filter

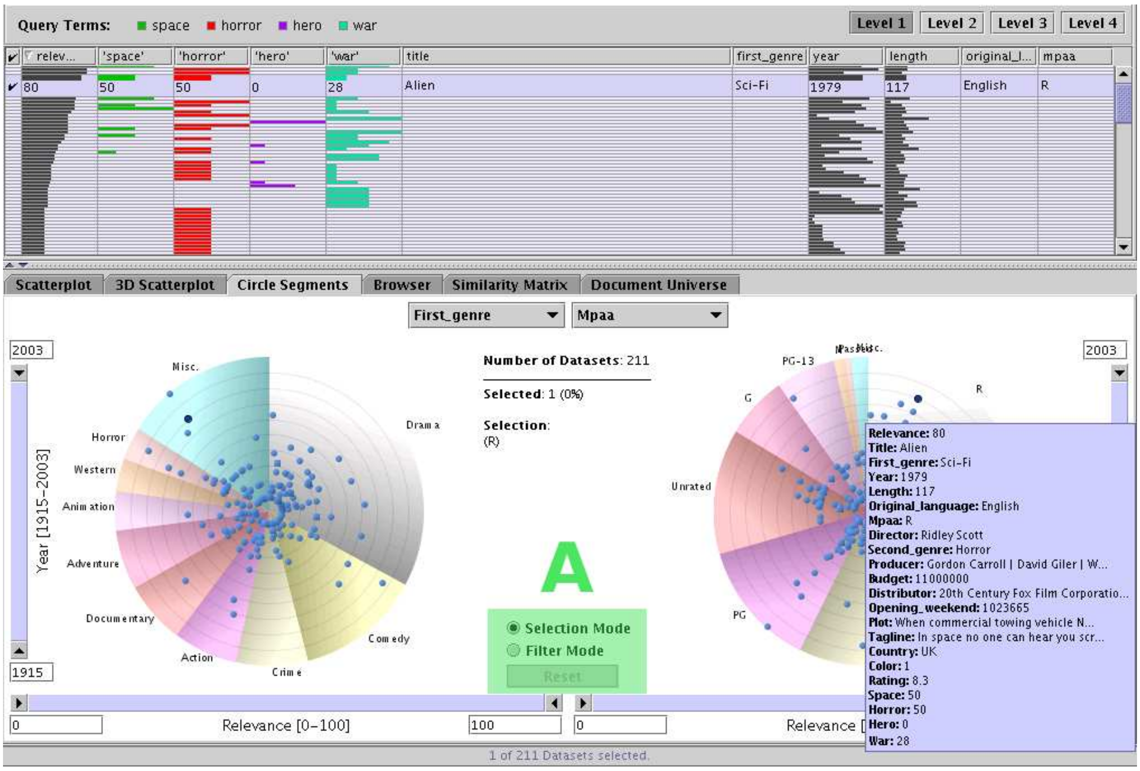


Figure 6.8: The CSV as a filter working in conjunction with VisMeB's other visualizations. Area 'A' shows the added UI elements. Selection of single documents: selected and focused documents in the CSV are highlighted in the table. Tool-tips give details on demand.

When the user works with VisMeB's result visualizations, she is able to use the CSV as a visual filter. For this example let us assume the user gave the system 'space', 'horror', 'hero' and 'war' as relevant keywords and did not set any further restrictions on the meta-data. The interface of the CSV is now given some additional widgets (see figure 6.8 on the page before A): a radio button that toggles the behaviour of the interface widgets (segments, sliders) and a 'reset' button that restores the initial result-set and the initial widget settings. When used in 'selection mode', a single click on a segment selects all documents that are visible in this set. The selection is also visible in the tables through a darkening of the appropriate row (for more information on the SuperTables interaction techniques see [KMRE02] [KMLR03b]) or chapter 5 on page 81. Changing sliders does not alter the number of documents selected. If the user wants an updated view she has to click the segment again after she has modified the sliders. Focal items are made visually distinctive in some way: first the selected data dots are rendered in a different color (red) and are rendered last so that they appear above any non-selected data; second, a tool-tip is provided (see figure 6.8 on the page before) and last but not least the data is highlighted in the appropriate row in the SuperTable.

Another feature appears when the user selects a segment. Now she sees the selected data on the second pie chart as well. This is very helpful if one wants to see the distribution of the selection under another criteria; e.g. how are the R-rated movies distributed among the 'genres'.

Additionally the user gets a context-menu (right mouse button) which offers useful options like:

- Filter selected
- Select all visible
- Deselected all visible
- Invert visible selection
- Select all
- Deselect all
- Invert selection

When used in 'filter mode' the segments and sliders work as assumed: they filter out everything that is not visible. This has an instant impact on the SuperTable (data rows vanish or appear), the layout of the circles (data dots move, vanish and appear) and of course on the preview area (showing the actual set size). The method of instantly responding in the display to the dynamic movement of the slider (and the selection of segments) allows users to explore the multidimensional space of films very quickly.

The placement of the data dots through our positioning algorithm and the above interaction techniques enables the user to rapidly scan for outliers, patterns or distributions under different meta-data criteria.

6.4 Analytic Inspection of the Filter: CSV vs. ScatterPlot

The ScatterPlot is often used as a filter tool in common VISS - see 4.1.2 on page 61. This makes VisMeB's ScatterPlot (see Figure 5.8 on page 90) a natural 'competitor' of the CSV. In this section, both visualizations are analyzed according to their ability to filter effectively and efficiently. Let us begin with a first example.

Example 6.1 *A geologist is interested in finding documents and maps for a certain topic. He is connected to the INVISIP meta-data space through VisMeB. He seeks recent information about environmental changes in germany's highlands north of the Donau.*

This example leaves the following meaningful filter criteria for the INVISIP ISO 19115 information space:

- $Country$ should be set to 'Germany'.
- $Theme_{Code}$ should be set to 'Environment'.
- The Ref_{date} should cover an interval for the last decade.
- The $South_{BC}$ should roughly match the Donau-line (approx. 47).
- Definition for 'Highlands' usually covers 500m - 1500m N.N. $\Rightarrow$ $Min_{evel} = 500$; $Max_{evel} = 1500$.

We can not assume that the assignments of the axes, sliders, circles, etc. already match those requirements, so the re-assignment must be taken into account. The necessary steps for the CSV were:

- Set the category of the first circle to *Country*.
- Set the category of the second circle to *Environment*.
- Set the first slider (left circle, vertically) to *South_BC*.
- Set the second slider (left circle, horizontally) to *Ref_date*.
- Set the third slider (right circle, vertically) to *max_evel*.
- Set the fourth slider (right circle, horizontally) to *min_evel*.

The assignment is now optimized to work on the given problem. The initialization costs 6 actions (mouse movements and/or clicks). To restrict the result-set, the user has to alter these settings:

For each selection of a category, the user has to click on the specific slice (1 action), select 'invert visible selection' (1 action) from the context menu and finally 'filter selected' from the contex menu (1 action) $\Rightarrow$ 6 actions to filter the correct categories.

Now the user switches to the 'Filter Mode' (1 action). He uses every slider to adjust to the needed values. This leaves us with two actions per slider (minimum and maximum value) $\Rightarrow$ 8 actions. The result-set in the SuperTable now shows the remaining maps and documents, that match the user's criteria. He achieved this with 21 actions (see table 6.2 on the facing page for summary) during which he has a constant feedback, on how many maps or documents were available.

The ScatterPlot also needs some initialization in order to offer useful assignments for its axes. The user selects *min_evel* for the y-axis and *Ref_date* for the x-axis (2 actions). Now the user draws an approximate bounding box around the area > 500m and $1993 - 2003$ (1 action). This can not be done precisely because the labelling of the axis and the lack of feedback do not allow an exact positioning for the bounding box. This raises the question of effectiveness at quite an early stage. The user now zooms into that bounding box (1 action). The zooming does not affect the status of the result-set. Visually-unavailable documents are not filtered out of the SuperTable, nor are the visible ones selected or marked in any way. This has to be done in later steps.

The user now uses the *'movable filters'* to mark the interesting documents. Each of the filters work on a single meta-data attribute. To configure a lens, the user needs at least 3 actions (boolean and categorical meta-data need 3, numeric ranges need 4). The user has to configure 4 lenses: *Country* to *'Germany'*, *Theme_Code* to *'Environment'*, *Max_evel* to '1500m' and *South_BC* to 47. This adds up to $12 + 8 = 20$ actions. The lenses must be positioned (1 action), resized (1 action) and combined with the boolean operator 'AND' (1 action) $\Rightarrow$ 12 actions. The appropriate documents that match all criteria are now marked in the resulting color of all lenses added. There is no possibility of filtering them out, or even of sorting them in the table. Thus, wether this task can be called 'solved' is a debatable point.

The number of *'actions'* also leaves room for some discussion. There are great differences in the counted actions, as this example has illustrated. A simple click in a pie-segment carries exactly as much weight as the drawing of a bounding box - which is very difficult to draw accurately. This comparison will not calculate any weights for different actions (let alone a detailed measure like Fitt's law [Fit54]).

***Table 6.2**: Summary of the efficiency for example 6.1*

Steps	CSV	ScatterPlot
Initialization	6	2
Adjustments	14	20
Additional set ups	1	14
Total:	**21**	**36**

Nevertheless, it is apparent that the ScatterPlot takes second place in this example. A rigorous view would even question if the task was solved with the ScatterPlot. To avoid this discussion in the following examples, a set that has been correctly selected with the ScatterPlot as being equivalent to *'filtered out'* is allowed. Alternatively 4 actions are added to the ScatterPlot - this is the amount of steps it takes to activate the filter dialoque, go to the *'selection'* Tab and choose *'Filter selected'*.

Example 6.2 *A movie fan wants to see a selection of the latest and best ranked films available at a video store. He uses VisMeB's IMDB*[3] *assignment.*

This example illustrates a typical 'fast filter' situation, were a user restricts the result-set only on a few meta-data attributes. This time he starts with the ScatterPlot:

The user must again initialize the ScatterPlot (2 actions). He chooses *'Year'* for the x-axis and *'Rating'* (IMDB speak for: Popularity, a scale ranging from 0 (Toothstick in the eye) to 10 (Enlightenment)) for the y-axis. The most popular and recent movies can be found in the upper right corner of the ScatterPlot. He draws a bounding box (this time an accurate drawing is not necessary) (1 action) and chooses *'select all'* (1 acion) from the context menu. Sorting the SuperTable according to the selection status of the documents he gets the desired list. Alternatively we add the 4 action steps to filter the unselected documents out of the information space. So this leaves us with 4 (+4) actions for the ScatterPlot.

Using the CSV, the user starts with initializing the two sliders of the left circle to *'Year'* and *'Rating'* (2 actions). Now he switches to *'filter mode'* (1 action) and uses the sliders to adjust the ranges (2 actions - only one part of each slider has to be moved) $\Rightarrow$ 5 actions.

Table 6.3 on the next page summarizes the steps needed. Again, the CSV wins in this comparision, but followed closely by the ScatterPlot. If we substract the actual filtering steps (which are not really necessary in this scenario) for the ScatterPlot, the result would

[3] Internet Movie Data Base (http://www.imdb.com)

Table 6.3: *Summary of the efficiency for example 6.2*

Steps	CSV	ScatterPlot
Initialization	2	2
Adjustments	2	2
Additional set ups	1	4
Total:	**5**	**8**

be reversed. We now examine this example in a little more detail.

From this point we let the user continue to explore the remaining movies. The ScatterPlot view will not provide him with additional insight on the type of movies without changing the axis, using the *'moveable filter'* or even zooming. Using the CSV, the user has two different categorizations of the remaining movies, e.g. he sees the *'genres'* and the *'MPAA rating'* which could serve as a starting point for further exploration.

What about the power of the result-set? If it is too small or too big, how much effort must the user spend to relax the discriminating / trimming values to adapt the set-size to his needs? For the ScatterPlot, he needs to draw a second bounding box and select *'all visibles'* again (2 actions). In the case of the CSV he can relax the values of the slider to his needs (2 or 4 actions - depending on how many ends of the Alphaslider will be used). Additionally, the user has the opportunity of filtering out additional meta-data attributes by simply clicking on pie-segments. Last but not least, the CSV is the only one of the two visualizations that provides a 'one-click' reset mechanism. We see that, apart from the CSV's predominance in the area of efficiency, it also outperforms the ScatterPlot with regard to features and effectiveness. Let us continue with a medium sized filter approach:

Example 6.3 *A teacher uses a web search engine to look for material he could use in his lectures. He uses the VisMeB applet to access a web-meta search engine. Because he is quite familiar with the internet and its propositions, he knows what to pay attention to. This knowledge combined with the lectures' context should be transfered in the query / filter process. So he knows the following: his lecture deals with geography. The content of .org, .net and .edu sites is usually more apropriate than that of .com sites. Bigger pages indicate more content and more embedded objects. Newer pages usually provide newer content; pages dated 1970 have usually been wrongly dated by accident.*

After the teacher has filled in some key words concerning his lecture, he is presented with the SuperTable together with the CSV. The usual initialization leaves him with the following setting:

- The left circle is set to *'Language'*.
- The right circle is set to *'Server Type'*.
- Alphaslider 1 is set to *'Relevance'*.
- Alphaslider 2 is set to *'Size in Bytes'*.
- Alphaslider 3 is set to *'Date'*.

Alphaslider 4 is not needed and therefore remains untouched ⇒ 5 steps for the initialization. The next step is to filter out all languages except for the one used in his lecture (*'german'*) ⇒ 3 actions. Then he selects the *'Server Types'* .org, .net and .de ⇒ 5 actions. After switching to *'Filter Mode'*, he can use the sliders to adapt the result-set to his needs. He starts with the *'Relevance'* (only the minimum value is raised, resulting in 1 action), then the date (again, only the minimum is altered ⇒ 1 action) and finally the size (ditto ⇒ 1 action). In a perfect session, he is now finised. Because this is a rare event, it is worth mentioning that, from here on, he has total control on relaxing the numeric meta-data with minimum effort (1 action step at a time).

Table 6.4: *Summary of the efficiency for example 6.3*

Steps	CSV	ScatterPlot
Initialization	5	2
Adjustments	11	17
Additional set ups	1	8
Total:	**17**	**27**

Processing this task with the ScatterPlot also starts with the initialization (2 actions). A bounding box (1 action) that limits the *'Date'* (x-axis) and the *'Relevance'* (y-axis) comes next. The zoom step brings the user to the first selection (1 action). Now he uses *'moveable filters'* to restrict the server type (3 MF ⇒ 9 actions), the language (1 MF ⇒ 3 actions) and the size of the documents (1 MF ⇒ 3 actions - exceptionally because only one border is edited). Adjusting the lenses with the required boolean operators (three times 'OR' and one 'AND') uses 4 actions. The usual 4 actions to complete the filter task finish this session. This leaves us with the result shown in table 6.4. Even if those final 4 actions were left out of the calculation, the CSV would clearly win.

This concentration of steps to complete a task may sound unfair for reasons such as:

- . . . but the CSV is cognitively demanding.

- People are used to a ScatterPlot (at least, more so than to a brand-new visualization).
- A lot of features discourage many users.
- ...and so on.

The ScatterPlot as well as the CSV have their advantages and disadvantages. But when it comes to filtering, the CSV clearly wins this competition (at least in the VisMeB implementations). The strengths and weaknesses of the ScatterPlot are summarized on table 6.5. Table 6.6 on the facing page does the same for the CircleSegmentView.

***Table 6.5**: Strengths and Weaknesses of VisMeB's ScatterPlot*

Strength	**Weakness**
⊕ High degree of familiarity.	⊖ Only visual zoom - no impact on the SuperTable.
⊕ Fast overview of information space.	⊖ Minimal support for categories.
⊕ Easy selection via bounding box.	⊖ Missing reset functionality.
⊕ Intuitive use of moveable filter	⊖ More than 3 moveable filters easily confuse the user.
⊕ Basic support for cluster or pattern detection.	⊖ Difficult to draw exact bounding boxes.
	⊖ No real filtering possible.
	⊖ Circumstantial filtering.
	⊖ General lack of features.
	⊖ Only linear placement available.

6.5 Evaluation

One of the main goals for the VisMeB project was to develop usable visualizations. During the whole design and implementation of the VisMeB framework, we followed a user-centered design process [RC02]. The tasks and users were clearly specified by the INVISIP project. The specifications (task analysis, etc.) for the other scenarios (movie, web, library) were provided by our own research. The different evaluations were also defined by our work packages. We conducted *'formative user tests'* with paper mock-ups, and HTML and Java$^{\text{TM}}$ prototypes. Additional *'focus groups'*, *'interviews'* and web-based *'questionnaires'* completed these studies.

Table 6.6: Strengths and Weaknesses of VisMeB's CSV

Strength	Weakness
$\oplus$ Feature-rich.	$\ominus$ Cognitively demanding.
$\oplus$ Separate filter and selection modes.	$\ominus$ Depends on carefully-chosen and properly-formatted meta-data.
$\oplus$ Real zoom = Filter mode.	$\ominus$ Reaction times do not scale very well with set size.
$\oplus$ Simple, fast and intuitive selection of categories	$\ominus$ Accuracy of the Alphaslider decreases with higher numeric ranges.
$\oplus$ Dynamic query with detailed preview.	
$\oplus$ Simple and fast relaxing of numeric meta-data.	
$\oplus$ Powerful pattern-recognition under different aspects.	
$\oplus$ Fast navigation to numeric ranges via Alphaslider.	
$\oplus$ Linear, square root and logarithmic positioning available	
$\oplus$ Powerful combination of different filter settings (4 numeric and 2 categorical).	

6.5.1 Motivation

To date, the common user interface for querying databases is still based on forms. That is why a user study is to be used to compare the CircleSegmentView to a common form-based user interface. The goal is to see if there are benefits for the user when working with the CircleSegementView. For this purpose, a form was designed that enabled querying of our Movie-database with the same power as the CSV. The test was planned with the hypothesis that the use of the CSV provides a statistical significant advantage over a form-based user interface.

Due to a user study performed by Tanin [TPS00], which also tested a query preview interface against a form-fill-in interface, it is believed, that the subjective satisfaction of the users is higher when the CSV is used.

6.5.2 Test Methodology

There exist several ways to evaluate a user interface. The *'heuristic evaluation'* describes a method where a few experts examine a given user interface (usually supported with a standardizes catalogue of norms). It is a powerful tool for discovering most of the existing usability flaws. The advantages of a *'heuristic evaluation'* are the low costs compared to other methods, and the relatively high amount of flaws discovered. One of the disadvantages is that you have to have several usability experts at hand.

Another possibility for an examination is the *'cognitive walkthrough'*. The developer should test his work against an antecedent list of claims. This is a very low-cost procedure, but this method usually opens the door for repeating faults.

'(Formative) usability studies' stand the test of providing good results. Although they are expensive in both time and money, they deliver high-quality results. During a *'usability study'* the participants have to solve several tasks. The efficiency is usually measured in time and/or key-strokes or mouse moves. The tasks are ideally designed to cover all aspects of the user interface. The costs for a *'usability study'* are usually a problem. At the Department of Computer and Information Science we have access to a mobile Usability Lab. What remains are the costs of paying the subjects and the time to design and perform the study, as well as the time needed for an in-depth analysis of the results. The settings were chosen according to the recommendations taken from Mayhew [May99]. This form of testing is based on two phases: the planning stage and the actual performance stage.

The *'planning and conception'* stage was characterized by preparing a session that complied with the following points:

- Focus on *'ease of use'*.
- Define the user and passure a sufficient number of them.
- Define pre- and post-test questionnaires.
- Prepare the settings.
- Define the scenario and the appropriate tasks.

We wanted users, who were familiar with the CircleSegementView. Unfortunately this would have restricted the potential users to the few members of the HCI-Lab at the University of Konstanz and a handful of users from the INVISIP project. In order not to falsify any results by using only experts for this test, the decision was taken to create appropriate users through training sessions. Fortunately, the new profile for an acceptable user was now serendipitously a more promising one: The user should have had experience with standard office applications for at least 2 years and should be comfortable using internet services. The remaining gap should be filled with training on the subject.

Before we could get started with the training, we had to define the training goals [RH94]. We aimed to train the user in a way that they are able to understand and to use all controls and GUI widgets. Furthermore, they should be able to successfully solve simple tasks. The extent of training a single person depends solely on the time he/she needs to fulfill the training exercises. The briefing of a test user can be performed with several different test materials. We deprecated video sessions or written directives because these usually:

- Leaves the user with questions (that he is afraid to ask)
- Are not designed for individuals
- Misguide the user (or the user misses some important facts)
- ...

For this reason the users were coached verbally. People are usually more attentive in a conversation. The test leader also acquires an understanding of the skill-level of the user through the interaction with him/her.

The test procedure followed the standard strategy for this situation [May99]:

1. Welcome.
2. Pre-test questionnaire, NDA, recording accommodation.
3. Training (approx. 15 minutes).
4. Test session.
5. Post-test questionnaire and interview.
6. Payment, farewell.

The independent variable is the type of user interface and the treatments are:

- A form fill-in interface (see Figure 6.9 on the next page) and
- The CircleSegmentView.

After an introduction to the user interface and the interaction techniques, the test persons had to solve some training tasks with guidance. These sessions lasted between 10 and 15 minutes. This was definitely not enough, as we saw during the tests. Some users showed clearly that they were unable to use the full potential of the CSV (as well as that of the form-based interface). Because of this, it is planned to perform a follow-up study with more intensive training sessions.

The users (n=20; experienced PC users) had to solve nine different tasks. The tasks can be divided into three categories:

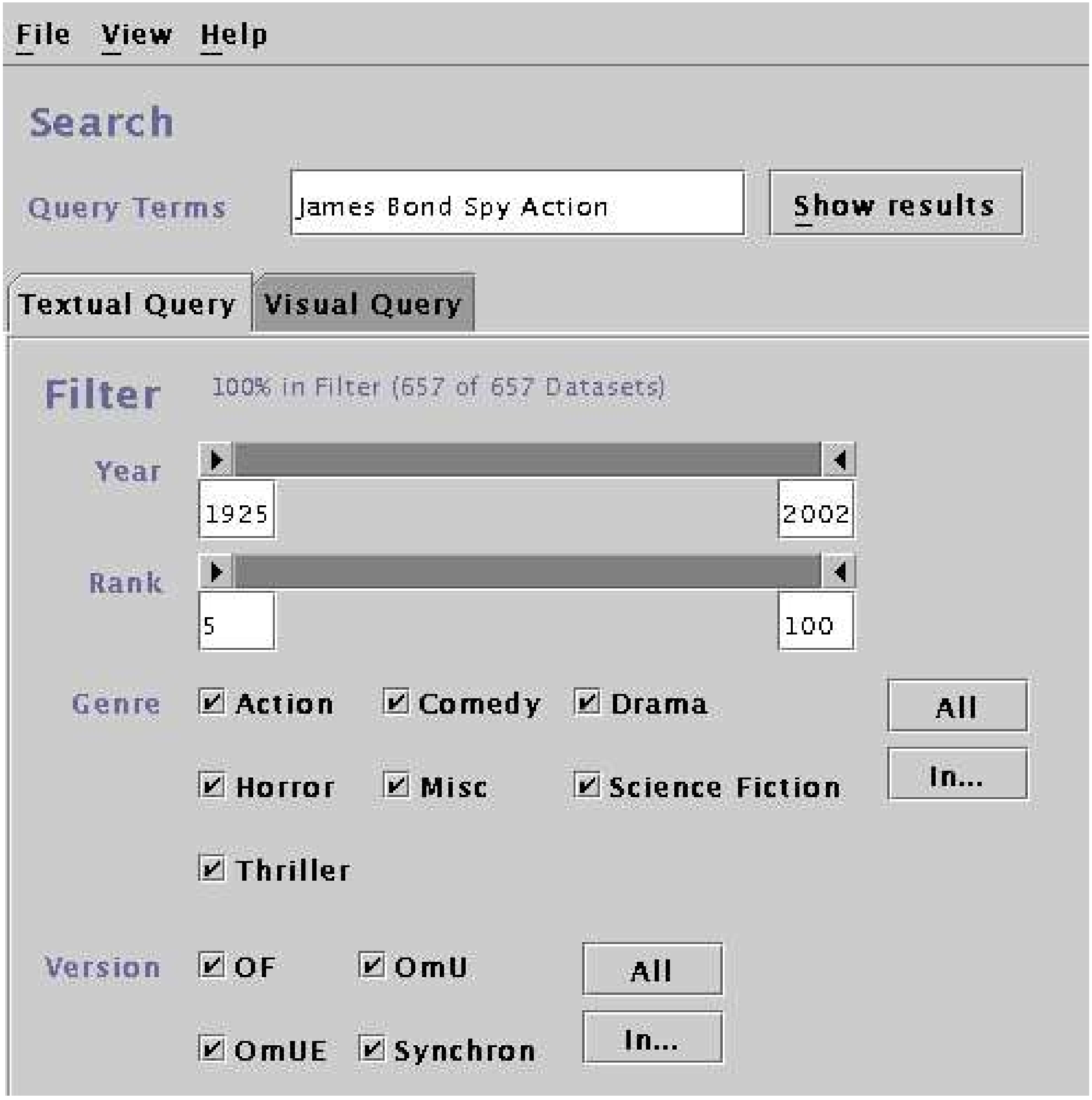


Figure 6.9: The form fill-in interface used in the evaluation.

A Questions about the distribution of documents (Task A).

B Questions about single documents (Task B).

C Comparison of different document attributes (Task C).

This set of tasks should emphasize the strengths (Tasks A, C) and weaknesses (Task B) of the CSV. Each category was represented by three tasks. Each subject was tested on both of the interfaces, but the order of the interfaces was reversed for half of the users. The tasks for each category differ slightly from GUI to GUI, for example:

A1 - CSV How many action films are in the database?

A1 - Form How many comedy films are in the database?

The sequence of UI testing had no influence on the results. We addressed this topic through the calculation of a U- test by Wilcoxon, Mann and Whitney [RH94]. To measure the efficiency we took the time to complete a task. Tasks that lasted more than 5 minutes, or were abandoned by the user, were marked as 'not solved'. If the user needed guidance or a hint, we also marked this task as 'not solved'.

Our hypotheses for this setting were very conservative and are expressed by this two-tailed hypothesis:

H0 Both interfaces provide the same degree of efficiency for solving the tasks.

H1 One of the query interfaces is significantly better suited.

Looking at the raw data, one could easily see that a lot of tasks were solved faster with the CSV (Figure 6.10 on the following page). We used the Shapiro-Wilk test [SW65] to examine for distributional adequacy. Only tasks A1, A2, B3 and C2 fulfilled the requirements. Next we calculated a sign test by Dixon that works on our results. To prove this statistically relevant, we ran Wilcoxon's matched-pairs rank test.

The results of this analysis are summarized in table 6.7 on the next page. The analysis and the raw data is based upon the data from [Bey04], where all original data can be found. A version not taking the distributional adequacy into account (perhaps an overkill on statistics) is added as a second analysis where the missing times (user did not solve the task $\rightarrow$ 300sec.) were substituted by the mean value of the appropriate task. Then a simple ANOVA (Analysis of Variance) is calculated. This can be found in the appendix on page 143. This version shows no significant difference between the CSV and the form for tasks A and C; for task B the form wins again.

We gave the users the opportunity to state their subjective satisfaction in a post-test questionnaire. We asked about intuitiveness, effectiveness, and joy of use and if they

Task	Description	Winner
1	Distribution	Draw
2	Needle in Haystack	Form
3	Set comparison	Draw (CSV[4])

***Table 6.7**: Results for the evaluation*

could imagine working with the CSV on a regular basis. The CSV clearly lost in the area of 'intuitive and comprehensive user interface'. In all other areas the CSV got more positive than negative ratings. Very promising was the the result of the last question: "Could you imagine working with the CSV on a daily basis?" which 19 out of 20 users answered with 'Yes'. However, the free associations of the users were clearly in favor of the form fill-in interface. We could trace this result to the facts, that the subjects were used to forms and that the CSV interface was not familiar to them. In addition the powerful and feature-rich user interface demands more training.

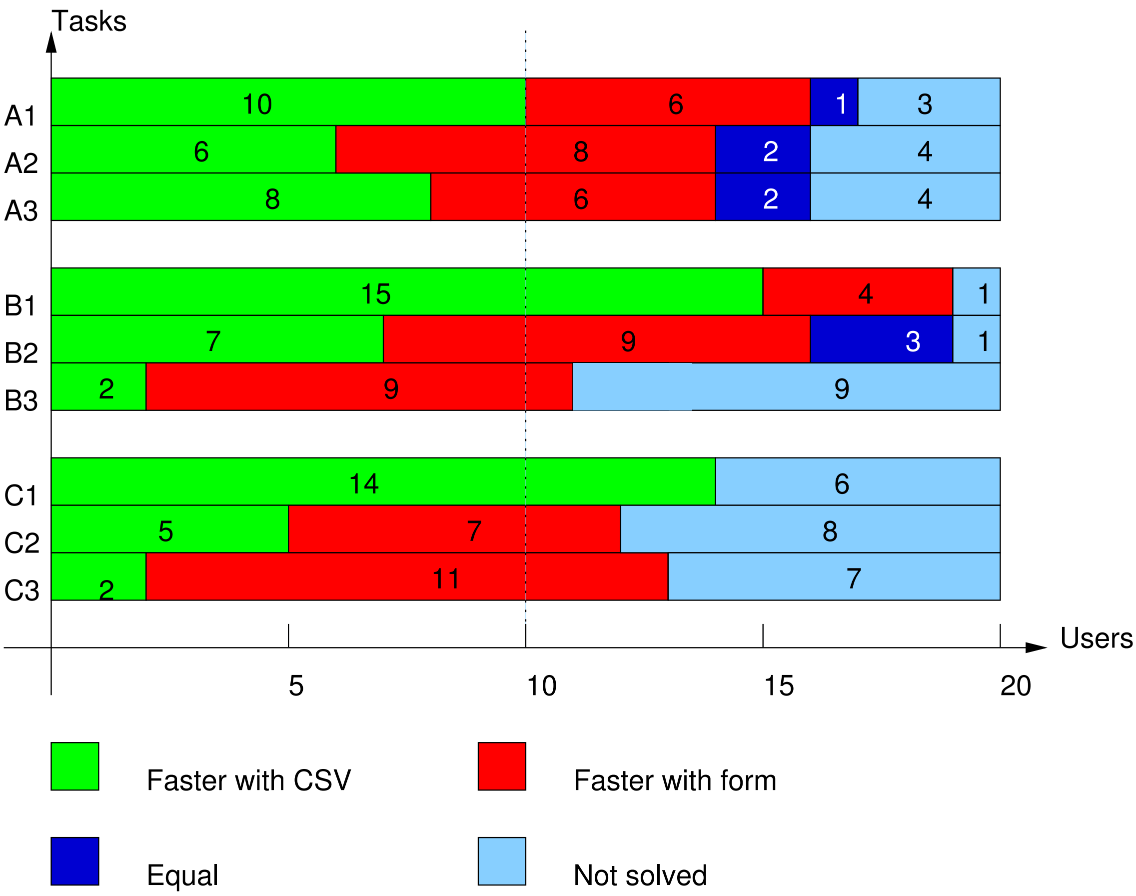


***Figure 6.10**: Raw data of the comparative usability test.*

6.6 Summary and Outlook

This chapter introduces a visualization specially designed to work as a query preview and as a visual filter. The whole VisMeB Framework is easily adaptable to a wide range of data sources. If the data source provides meta-data according to a standard (e.g. Dublin Core or ISO19115 for geo-meta-data), almost all of the configuration work is done: only the mapping from the views to the visualization is left. If on the other hand we are faced with raw data, analysis of the data and generation of meta-data and views is unavoidable.

The CSV is focused on human search behaviour and on a user-centered design process [KR05]. The visualization itself is based on various information-visualization paradigms such as:

- ”Space is perceptually dominant, it is good for discriminating values and picking out patterns” [Mac95]. → Positioning algorithm, data relations mapped to visual encodings.
- Dynamic queries [TPS00]. → Instant update on distribution data.
- ’Details on demand’, ’overview and detail’. → Tool-tips, interaction with SuperTable.
- Brushing and linking. → Interacting with VisMeB’s other visualizations.
- Overview first, zoom and filter, then details on demand [Shn04]. → Information Retrieval pipeline as stated in section 3.3 on page 38.

The evaluations so far were encouraging and it is planned to perform more usability tests with a slightly changed setup to compensate for the novel user interface, e.g. more training sessions, long-term observation and a comparative evaluation to find out about the suitability as a visual filter (e.g. ScatterPlot vs. CSV). A promising approach seems to be the remote usability project called DROID [Jet03][5].

DROID specifies a software framework and server application for capturing user behavior in all kinds of applications and on different platforms (desktop, cellular phones, PDAs). DROID will automatically collect data about user behaviour in the background of day-by-day operation and transmit all relevant user interactions and system incidents within an application over the web to a central logging server. This server plays the role of a usability data warehouse which provides a steady flow of usability-relevant data during development and post-deployment phase. Key features of DROID are:

- **Dynamic Logging** - the amount and the focus of transmitted data will be dynamically adapted to current usability questions, user privacy and available bandwidth

[5]Dynamic Remote Operation Incident Detection

- **Remote Testing** - the logging does not need to be done in a lab environment, but can be performed over the net from every application that is equipped with DROID components

- **Operation Incident Detection** - all incidents or system events (ranging from normal UI operations like mouse selection or keyboard input to bug reports after system crashes) can be automatically detected in the background or can be manually triggered by the user

As a method of remote usability testing DROID will offer many advantages compared to classical usability data collection in the lab:

- Everyday tasks are performed by real users

- The users are located in normal working environments

- The data is captured in day-by-day task situations and not during artificial test settings

- Data capture is highly cost-effective and can be dynamically focused on current usability issues

- All installations with web access can participate in usability evaluation without the need for installing and activating special event-recording software

- No direct interaction between evaluator and user is necessary ('24-7' evaluation)

- High quality machine-readable data without the need for intense human assessment and analysis of video material

- Different levels of data capture: from simple start-/stop-session logging for usage-frequency statistics to detailed capturing of mouse motion and keyboard events

- Additional possibilities like Quality Feedback Agents or user-reported critical incidents can support the post-deployment phase [HC98].

The DROID framework is currently installed on a VisMeB adoption for the campus media library (a system called *MedioVis*). Unfortunately, the CSV was not included in that project because the focus was placed on novice and casual users. Using DROID together with the CSV in an appropriate scenario would certainly enhance usability.

The future development includes investgation and design of new interface widgets that work more intuitively than, for example, the sliders. A first mock-up can be seen in Figure 6.11 on the next page. Here, the information space and its possible limitation to just two numeric meta-data is rendered as a quarter pie and the intervals can be altered by small boxes on the radius (bottom) and on the arc. The remaining information space

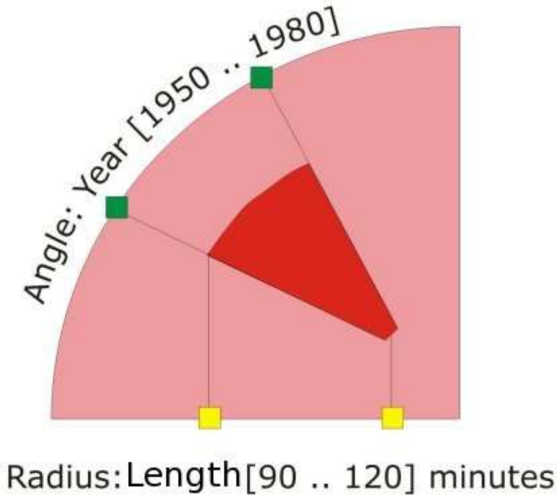


Figure 6.11: *Intuitive widget as a replacement for both Alphasliders.*

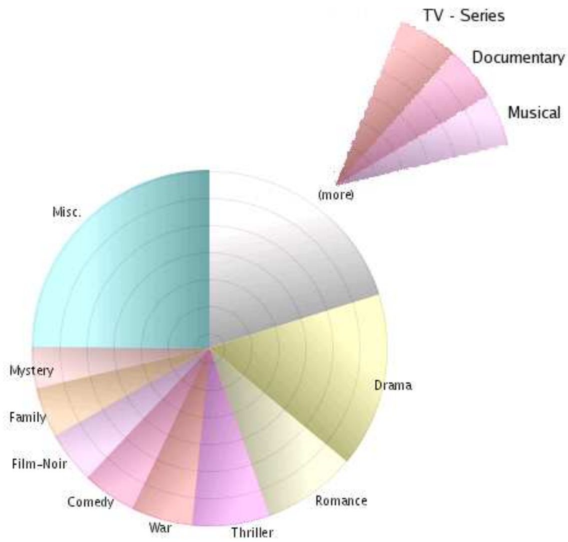


Figure 6.12: *Mockup for a sub-section or 2nd stage segmentation.*

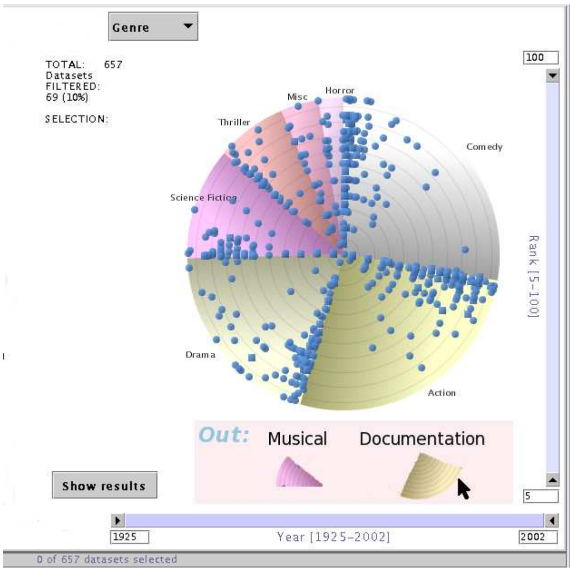


***Figure 6.13**: Drag and drop for slices: visualization updates after recalculation of the remaining data dots.*

(rendered as dark red area) can be moved by the user (similar to the behavior of the Alphaslider).

Improvements to the preview could include the direct labelling of the segments, so the users see not only which category is represented by which segment, but also how much selected/unselected data is in this wedge (e.g. Comedy: 20/45 selected).

Another interesting idea is the sub-division of a segment (see Figure 6.12 on the preceding page). Clicking on a segment allows a quarter pie with subsegments to pop up; for example, clicking on 'More' reveals functions for a detailed sub-segmentation like 'TV-Series', 'Documentary' and 'Musical'.

Some drag and drop functions for segments (to filter whole segments out of the calculation: the user drags the slice 'Comedy' from the pie and the freed space is used up by the remaining slices) are features that should be considered in a next release (see Figure 6.13). Further 'to do' items would include:

- More usability studies or use of DROID as stated above.
- Transfer the knowldege to other information-seeking systems and every-day applications like Email, Web Search Engines, Product Catalogues, Visual Data Mining, Corporate Knowledge Repository ...
- Stripped-down versions for novice users?

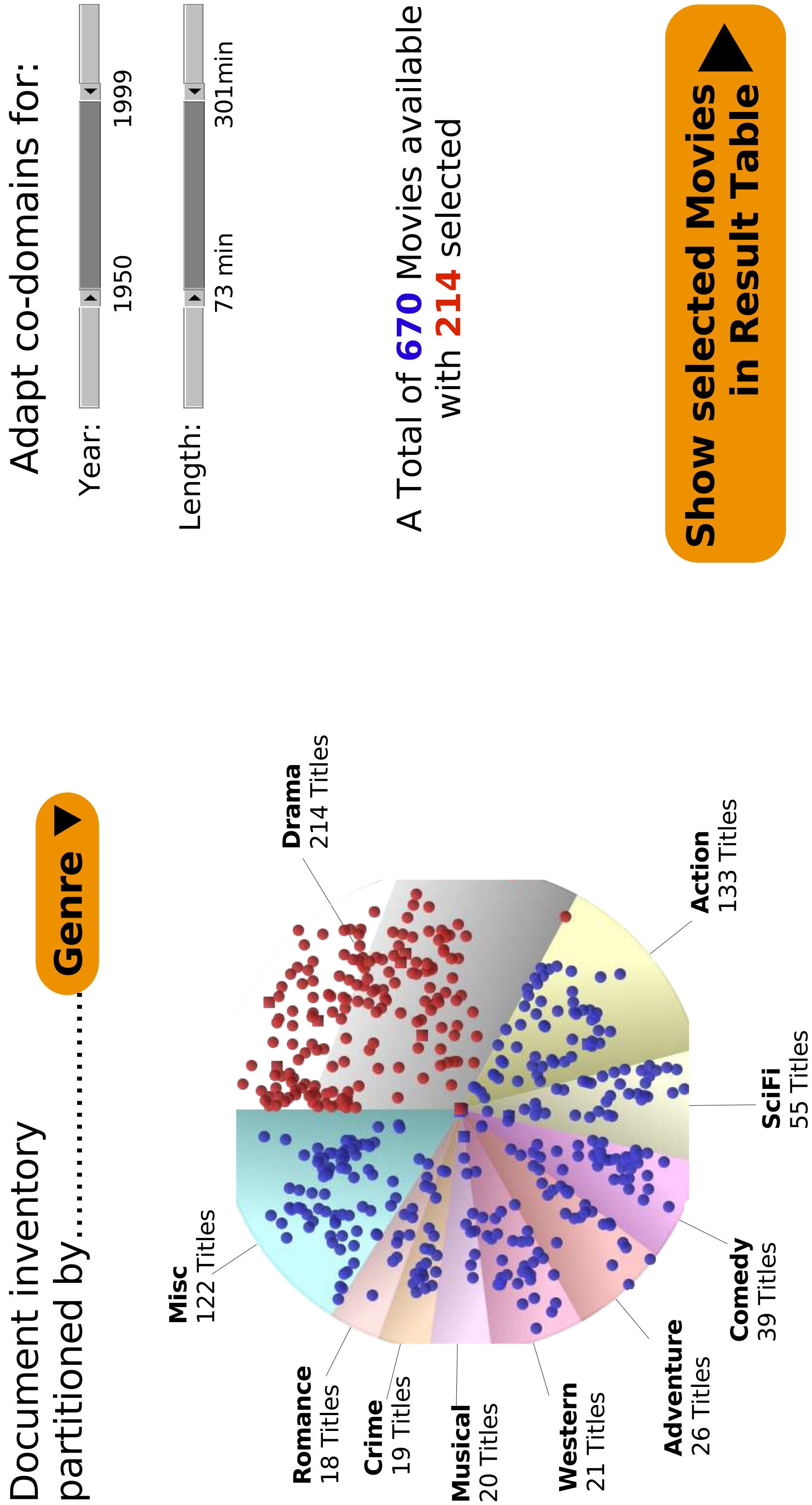


***Figure 6.14**: Mockup for a CircleSegmentView 'light'.*

The last point seems an interesting idea. What has to be changed to give the CSV a self-explanatory interface that a novice user could handle? Let us start with the definition of a 'novice' user:

Definition 6.1 (Novice User) *A novice user is assumed to know little to nothing of the task or the interface concept. Because of the peculiar situation of visual information-seeking, the novice user was assumed to have knowledge about general capabilities like email writing and mouse usage.*

This group of users may arrive with learning-inhibiting anxiety about the human-machine interface. The first step in overcoming the limitations is a reduction in the complexity of the UI. For a simple *Dynamic Query* interface, one pie chart is sufficient. Restricting the vocabulary to natural language and context-sensitive terms is the next step. The number of available actions should be small, so that the novice user can carry out simple tasks successfully and thus reduce anxiety, build confidence and gain positive reinforcement. Informative feedback about the accomplishment of each task is helpful. Task-oriented online help may be effective to get the user started (e.g. 'Post-it note'-like messages with descriptive help). A LoFi mock-up can be seen in Figure 6.14 on the preceding page. The segment labelling is now separated from the segments themselves and contains more information. The preview area is held in natural language, as is the case for the segmentation box on top of the pie chart. The positioning algorithm is hidden from the user, so he is not able to determine why a document dot is rendered at its specific location. The user is also limited to the two most common numeric meta-data.

The typical user as stated by the INVISIP project was a trained *'power user'*. Even though we addressed the current design to that group, it is still possible to further enhance the UI for that group. Again, let us start with a definition for the 'power' user.

Definition 6.2 (Power User:) *are frequent users, trained on the interface, with an extensive knowledge of the underlying data repository and they are experts in the application domain. They need powerful controls to optimize on efficiency.*

The enhancements for *power users* affect not only the visible parts. To raise efficiency it is important to optimize mouse movements, apply short-cuts, make the system mouse-independent, etc. It is also very important to offer tools like ReDo / UnDo, save a session or a history tree.

The enhancements for the visualization could consist of the option to provide more circles and different boolean operators (on the assumption, that the screen resolution for an information broker is usually HDTV (1920x1200) or two 1600x1200 displays). This solution could offer various different views of the data. Merged with the improvements mentioned on page 129, the usability of the interface improves.

Recent cooperation with design students from the Fachhochschule Konstanz provided some major changes in the interface appearance (see Figure 6.15 and 6.16 on page 135). The visualization of the CircleSegmentView was completely revised. The color gradient in the segments that should emphasize the numeric meta-data that is mapped on the angle was introduced into the widget that controls these ranges. The mapping for the radius is also emphasized in the control slider by different line thickness. The change from the different categories is more obvious when used as a more prominent drop-down widget. The subject of labelling was also addressed and a legend was introduced to eliminate the case of overlapping labels. Nonetheless, it would be a good idea to emphasize that point,

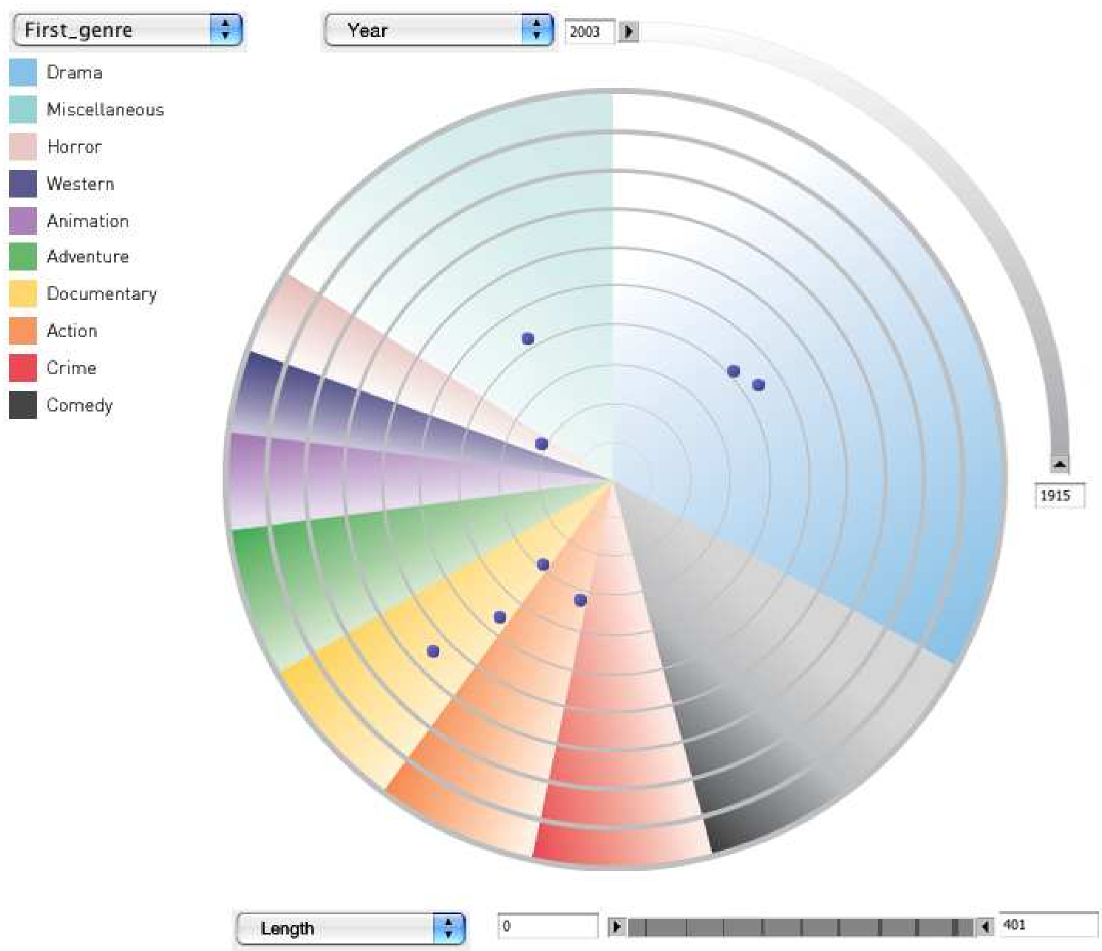


Figure 6.15: *Mockup for a CircleSegmentView based on a design by students from FH Konstanz (detailed view). Segments are labeled with the number of documents they contain. A Mouse-over effect for the labels improves perception.*

perhaps with additional 'help' lines from the legend to the respective segment.

Both circles are now presented asymmetrically to avoid any associations of symmetry concerning the data. The control elements for filtering and selecting, as well as the information about the (selected, filtered) data, are moved between table and visualization in order to make in more obvious that these controls affect both: the SuperTable and the CSV.

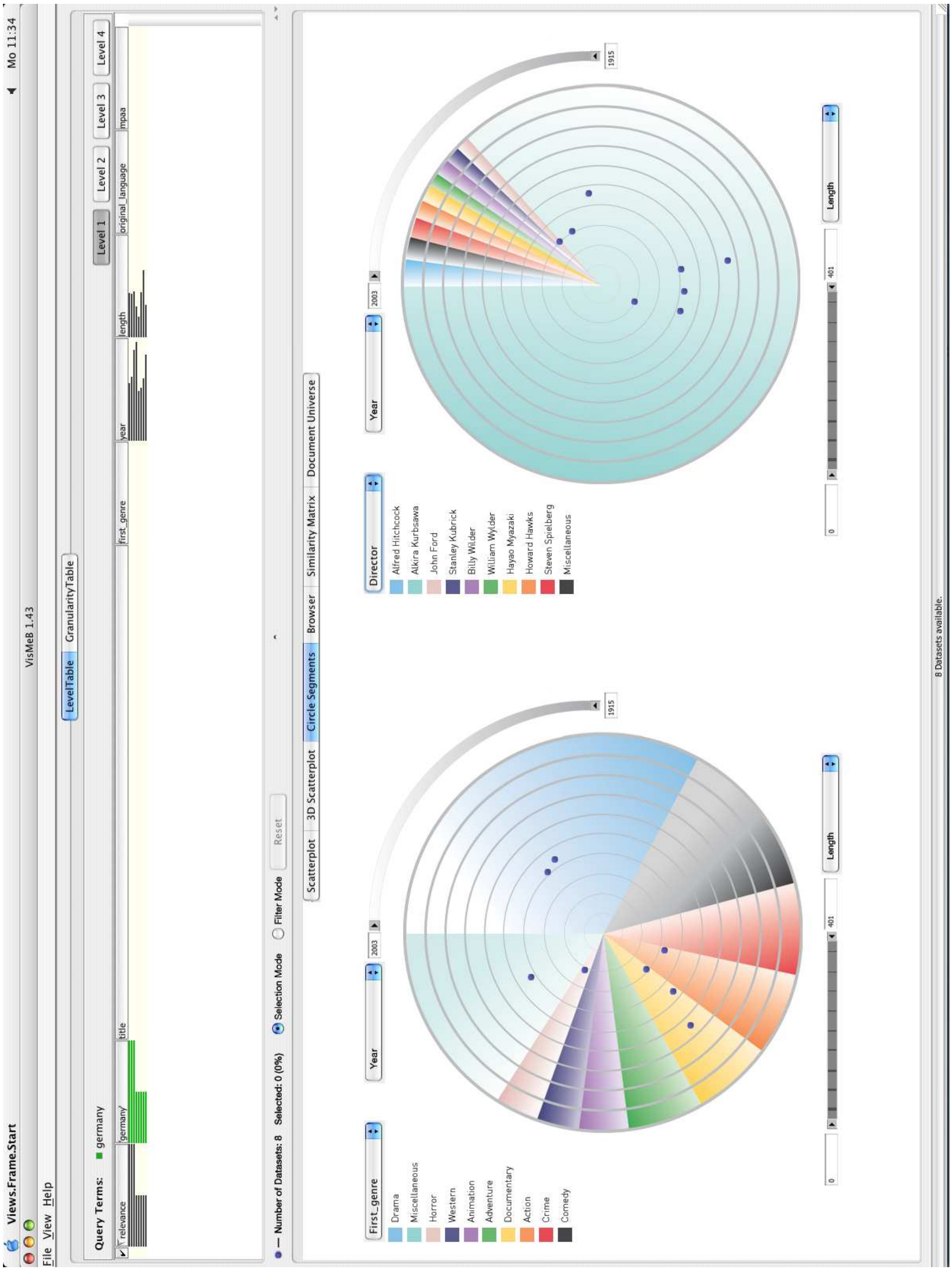


***Figure 6.16**: Mockup for a CircleSegmentView by design students from FH Konstanz (full view).*

7 CONCLUSION

When building a visual information-seeking system, it is important to consider the layered aspects of each component. With regard to the research disciplines of information retrieval, visualization and human-machine interaction, each tends to stay within its own framework. In order to develop a successful system, it is demandable to work closer together. A first step in this scenario could be to reduce the complex models of each discipline to a core, and to distil the most important steps, identify possible interfaces between the models and suggest further interaction. Models like the information-visualization reference model by Card try to merge the data side with the visualization side. Other attempts focus on the interdisciplinary work between human-machine interaction and information visualization.

During the INVISIP project (and any other project, that has a VISS as a goal will face similar problems), three major aspects of a visual information-seeking system were identified:

1. Data
2. Visualizations
3. HCI

Data problems accompanied us during the whole project. Different national Geo-meta-data standards, lack of control and poor quality of meta-data were just a few, but nevertheless troublesome, shortcomings (Figure 7.1 on the next page mentions the key parts).

Visualizations and the UI were also a source of potential pitfalls. From the researches point of view, the specifications for searching and browsing the available data included the provision of support for the geo-meta-market, among others. The context of geospatial data should be an integral part, with interfaces to Geo Information Systems (Figure 7.2 on the following page shows the desired kinds of visualizations).

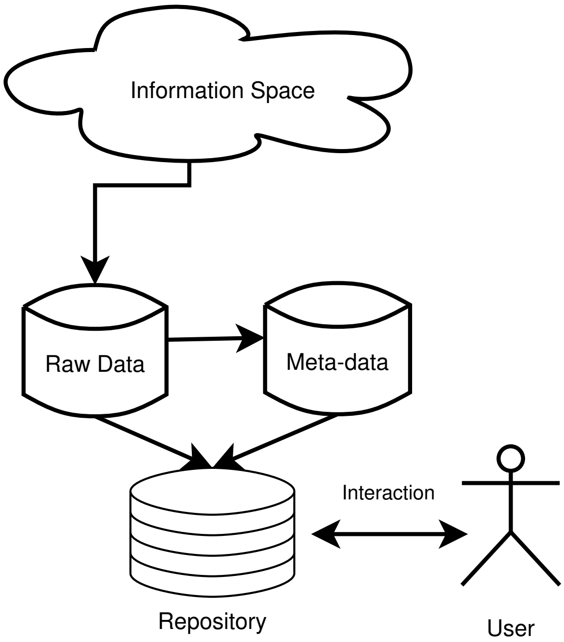


Figure 7.1: *Data sources.*

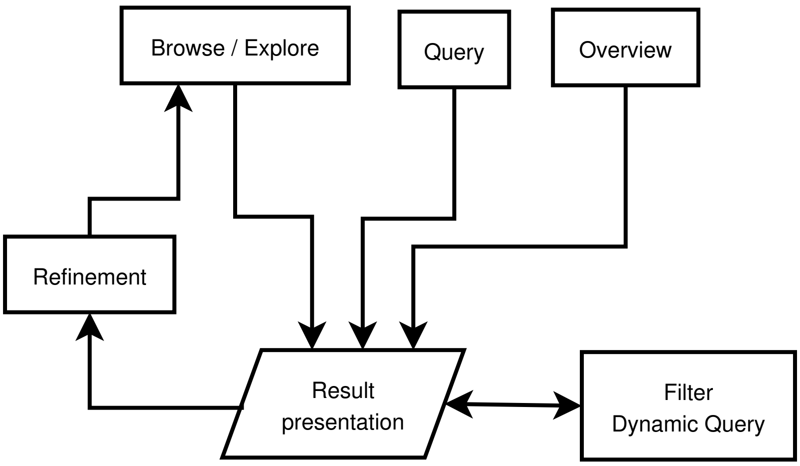


Figure 7.2: *Visualization specification.*

The User was clearly defined by the project, but initial interviews identified a wide spectrum of possible variables. These ranges from different experiences with different query interfaces to vocabulary differences that constitute a 'cultural burden' (for example, there is no direct German equivalent for the Polish word that translates as 'masterplan' but actually means 'Raumordnungsplan' (local area development plan)). Figure 7.3 shows some considerations for the user and his/her task.

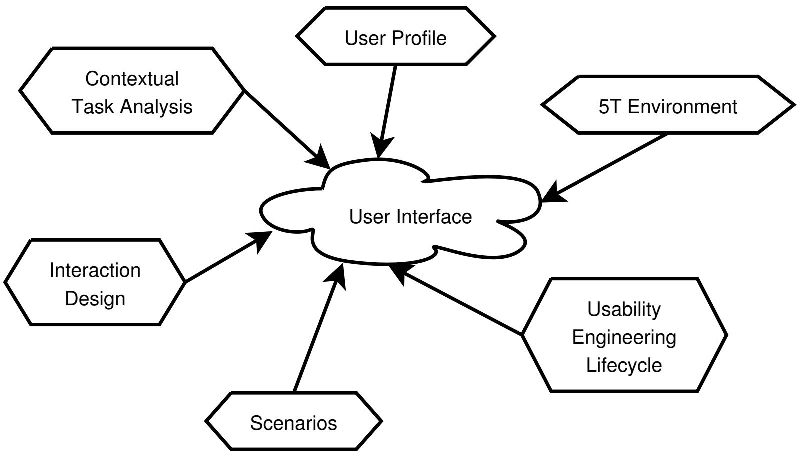


Figure 7.3: *Visualization specification.*

All three aspects have impacts on one another. Some steps in the development can be taken in parallel, some have to influence other parts. To achieve optimum results, an integrated approach to these three major aspects could prove interesting (as stated in 3.3 on page 38).

The main consideration is focused on two areas: firstly, the user and his/her tasks and secondly, the available data. The analysis of both scopes is described in the literature as reviewed in chapter 3 on page 31 for the data part, and in section 3.3 on page 38 for the user part. The first steps in this phase of the project can be executed in parallel. Whereas the 'User Profile', 'Contextual Task Analysis' and other considerations (like 'Platform Capability' ...) result in a definition of 'Usability / Feasibility Goals' and specific 'Design Principles', the corresponding analysis of the available raw data results in a 'Meta-data Specification' (see Figure 3.4 on page 42). This specification also receives input from the 'user-side' so that it is possible to integrate the human factors in the meta-data specification. Such valuable information could contain answers to questions such as:

- What kind of attributes is the user interested in?
- Which data format does he/she prefer?
- What information is expected at which level (AOI / DOI)?

Based on these formal specifications, the next step includes the building of a so called 'Meta-data Toolbox' which includes utilities for data entry, data evaluation and maintenance as well as an 'Aging / Quality Strategy' for data. These tools will define the backbone of the data repository.

The user side is emphatically characterized by the 'mockup - evaluation - cycle'. The first LoFi prototypes should determine if the design principle is usable and if an effective UI is feasible. The HiFi prototypes are used to check the interaction design and eliminate any remaining usability flaws. The results of these evaluations should no longer have an impact on the 'Meta-data Specification'. But if this eventuality arises, it should be compensated for by the availability of the 'Meta-data Toolbox'.

The final steps of this strategy include the fine-tuning of the database or the underlying information repository (such as: building hashes, indices, views, ...) as well as usability studies with a first version of the UI.

During the whole process, the work on the visualizations must be closely coordinated with both sides. The *visual information-seeking mantra* as stated by Shneiderman is the foundation for the collaboration with the user side. Visualizations must be chosen, and interaction design must be specified to feed the testing-evaluation circuit. The results of the evaluations represent input for redesign ideas.

Visualizations rest upon the data. Questions about the data types, data formats and the availablity must be laid down in the 'Meta-data Specification'.

All these lessons were learned during the development of VisMeB and the CircleSegmentView. This visualization offers an overview of the available documents / data. The chosen solution for this kind of introduction to the information seeking process emphasize the human search behavior as stated by Bates integrated model of information seeking and searching (see section 2.2.4 on page 22). It involves the user's knowledge of the data (meta-data) and gives him/her control over the information space. The support for categorical types of meta-data is the strength of the CSV. This categorical representation of the information space supports human search behavior. The possibilities for filtering and zooming different aspects of the information space offer powerful tools for the professional information worker. The concept of multiple CSVs combined through boolean expressions is implemented for two CSVs. This enhances the basic possibilities but increases the cognitive load for the user.

The CSV not only perfomed well as a query preview tool but was also very useful and powerful as a visual filter in combination with VisMeB's other visualizations. By combining the characteristics of scatter charts and pie charts, the CSV reveals information not only about distribution of data but also about single documents, clusters and semantic relations (when used with the second CSV). Easy relaxation of meta-data boundaries invites the user to explore (or: browse) the information space. In using this visualization, the user will never again be faced with zero or mega-hit problems.

The CSV has proven to be a promising visualization for dynamic querying and for working as a filter. The results of the evaluations, together with the redesign ideas from section 6.6 on page 127, should help build effective and efficient Visual information-seeking Systems with an added value for the user. Last, but nor least, the joy of use - as stated by various test persons - should ease the acceptance of new visualizations like the CSV. Getting the casual user to become accustomed to such visualizations by implementing them in every-day applications like Email or Web browsers is a first step in opening the world of information visualization to the wider public. Systems such as VisMeB and its successor MedioVis are good examples, evolving in the right direction.

A EVALUATION RESULTS

Results of the Evaluation. Missing times (user did not solve the task, or took more than 300 seconds.) were substituted by the mean value of ALL appropriate tasks - regardless of wether they were suitable (which is NOT the case for tasks A3, B1, B2, C1, C3!)[1]. All calculations were perfomed using SPSS.

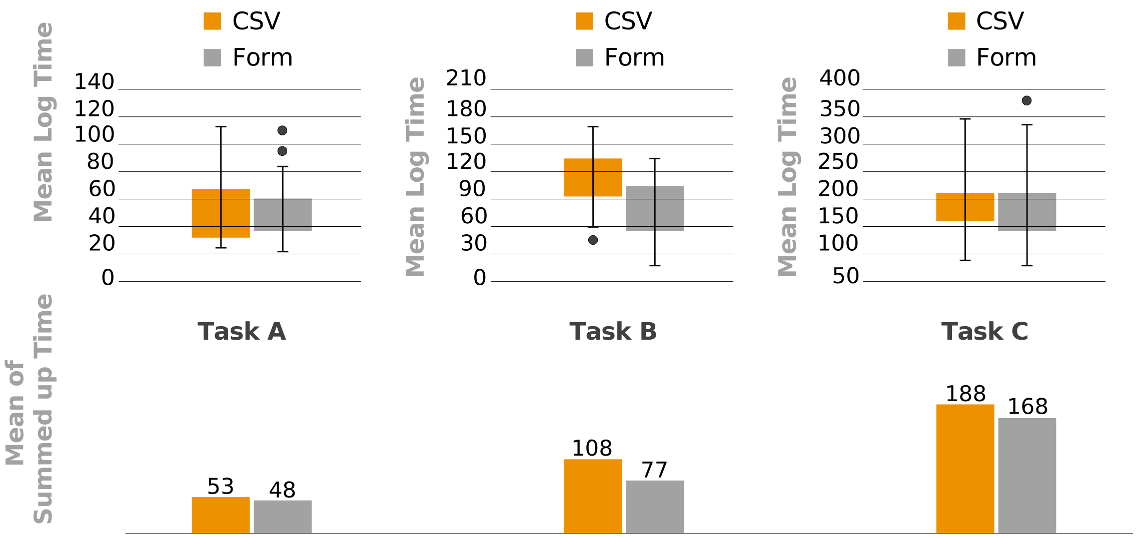


***Figure A.1**: Boxplots and arithmetic means of the raw data, summed over all tasks.*

[1]The Shapiro-Wilk test [SW65] for distributional adequacy failed on these

	N	Mean	Standard Deviation	Standard Error	95% confidence interval		Minimum	Maximum
					Limit (Min)	Limit (Max)		
CSV	20	52.9430	22.65699	5.06626	42.3392	63.5468	27.00	115.00
Form	18	47.9877	16.45954	3.87955	39.8025	56.1728	23.00	90.00
Total	38	50.5957	19.85873	3.22151	44.0683	57.1231	23.00	115.00

Table A.1: *ONEWAY descriptive statistics for Task 'A'. Unit used is seconds per task.*

	N	Mean	Standard Deviation	Standard Error	95% confidence interval		Minimum	Maximum
					Limit (Min)	Limit (Max)		
CSV	17	104,4299	29,80663	7,22917	89,1047	119,7550	43,00	159,00
Form	17	76,8992	27,40400	6,64645	62,8093	90,9890	29,00	115,00
Total	34	90,6645	31,46576	5,39633	79,6856	101,6434	29,00	115,00

***Table A.2**: ONEWAY descriptive statistics for Task 'B'.*

	N	Mean	Standard Deviation	Standard Error	95% confidence interval		Minimum	Maximum
					Limit (Min)	Limit (Max)		
CSV	20	187,9624	80,98566	18,10894	150,0600	225,8649	65,00	345,00
Form	20	181,6654	93,36762	20,87763	137,9680	225,3628	53,00	433,00
Total	40	184,8139	86,32736	13,64955	157,2051	212,4228	53,00	433,00

***Table A.3**: ONEWAY descriptive statistics for Task 'C'.*

	Sum Squared	df	Mean of Squares	F	Significance
Between Groups	7,155	1	7,155	0,13	0,909
Inside Groups	20434797	38	537,758		
Total	20441952	39			

Table A.4: *ONEWAY ANOVA - Task 'A': No significant difference between CSV and Form (sg.* $0.909 > 0.05$*); 95% Confidence Interval.*

	Sum Squared	df	Mean of Squares	F	Significance
Between Groups	6442,487	1	6442,487	7,860	0,009
Inside Groups	26230,623	32	819,707		
Total	32673,111	33			

Table A.5: *ONEWAY ANOVA - Task 'B': Significant difference between CSV and Form (sg.* $0.009 < 0.05$*); 95% Confidence Interval.*

	Sum Squared	df	Mean of Squares	F	Significance
Between Groups	396,518	1	396,518	0,52	0,821
Inside Groups	290247,598	38	7638,095		
Total	290644,116	39			

Table A.6: *ONEWAY ANOVA - Task 'C': No significant difference between CSV and Form (sg.* $0.821 > 0.05$*); 95% Confidence Interval.*

B PUBLICATIONS AND CONFERENCES

The CircleSegmentView - A Visualization for Query Preview and Visual Filtering
Klein, P., Reiterer, H.
Proceedings of SPIE Vol. 5669, IS&T / SPIE, San Jose, California (U.S.A.), Visualization and Data Analysis, p. 327-338, Jan 2005

Ein visueller Metadaten Browser für die explorative Erkundung groer Datenmengen
Reiterer, H., Limbach, T., Klein, P., Müller, F., Jetter, H.-C.
Mensch und Computer 2003 - Interaktion in Bewegung, in: Ziegler J. Szwillus, B.G. Teubner, Stuttgart, p. 165-176, Mensch & Computer 2003, Sep 2003

Scenario based Design of a Visual Meta-data Browser
Limbach, T., Reiterer, H., Klein, P., Müller, F.
Human-Computer Interaction Interact'03, in: Rauterberg, M., IOS Press, Amsterdam, p. 993-996, Human-Computer Interaction - Interact'03 IFIP, Sep 2003

Visualization and Interaction Techniques of the Visual Metadata Browser VisMeB
Müller, F., Klein, P., Limbach, T., Reiterer, H.
3rd International Conference on Knowledge Management, in: Maurer, H., Tochtermann, K., Knowcenter, Journal of Universal Computer Science, Graz, (Austria), p. 89-95, I-Know03, Proceedings of 3rd International Conference on Knowledge Management, Jul 2003

Metadata Visualization with VisMeB
Klein P., Reiterer H., Müller, F., Limbach, T.
7th International Conference on Information Visualisation 2003, in: Titsworth, F., IEEEComputer Society, "Los Alamitos, CA, USA", p. 600-605, IV03, 7th Intenational Conference on Information Visualization, Jul 2003

Visualizing Metadata: LevelTable vs. GranularityTable in the SuperTable/Scatterplot Framework
Limbach, T., Reiterer, H., Klein, P., Müller, F.
Human-Computer Interaction: Theory and Practice (Part II), in: Stephanidis, C., Lawrence Erlbaum Associates, New Jersey, USA, p. 1106-1110, HCI International, May 2003

Visualization of Metadata using the SuperTable+Scatterplot
Limbach, T., Müller, F., Klein, P., Reiterer, H., Eibl, M.
"Information und Mobilitt", in: Hammwhner R., Wolff C., Womser-Hacker C. , UKV Universittsverlag Konstanz, "Schriften zur Informationswissenschaft", Konstanz, p. 147-163, "8. Internationales Symposium für Informationswissenschaft ", Sep 2002

Visual Information Retrieval with the SuperTable + Scatterplot
Klein P., Müller F., Reiterer H., Eibl M.
"Information Visualisation - IV02", in: "Williams, A.", IEEE, Los Alamitos, CA, USA, p. 70-75, 6th International Conference on Information Visualisation, Jul 2002

Ranking Mechanisms in Meta-data Information Systems for Geo-spatial Data
Gbel, S., Klein P.
Proceedings of EoGeo 2002 - Conference, May 2002, Ispra (I)

C Contents of the CD-ROM

***Table C.1**: Contents of the CD-ROM*

Folder	**Contents**
Thesis	This thesis as a PDF
Source Code	The source-code of the CircleSegmentView
Video	Video Demonstration of VisMeB
VisMeB	Compiled Version for the Windows OS
Publications	My publications, so far

D Usability Evaluation - Tasks

Questions for the form (translated from german - the originals can be found in [Bey04]):

A1 How many comedies contains the movie database?

A2 Locate two french comedies and read their titles.

A3 What is the total amount of french comedies in the movie database?

B1 Please use the movie database to determine the release date of the movie 'Road trip'

B2 How many different genres exit in the movie database?

B3 Are there any action movies with release year 1987 or 1988 in the movie database?

C1 Which genre contains the most movies?

C2 Which decade offers the fewest movie releases?

C3 Please compare the two genres comedy and thriller. Which genre has more movies ranked '20'?

Questions for the CSV (translated from german - the originals can be found in [Bey04]):

A1 How many science-fiction movies contains the movie database?

A2 Locate two english dramas and read their titles.

A3 What is the total amount of german comedies in the movie database?

B1 Please use the movie database to determine the release date of the movie 'American Pie'

B2 How many different languages exit in the movie database?

B3 Are there any thriller movies with release year 1987 or 1988 in the movie database?

C1 In which language are most of the movies released?

C2 Which decade offers most of the movie releases?

C3 Please compare the two genres science-fiction and drama. Which genre has more movies ranked '20'?

E BIBLIOGRAPHY

Bibliography

[Agu67] AGUILAR, F. J.: *Scanning the Business Environment*. 1. Macmillan Co, New York, 1967

[AS94a] AHLBERG, C. ; SHNEIDERMAN, B.: The Alphaslider: A Compact and Rapid Selector. In: *Proceedings of the CHI '94 Conference*, ACM Press, 1994, pp. 365–371

[AS94b] AHLBERG, Christopher ; SHNEIDERMAN, Ben: Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In: *Human Factors in Computing Systems. Conference Proceedings CHI'94*, 1994, pp. 313–317

[AS94c] AHLBERG, Christopher ; SHNEIDERMAN, Ben: Visual information seeking using the FilmFinder. In: *Conference companion on Human factors in computing systems*, ACM Press, 1994. – ISBN 0–89791–651–4, pp. 433–434

[AW95] AHLBERG, C. ; WISTRAND, E.: IVEE: an Information Visualization and Exploration Environment. In: *Proceedings of the 1995 IEEE Symposium on Information Visualization*, IEEE Computer Society, 1995. – ISBN 0–8186–7201–3, pp. 66

[Bat85] BATES, M. J.: An exploratory paradigm for online information retrieval. In: *Proceedings of the sixth international research forum in information science (IRFIS 6), Frascati, Italy, September 16-18*, 1985, pp. 91–99

[Bat02] BATES, M. J.: Toward an integrated Model of Information Seeking and Searching. In: *The fourth International Conference on Information Needs, Seeking and Use in Different Contexts - Keynote, Lisbon, Portugal*, 2002

[BC88] BECKER, R. A. ; CLEVELAND, W. S.: Brushing Scatterplots. In: CLEVELAND, W. S. (Hrsg.) ; MCGILL, M.E. (Hrsg.): *Dynamic Graphics for Statistics*, Wadsworth, 1988, pp. 201–224

[Bey04] BEY, A.: *Der CircleSegmentView - Design und Evaluation einer Query Preview*, University of Konstanz, Bachelor Thesis, 2004

[BH94] BEDERSON, Benjamin B. ; HOLLAN, James D.: Pad++: a zooming graphical interface for exploring alternate interface physics. In: *Proceedings of the 7th annual ACM symposium on User interface software and technology*, ACM Press, 1994. – ISBN 0–89791–657–3, pp. 17–26

[BHN+98] BEST, C. ; HAYES, L. ; NEBERT, D. ; REICH, L. ; PERCIVAL, G. ; HYLTON, S.: Alignment of CIP and GEO Profiles. In: *Proceedings EOGEO'98*, 1998

[BS97] BEARD, K. ; SHARMA, V: Multidimensional Ranking in Digital Spatial Libraries. In: *Special Issue of Meta-data Journal of Digital Libraries* Bd. 1, 1997, pp. 153–160

[Buc92] BUCKLAND, M.K.: Agenda for online catalog designers. In: *Information Technology and Libraries* 11 (1992), Nr. 2, pp. 157–163

[BYRN99] BAEZA-YATES, R. ; RIBEIRO NETO, B.: *Modern Information Retrieval*. 1. Addison Wesley (ACM Press), 1999. – 257–324 S

[CCCL01] CHEN, Y. ; CHEN, S. ; CHANG, Y. ; LIN, S. C. *A Metadata Lifecycle for Content Analysis in Digital Libraries*. 2001

[CD00] CHEN, H. ; DUMAIS, S.: Bringing order to the Web: Automatically Categorizing Search Results. In: *Proceedings of the CHI 2000 - Conference on Human Factors in Computing Systems*, 2000, pp. 145–152

[CDT99] CHOO, W. ; DETLOR, B. ; TUMBULL, D.: Information-seeking on the Web: An integrated model of browsing and searching. In: *Proceedings of the 62nd Annual Meeting of the American Society for Information Science*, 1999, pp. 3–16

[CM99] CARD, S.K. ; MACKINLAY, B.: Readings in Information Visualization. In: *Using Vision to Think*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1999

[DCC+03] DUMAIS, Susan ; CUTRELL, Edward ; CADIZ, JJ ; JANCKE, Gavin ; SARIN, Raman ; ROBBINS, Daniel C.: Stuff I've seen: a system for personal information retrieval and re-use. In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press, 2003. – ISBN 1–58113–646–3, pp. 72–79

[DDL+90] DEERWESTER, Scott C. ; DUMAIS, Susan T. ; LANDAUER, Thomas K. ; FURNAS, George W. ; HARSHMAN, Richard A.: Indexing by Latent Semantic Analysis. In: *Journal of the American Society of Information Science* 41 (1990), Nr. 6, pp. 391–407

[DDM86] DICKSON, G.W. ; DESANCTIS, G. ; MCBRIDE, D.J.: Understanding the effectiveness of computer graphics for decision support - a cummulative experimental approach. In: *Communications of the ACM* Bd. 29, 1986, pp. 40–47

[DPSB97] DOAN, K. ; PLAISANT, C. ; SHNEIDERMAN, B. ; BRUNS, T.: Query previews for networked information systems: A case study with NASA environmental data. In: *SIGMOD Record (ACM Special Interest Group on Management of Data)* 26 (1997), ????, Nr. 1, pp. 75–??

[DW84] DAFT, Richard L. ; WEICK, Karl E.: Toward a Model of Organizations as Interpretation Systems. In: *Academy of Management Review* 9 (1984), Nr. 2, pp. 284–295

[ECH93] ELLIS, D. ; COX, D. ; HALL, K.: A Comparison of the Information Seeking Patterns of Researchers in the Physical and Social Sciences. In: *Journal of Documentation* 4 (1993), Nr. 49, pp. 356–369

[EH97] ELLIS, David ; HAUGAN, Merete: Modelling the Information Seeking Patterns of Engineers and Research Scientists in an Industrial Environment. In: *Journal of Documentation* 53 (1997), Nr. 4, pp. 384–403

[Ell89] ELLIS, D.: A Behavioural Model for Information Retrieval System Design. In: *Journal of Information Science* 15 (1989), Nr. 4, pp. 237–247

[Fei03] FEITELSON, D.G.: Comparing Partitions with Spie Charts / Computer Science and Engineering, The Hebrew University of Jerusalem. 2003. – Research Paper. Techreport

[Fit54] FITTS, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. In: *Journal of Experimental Psychology* Bd. 47, 1954, pp. 381–391

[Fra88] FRANK, G. H.: Reflections on NoteCards: seven issues for the next generation of hypermedia systems. In: *Commun. ACM* 31 (1988), Nr. 7, pp. 836–852. – ISSN 0001–0782

[FS95] FISHKIN, Kenneth P. ; STONE, Maureen C.: Enhanced Dynamic Queries via Movable Filters. In: *CHI*, 1995, pp. 415–420

[Fur81] FURNAS, G.W.: The FISHEYE view: a new look at structured files / Bell Laboratories. 1981. – Technical Memorandum #81-11221-9. -

[GHJ02] GÖBEL, Stefan ; HAIST, Jörg ; JASNOCH, Uwe: GeoCrystal: graphic-interactive access to geodata archives. In: ERBACHER, Robert F. (Hrsg.) ; CHEN, Philip C. (Hrsg.) ; GROEHN, Matti (Hrsg.) ; ROBERTS, Jonathan C.

(Hrsg.) ; WITTENBRINK, Craig M. (Hrsg.): *Proceedings of VDA 2002* Bd. 4665, SPIE, 2002, pp. 391–402

[GHJV93] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph E. ; VLISSIDES, John M.: Design Patterns: Abstraction and Reuse of Object-Oriented Design. In: *ECOOP '93: Proceedings of the 7th European Conference on Object-Oriented Programming*, Springer-Verlag, 1993. – ISBN 3–540–57120–5, pp. 406–431

[GJ01a] GÖBEL, S. ; JASNOCH, U.: Development of a semantic network for environmental Geo-data. In: LORENZ M. HILTY, Paul W. G. (Hrsg.): *Published in Sustainability in the Information Society, 15th International Symposium Informatics for Environmental Protection, Zurich*, Metropolis Verlag, Marburg, 2001. – ISBN 3–89518–370–9, pp. 679–684

[GJ01b] GÖBEL, S. ; JASNOCH, U.: Meta-data Information Systems for Geo-spatial Data. In: CLAUS RAUTENSTRAUCH, Susanne P. (Hrsg.): *Published in Environmental Information Systems in Industry and Public Administration*, 2001. – ISBN 1–930708–02–5, pp. 283–296

[Göb02] GÖBEL, S.: *Intelligente graphisch interaktive Benutzerführung zu Geodatenarchiven*, Zentrum für graphische Datenverarbeitung, Diss., 2002

[Grü04] GRÜN, C.: *Entwicklung eines visuellen Metadaten Browsers für die Mediothek Konstanz*, University of Konstanz, Master Thesis, 2004

[HAB] HUDGINS, J. ; AGNEW, G. ; BROWN, E. *Getting mileage out of metadata: Applications for the library*

[HC98] HARTSON, H. R. ; CASTILLO, Jose C.: Remote evaluation for post-deployment usability improvement. In: *AVI '98: Proceedings of the working conference on Advanced visual interfaces*, ACM Press, 1998, pp. 22–29

[ID90] INSELBERG, Alfred ; DIMSDALE, Bernard: Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: *Proceedings of the 1st conference on Visualization '90*, IEEE Computer Society Press, 1990. – ISBN 0–8186–2083–8 (PAPER), pp. 361–378

[ISO96] ISO: ISO9241-10: Ergonomic requirements for office work with visual display terminals (VDTs) - Part 10: Dialog Principles / ISO. 1996. – Research Paper. n/a

[Jet03] JETTER, H.C.: *Usability-Evaluation im Rahmen von INVISIP*, University of Konstanz, Bachelor Thesis, 2003

[JGB00] JASNOCH, U. ; GÖBEL, S. ; BALFANZ, D.: InGeo IC: The portal to Geodata. In: *Proceedings e2000*, IOS Press, Amsterdam, 2000

[Jon94] JONES, T.: A Model of Landscapes / Santa Fe Institute. 1994 (94-02-002). – Working Papers. Available at http://ideas.repec.org/p/wop/safiwp/94-02-002.html

[Kam98] KAMADA, T.: *On visualization of abstract objects and relations*, University of Tokyo, Department of Information Science, Diss., October 1998

[Kei02] KEIM, Daniel A.: Information Visualization and Visual Data Mining. In: *IEEE Transactions on Visualization and Computer Graphics* 8 (2002), Nr. 1, pp. 1–8. – ISSN 1077–2626

[KG02] KLEIN, P. ; GÖBEL, S.: Ranking Mechanisms in Meta-data Information Systems for Geo-spatial Data. In: *Proceedings EOGEO 2002*, 2002

[KMLR03a] KLEIN, P. ; MÜLLER, F. ; LIMBACH, T. ; REITERER, H.: Scenario based design of a Visual Meta-data Browser. In: *Human Computer Interaction Interact'03*, Rauterberg, M., IOS Press Amsterdam, Sep 2003, pp. 993–996

[KMLR03b] KLEIN, P. ; MÜLLER, F. ; LIMBACH, T. ; REITERER, H.: Visualization and Interaction Techniques of the Visual Metadata Browser VisMeB. In: *3rd International Conference on Knowledge Management*, 2003, pp. 89–95

[KMLR03c] KLEIN, P. ; MÜLLER, F. ; LIMBACH, T. ; REITERER, H.: Ein visueller Metadatenbrowser für die explorative Erkundung großer Datenmengen. In: ZIEGLER, J. (Hrsg.) ; SZWILLUS, B.G. (Hrsg.): *Mensch & Computer 2003 - Interaktion in Bewegung*, Teubner, Stuttgart, Sep 2003, pp. 165–176

[KMRE02] KLEIN, P. ; MUELLER, F. ; REITERER, H. ; EIBL, M.: Visual Information Retrieval with the SuperTable+Scatterplot. In: *Proceedings of the 6th International Conference on Information Visualization (IV 02)*, 2002, pp. 59–66

[KMRL03] KLEIN, P. ; MÜLLER, F. ; REITERER, H. ; LIMBACH, T.: Metadata Visualization with VisMeB. In: *Proceedings of the 7th International Conference on Information Visualization (IV 03)*, 2003, pp. 600–605

[KR05] KLEIN, P. ; REITERER, H.: The CircleSegmentView - A Visualization for Query Preview and Visual Filtering. In: *Proceedings of SPIE Vol. 5669, IS&T / SPIE, San Jose, California (U.S.A.), Visualization and Data Analysis*, 2005, pp. 327–338

[Kuh03] KUHLEN, R.: Change of Paradigm in Knowledge Management - Framework for the Collaborative Production and Exchange of Knowledge. In: *Plenary Session, 3rd August 2003, of the Worlds Lbrary and Information Congress: 69th IFLA General Conference and Council, Berlin*, 2003

[Kwa92] KWASNIK, B. H.: Descriptive Study of the Functional Components of Browsing. In: *Proceedings of the IFIP TC2-WG2.7 working conference on engineering for human-computer interaction, August 10-14, 1992, Elivuoi, Finland*, 1992

[LKMR03] LIMBACH, T. ; KLEIN, P. ; MÜLLER, F. ; REITERER, H.: Visualising Metadata: Level- vs. Granularity Concept in the SuperTable/Scatterplot Framework. In: *HCI International 2003*, 2003

[Mac95] MACEACHREN, A.M.: *How maps work*. 1. The Guilford Press, New York, 1995

[Man02] MANN, T.: *Visualization of Search Results from the World Wide Web*, University of Konstanz, Department of Computer and Information Science, Diss., 2002. – http://www.ub.unikonstanz.de/kops/volltexte/2002/751/

[Mar95] MARCHINONINI, G. M.: *Information Seeking in Electronic Environments*. 1. Cambridge University Press, 1995

[Mar00] MARCO, D. ; ELLIOT, R.M. (Hrsg.): *Building and Managing the Meta Data Repository - A full Lifecycle Guide*. 1. John Wiley and Sons, Inc., 2000

[May99] MAYHEW, Deborah J.: *The usability engineering lifecycle: a practitioner's handbook for user interface design*. 1. Morgan Kaufmann Publishers Inc., 1999. – ISBN 1–55860–561–4

[Mey96] MEYER, J.A.: *Visualisierung im Management*. 1. Deutscher Universitätsverlag, 1996. – ISBN 3–8244–0323–4

[Mos03] MOSTAFA, J. *Workshop - Information Visualization for Retrieval and Analysis (IVIRA)*. 2003

[MR99] MANN, T.M. ; REITERER, H.: Case Study: A Combined Visualization Approach for WWW-Search Results. In: *Proceedings IEEE Symposium on Information Visualization.*, 1999

[Mül05] MÜLLER, F.: *VisMeB*, University of Konstanz, Department of Computer and Information Science Germany, Diss., March (ETA) 2005

[Muß02] MUSSLER, G.: *Retrieving Business Information from the WWW*, University of Konstanz, Germany, Diss., 2002

[Nie97] NIELSEN, J.: Search and You May Find. 1997. – Research Paper. Available at http://www.useit.com/alertbox/9707b.html

[NM94] NIELSEN, J. ; MACK, R.L.: *Usability Inspection Methods*. 1. John Wiley & Sons, New York, 1994

[Noi94] NOIK, E.: A space of Presentation Emphasis Techniques for Visualizing Graphs. In: *Proceedings of Graphics Interface '94*, 1994, pp. 225–233

[NTK98] NIKOLAI, R. ; TRAUPE, A. ; KRAMER, R.: Thesaurus Federations: A Framework for the Flexible Integration of Heterogeneous, Autonomous Thesauri. In: *Proceedings ADL'98*, 1998

[PH96] POLLOCK, A. ; HOCKLEY, A.: What's Wrong with Internet Searching. In: *Designing for the Web: Empirical Studies*, 1996

[Poo85] POOLE, H. L.: *Theories of the middle range*. 1. Norwood, N.J.: Ablex, 1985

[Pre99] PREIM, B.: *Entwicklung interaktiver Systeme*. 1. Springer, 1999

[Pre00] PRESTIPINO, M. *Multidimensional Visualization (Term Paper)*. 2000

[Pro04] PROJECT, BASELINE. *Glossary of Terms used in Usability Engineering*. 2004

[PVN+99] PLAISANT, Catherine ; VENKATRAMAN, Maya ; NGAMKAJORNWIWAT, Kawin ; BARTH, Randy ; HARBERTS, Bob ; WENLANNG, W.: Refining Query Previews Techniques for Data with Multivalued Attributes: The Case of NASA EOSDIS. In: *Advances in Digital Libraries*, 1999, pp. 50–59

[RC02] ROSSON, M.B. ; CARROLL, J.M.: *Usability Engineering. Scenario-Based Development of Human-Computer Interaction*. 1. Academic Press, San Diego, U.S.A., 2002

[Rex05] REXHAUSEN, S.: *LSI for small Texts with Homogenous Vocabulary*, University of Konstanz, Bachelor Thesis, 2005

[RH94] RUBIN, Jeffrey ; HUDSON, Theresa: *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. 1. John Wiley & Sons, Inc., 1994. – ISBN 0471594032

[RMM01] REITERER, H. ; MUSSLER, G. ; MANN, Thomas M.: A visual information seeking system for Web search. In: *Mensch & Computer 2001. Conference: Bad Honnef, Germany, March 5-8 2001. Stuttgart*, 2001, pp. 97–306

[RMMH00] REITERER, H. ; MUSSLER, G. ; MANN, Thomas M. ; HANDSCHUH, S.: INSYDER - An Information Assistant for Business Intelligence. In: *Proceedings of the 23 Annual International ACM SIGIR 2000 Conference on Research and Development in Information Retrieval*, ACM Press, 2000, pp. 112–119

[Ros02] ROSENFELD, P.: *Information Architecture for the World Wide Web*. 2. O'Reilly and Associates, 2002. – ISBN 0596000359

[Sal89] SALTON, G.: *Automatic Text Processing*. 1. Addison-Wesley, 1989

[Shn04] SHNEIDERMAN, Ben: *Designing the user interface*. 4. Addison-Wesley, 2004

[SJWS02] SPINK, Amanda ; JANSEN, Bernard J. ; WOLFRAM, Dietmar ; SARACEVIC, Tefko: From E-Sex to E-Commerce: Web Search Changes. In: *Computer* 35 (2002), Nr. 3, pp. 107–109. – ISSN 0018–9162

[Spe01] SPENCE, R.: *Information Visualization*. 1. Addison Wesley (ACM Press), 2001. – 127–128 S

[Spo93] SPOERRI, A.: InfoCrystal: A Visual Tool for Information Retrieval. In: *Proceedings IEEE Visualization'93*, 1993, pp. 150–157

[Sta95] STANTON, N.: Content Analysis: A methodology for hypermedia design. In: *IEE Colloquium on Authoring and Application of Hypermedia-based User Interfaces*, 1995

[Ste03] STEPHENS, R. T. *Knowledge: The Essence of Meta:Meta Data Management Life Cycle Reviewed.* 2003

[SW65] SHAPIRO, S. ; WILK, M.B.: An analysis of variance test for normality. In: *Biometrika* Bd. 52, 1965, pp. 591–611

[Tan01] TANIN, Egemen: *Browsing large online data using generalized query previews*, University of Maryland, Diss., 2001. – Director-Ben A. Shneiderman

[Tho33] THOMAS, K.C.: The ability of children to interpret graphs. In: *The teaching of geography* Bd. 32, 1933

[TPS00] TANIN, E. ; PLAISANT, C. ; SHNEIDERMAN, B.: Browsing Large Online Data with Query Previews. In: *Proceedings of the Symposium on New Paradigms in Information Visualization and Manipulation (NPIVM) 2000*, ACM Press, 2000

[Tuf90] TUFTE, Edward: *Envisioning information*. 1. Graphics Press, 1990. – ISBN 0–9613921–1–8

[TzuBC] TZU, S.: *The art of war*. 1. UNESCO Collection of representative works: Chinese series., 500 B.C.

[War00] WARE, C.: *Information Visualization - Perception for Design*. 1. Morgan Kaufman, 2000

[WD83] WEICK, Karl E. ; DAFT, Richard L.: The Effectiveness of Interpretation Systems. In: *Organizational Effectiveness: A Comparison of Multiple Models* (1983), pp. 71–93

[Wil96] WILLS, Graham J.: 524,288 Ways to Say: This is Interesting. In: *Proceedings of the IEEE Symposium on Information Visualization*, IEEE Computer Society Press, 1996, pp. 54–61

[Wil97a] WILLE, R.: Conceptual Landscapes of Knowledge: A Paradigm for Knowledge Processing. In: *Proceedings KRUSE'97*, 1997, pp. 2–13

[Wil97b] WILSON, T. D.: Information Behaviour: An Interdisciplinary Perspective. In: *Information Processing and Management* 33 (1997), Nr. 4, pp. 551–572

[WS92] WILLIAMSON, Christopher ; SHNEIDERMAN, Ben: The dynamic HomeFinder: evaluating dynamic queries in a real-estate information exploration system. In: *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press, 1992. – ISBN 0–89791–523–2, pp. 338–346

[WWK00] WANG BALDONADO, Michelle Q. ; WOODRUFF, Allison ; KUCHINSKY, Allan: Guidelines for using multiple views in information visualization. In: *Proceedings of the working conference on advanced visual interfaces*, ACM Press, 2000. – ISBN 1–58113–252–2, pp. 110–119

[YWB03] YAMAUCHI, Yutaka ; WHALEN, Jack ; BOBROW, Daniel G.: Information use of service technicians in difficult cases. In: *Proceedings of the conference on human factors in computing systems*, ACM Press, 2003. – ISBN 1–58113–630–7, pp. 81–88

F INDEX

Index