Granularity Based Multiple Coordinated Views to Improve the Information Seeking Process

Dissertation
zur Erlangung des akademischen Grades
des Doktors der Naturwissenschaften (Dr.rer.nat.)
Fachbereich Informatik u. Informationswissenschaft
der Universität Konstanz, Mathematisch-Naturwissenschaftliche Sektion

vorgelegt von

Frank Müller

Diese Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Konstanz, den 21. Dezember 2005
Acknowledgment

When I started to work at the University of Konstanz in the summer of 2000 I was con-
fronted with completely new working areas like teaching students, preparing presentations
for international conferences and meetings, and fighting with administrative difficulties.
Nevertheless, the prime area of my interest was research in the field of Information Vi-

sualization and Human-Computer Interaction. Therefore, I have to thank my supervisor,
Prof. Dr. Harald Reiterer. He made these research disciplines accessible to me and in-
spired as well as supported me during the whole project. I learned a lot while working in
his Human-Computer Interaction group. Thanks are also due to Prof. Dr. Daniel Keim
and Prof. Dr. Ronald Hübner for taking the jobs of referees.

From my colleagues I have to thank especially Peter Klein, Tobias Limbach, Georg
Odenthal, Torsten Grust, and Ulrik Brandes. They supported me by advice, inspiration,
cooperation, and lively conversations and meetings - at working hours and in leisure time.

Thanks to all the members of the INVISIP project team, especially Jörg Haist and
Stefan Göbel, who had a very large part in the success of the project. It was a pleasure to
work with you!

Special thanks are also due to Fredrik Gundelsweiler, Thomas Memmel, Christian
Grün, Werner König, Philipp Liebrenz, and Jens Gerken for spending a lot of time and
effort implementing and evaluating the VisMeB framework.

I want to thank my family for supporting me all the years, especially in hard times. I
don’t know if I could have done it without your help. Thanks Mom, Grandma and espe-
cially Grandpa. I am very sorry that you don’t live to see me getting my doctorate!

Last but not least, I want to thank Sarah for being on my side, giving me hope and
bearing all my moods.


Der zweite Teil behandelt das Thema Usability Engineering oder genauer gesagt den so- genannten User Centered Design Process. Die Entwicklung von VisMeB folgt diesem Prozess was sich in frühen Benutzertests widerspiegelt die für wichtige Designfragen entscheidend sind. Durch diese frühe Einbindung können Fehler schon früh in der Entwicklung erkannt und vermieden werden. Der Einführung in den Bereich Usability Engi-
neering folgt eine Darstellung der verschiedenen Entwicklungsphasen die VisMeB durchlaufen hat. Die Verwendung von Prototypen spielt hierbei eine wichtige Rolle, was sich in Verbindung mit Benutzertests in den positiven Ergebnissen der Untersuchungen zeigt. Zusammenfassung und Ausblick bilden den Abschluss dieser Dissertation.
This thesis introduces a new concept for visualizing search results from database inquiries. Techniques that are already known and proven are combined in a way that emphasizes the advantages and eclipses the drawbacks of individual features. For this purpose Multiple Coordinated Views and a Granularity Concept based on the idea of a semantic zoom were unified. The approach is not restricted to a specific domain and the visualizations used can be easily adapted. The concept is implemented within the VisMeB framework, a Java-based "Visual Metadata Browser" that is available in diverse versions.

Two main disciplines guide this thesis - Information Visualization and Usability Engineering. Thus, the presented work adheres to this division. The first part of the thesis deals with Information Visualization in general and gives an overview of the interaction techniques used and applications that provided inspiration. This progresses to a detailed description of the multiple coordinated views implemented and the granularity concept. Because the use of multiple coordinated views offers advantages as well as drawbacks, it is necessary to clarify a) whether to use them at all, b) if yes, which visualizations to choose, and c) how the layout and the interaction are defined, which leads to the three-phase model introduced and described in this work the first time. Furthermore, three differently-implemented granularity versions are introduced - the TableZoom, the RowZoom, and the CellZoom, which have a strong influence on the display and user interaction. A detailed description of interaction techniques between the multiple coordinated views subject to the granularity concept closes the first part of the thesis.

The second part deals with Usability Engineering or, more precisely, the User-Centered Design Process. The development of VisMeB follows the user-centered design process, which results in early user tests that are responsible for important design decisions. This leads to an enormous advantage compared to systems that did not involve users during the development. After an introduction into the field of usability engineering, the different development steps of VisMeB are considered. Prototyping played an important role and, in combination with user tests, the design process was guided by the results of these investigations.

An outlook and a conclusion brings this thesis to a close.
## Contents

1 Introduction 1
   1.1 Problem Space 1
   1.2 Methods of Resolution 2
   1.3 Thesis Structure 4

2 State of the Art 7
   2.1 Information Visualization - An Introduction 7
      2.1.1 Visualization Reference Model 11
      2.1.2 Meta-data 13
      2.1.3 Interactive Techniques 14
         2.1.3.1 Dynamic Queries 14
         2.1.3.2 Brushing & Linking 15
         2.1.3.3 Movable Filters 16
         2.1.3.4 Overview-Plus-Detail 17
         2.1.3.5 Focus & Context 18
         2.1.3.6 Panning & Zooming 19
   2.2 Scatterplots 22
      2.2.1 2D-Scatterplots 23
         2.2.1.1 Geographical Scatterplots 23
         2.2.1.2 Scatterplots for Abstract Data 25
         2.2.1.3 Scatterplot Matrices 29
      2.2.2 3D-Scatterplots 33
   2.3 Semantic Similarity Maps 37
      2.3.1 Galaxies 41
      2.3.2 Themescapes 42
      2.3.3 Kohonen Maps 48
   2.4 Table-Based Visualizations 49
   2.5 Semantic Zoom 50
   2.6 Multiple Coordinated Views 55
      2.6.1 MCVs Realizing The Select ⇔ Select Relationship 56
      2.6.2 MCVs Realizing The Navigate ⇔ Navigate Relationship 60
## 2.6.3 MCVs Realizing The Select ⇔ Navigate Relationship 63

## 2.7 Summary 65

### 3 Multiple Coordinated Views 71

#### 3.1 Introduction 71

#### 3.2 Phase 1: Acceptance Decision 72

#### 3.3 Phase 2: Choice of Visualizations 73

#### 3.4 Phase 3: Usage, Interaction and Layout 79

##### 3.4.1 Coordination Models 80

###### 3.4.1.1 Snap Conceptual Model 80

###### 3.4.1.2 View Coordination Architecture 81

###### 3.4.1.3 Coordination Model for Exploratory Visualization 82

###### 3.4.1.4 The VisMeB Conceptual Model 84

#### 3.5 The VisMeB Framework 86

##### 3.5.1 SuperTable 91

###### 3.5.1.1 Short History of the MediaGrid 91

###### 3.5.1.2 LevelTable 92

###### 3.5.1.3 GranularityTable 94

###### 3.5.1.4 GridTable 96

###### 3.5.1.5 MediaGrid 96

##### 3.5.2 Detailed History of the MediaGrid 100

#### 3.6 Summary 104

### 4 The Granularity Concept 105

#### 4.1 Zoom Introduction 105

#### 4.2 Taxonomy of zooming behavior 111

#### 4.3 The Semantic Zoom 112

#### 4.4 Degree of Interest 113

#### 4.5 Granularity Zoom Variants 116

##### 4.5.1 TableZoom 117

##### 4.5.2 RowZoom 117

##### 4.5.3 CellZoom 119

#### 4.6 Summary 122

### 5 Interaction Between Views 125

#### 5.1 Introduction 125

#### 5.2 Taxonomy of Interaction Techniques for Multiple Coordinated Views 126

##### 5.2.1 Select ⇔ Select 127

##### 5.2.2 Navigate ⇔ Navigate 127

##### 5.2.3 Select ⇔ Navigate 127

#### 5.3 TableZoom 128

##### 5.3.1 Coordination with GOViews 128

##### 5.3.2 Coordination with Textual Views 129
CONTENTS

5.4 RowZoom .............................................. 132
  5.4.1 Coordination with GOViews ...................... 132
  5.4.2 Coordination with Textual Views ................ 133
5.5 CellZoom ............................................. 136
  5.5.1 Coordination with GOViews ...................... 136
  5.5.2 Coordination with Textual Views ................ 138
5.6 Filter ................................................ 140
  5.6.1 Dialog Box ........................................ 140
  5.6.2 Circle Segment View .............................. 141
  5.6.3 Movable Filter .................................... 141
5.7 Benefits and Shortcomings ........................... 145
5.8 Summary .............................................. 149

6 User centered design process 151
  6.1 Interaction Design .................................. 154
    6.1.1 Requirements Analysis ............................ 156
    6.1.2 Design and Evaluation ............................ 158
  6.2 Development Stages of VisMeB / MedioVis .......... 163
    6.2.1 Paper-based Mockup ............................... 164
    6.2.2 HTML Mockup ...................................... 166
    6.2.3 Java Prototypes ..................................... 170
      6.2.3.1 The VisMeB Prototype ......................... 170
      6.2.3.2 The MedioVis Prototype ....................... 173
  6.3 Summary .............................................. 174

7 Evaluation List vs. LevelTable 177
  7.1 Introduction ......................................... 177
  7.2 Test Setting ......................................... 179
  7.3 Pretest ............................................... 179
  7.4 Test Persons ......................................... 179
  7.5 Entry Questionnaire .................................. 179
  7.6 Maintest ............................................. 180
    7.6.1 Tasks ............................................. 181
    7.6.2 Type of Data ....................................... 181
    7.6.3 Type of User ....................................... 181
    7.6.4 Technical Environment ............................ 182
    7.6.5 Training ........................................... 183
  7.7 Posttest .............................................. 183
  7.8 Test Results ......................................... 183
    7.8.1 Test1 - Baseline Test ............................. 183
    7.8.2 Test2 - Main Test .................................. 186
      7.8.2.1 Results concerning the total time of task completion . 186
      7.8.2.2 Results concerning the time for completion of single tasks 188
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.8.3 Posttest Results</td>
<td>193</td>
</tr>
<tr>
<td>7.9 Summary</td>
<td>194</td>
</tr>
<tr>
<td><strong>8 Outlook</strong></td>
<td><strong>195</strong></td>
</tr>
<tr>
<td>8.1 Similarity</td>
<td>195</td>
</tr>
<tr>
<td>8.2 Dimension of Interest</td>
<td>198</td>
</tr>
<tr>
<td><strong>9 Conclusion</strong></td>
<td><strong>201</strong></td>
</tr>
<tr>
<td><strong>A Appendix</strong></td>
<td><strong>213</strong></td>
</tr>
<tr>
<td>A.1 Semantic Similarity Maps</td>
<td>213</td>
</tr>
<tr>
<td>A.2 VisMeB Visualizations</td>
<td>219</td>
</tr>
<tr>
<td>A.2.1 ScatterPlot-2D</td>
<td>219</td>
</tr>
<tr>
<td>A.2.2 ScatterPlot-3D</td>
<td>219</td>
</tr>
<tr>
<td>A.2.2.1 MultiDataPointView</td>
<td>220</td>
</tr>
<tr>
<td>A.2.3 BrowserView</td>
<td>221</td>
</tr>
<tr>
<td>A.2.4 DocumentUniverse</td>
<td>222</td>
</tr>
<tr>
<td>A.2.5 LocationMap</td>
<td>222</td>
</tr>
<tr>
<td>A.2.6 Visual Configurator</td>
<td>222</td>
</tr>
<tr>
<td><strong>B Evaluation Tasks</strong></td>
<td><strong>225</strong></td>
</tr>
<tr>
<td>B.1 Pre-Test Fragebogen</td>
<td>225</td>
</tr>
<tr>
<td>B.2 Performance Test Ergebnisse</td>
<td>225</td>
</tr>
<tr>
<td>B.3 Post-Test Fragebogen</td>
<td>229</td>
</tr>
<tr>
<td>B.4 Pre-Test Questionnaire</td>
<td>230</td>
</tr>
<tr>
<td>B.5 Main Test Performance Results</td>
<td>233</td>
</tr>
<tr>
<td>B.6 Post-Test Questionnaire</td>
<td>235</td>
</tr>
<tr>
<td><strong>C Bibliography</strong></td>
<td><strong>237</strong></td>
</tr>
<tr>
<td><strong>D Publications and Conferences</strong></td>
<td><strong>253</strong></td>
</tr>
</tbody>
</table>
1

1.1 Problem Space

Nowadays, the amount of information surrounding us increases from day to day. The World Wide Web is a typical example nearly everyone is familiar with, whether it be at our everyday work or in our leisure time. At any time, new websites emerge and the opportunity to gain more knowledge is given. But are we able to deal with this variety or are we getting more and more "lost in information space"? There is no apparent limit to the spread of the Web. Hard disks, for example, have reached storage sizes of many gigabytes and therefore entrap the user into storing more and more data. Thus, we are confronted with an enormous data flood that has to be administered - and, most important, used - in an efficient way. To handle this problem, different methods of resolution are available. One kind of meaningful data management is to store interrelated data in a database. Depending on the way a database is used, it is possible to store nearly any kind of data. Examples are a database containing a CD collection (e.g. audio CDs for private use or software CDs for business use), an address book (for private or business connections), Web content (e.g. for use in a content management system), and so on. The list could be extended ad infinitum. In this context meta-data - which can be described as "data about data" - play an important role in defining, locating and exploring data. Standards for different domains are given by e.g. the Dublin Core meta-data standard for web documents or the ISO 19115 standard for geo meta-data. These topics are referred to in Chapter 2. For the moment, a typical scenario is described to help understand the usefulness of meta-data in their day-to-day usage.

Scenario 1:
Tim Herzog is a 34-year old unmarried man who has worked in a site planning bureau in Hannover for about one and a half years as an architect. His latest task is to conceptualize a new shopping mall in the center of Braunschweig. Therefore, he is interested in maps concerning potential building lots. He has access to a geo meta-data base (meta-data concerning geographical data like maps) that gives an insight into data concerning available maps. When he wants to order a map he just has to write an email to the corresponding contractor and request the respective digital or paper version depending on
the desired field of application. Before placing an order, Tim has to decide which specific map he needs, or if he needs several maps. This is important because most probably the maps are not free or take some time to be delivered. Otherwise it would be easier to keep all maps locally and print them out as necessary. Because of his work Tim is confident with the different types of meta-data available, for example “resolution” (e.g. 1:2000), “format” (e.g. tiff), “reference date” (e.g. 1/5/2003), “location”, “price”, and so on. To find the appropriate data he enters the query terms “center” and “Braunschweig” into the available form fillin interface. He gets about 500 data sets relevant to this query and therefore has to restrict it depending on specific characteristics. Because of the shopping mall’s size Tim needs a specific resolution of the map. The site planning tool used in the office imposes a second characteristic, the format, with which the map’s format has to be compatible. As a third constraint he wants to get a map as cheaply as possible. He activates the filter dialog and makes the necessary settings. As a result of this limitation Tim gets three possible maps that fit his constraints. Thus, he orders all three to get a better insight into the potential building lots.

One assumption made in this scenario is neither self-evident nor the general rule. Tim had no problem in getting an overview of the result set and limiting the obtained data by a filter mechanism. In real life this often looks different. Usually, the set of data containing the meta-data is presented in a simple list presentation, like the one known from Google: First, the title is given, followed by some meta-data, maybe (or perhaps hopefully) like the ones presented above. What are the next steps in finding the correct data? Tim probably has to scan every list item for the characteristics he’s interested in. This approach is definitely very inefficient and not very favored by users (see also: [PH97], [Nie04b], or [ZE98]). A first solution could be to present the meta-data in a table. However, another problem arises: how to present the whole bulk of the information? Is it possible to display all the data simultaneously? If there is a large variety of meta-data and each meta-data is shown in a single column, different approaches are conceivable. A first solution is to make all columns as wide as necessary to read the whole content, which can lead to the situation in which the user is constrained to scroll horizontally, which should be avoided. As a variant, heightening the row instead of broadening it would create a lot of line breaks and make the text nearly unreadable. The third possibility is to make all columns fit onto one screen, which makes long entries unreadable. Another fact neglected so far is the lack of an overview. The user is not able to view all the data sets in correlation to compare them, or to find clusters or outliers.

1.2 Methods of Resolution

The scenario introduced above leads to the central point of this work - the goal we are aiming to achieve. This can be described as follows:
1.2 METHODS OF RESOLUTION

Goal:
To provide a self-contained information-seeking system to facilitate information access and data handling for meta-data.

This goal can be further specified by the following refinements:

- Support the exploration of large information spaces as well as visual data mining and the analysis of expressive data.
- Improve the information seeking process by including all steps from query formulation via result-set presentation and query reformulation up to processing all information selected by users.

The approach presented in this work is based on a set of important design decisions that influenced and controlled the whole development process. They solved or reduced the problems that are addressed above. These decisions were:

1. Combine a normal table presentation for result sets with adequate visualizations to support the exploration process.
2. Multiple visualizations support the user in finding relevant data, dependent on the task.
3. Using an overview and detail presentation of search results enables the user to detect patterns, outliers and clusters, and to explore interesting data sets simultaneously.
4. A reduced stimulus-overload can be achieved by using chunks.
5. Using a generic approach for the system enables tests and assignments in a variety of application domains.

All these points are very specific and concern just a single problem, but as a whole they lead to the global aim of this approach, which is "to improve the information-seeking process by enhanced usability". Usability can be defined as follows:

**Definition 1.1 (Usability (1))** Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process." [Nie04a]

Three factors are decisive for usability:

- effectiveness,
- efficiency, and
- subjective user satisfaction
which can be found in the definition of usability in [ISO98] (see definition 6.4 (Usability(2))).

Altogether, keeping in mind the design decisions as well as the global aim, this brings us to the approach that was decided on:

*To use multiple visualizations that react in a coordinated manner and divide the bulk of information into different levels of detail.*

The realization of this approach took place during the development of the VisMeB framework (*Visual Meta-data Browser*) that was part of the EC funded project INVISIP 1. A combination of various views adapted to the application domain and a concept for structuring and distributing the enormous amount of data over several stages or more precisely "levels of detail" was implemented as a fully operational Java system.

### 1.3 Thesis Structure

As a lead-in to the field of information visualization, Chapter 2 gives an introduction to the techniques that were used in the VisMeB system as well as a State-of-the-Art analysis of systems that influenced the development. This introduction is restricted only to facts that are directly connected to this thesis and will not give a complete overview of systems or techniques that are used in information visualization nowadays.

Chapter 3 presents the idea of multiple coordinated views (MCVs), its meaning and effect on the scenario. Three phases are shown that support the user in his decision if the usage of MCVs is meaningful in the current situation. Theoretical models to depict the method of view coordination are described, followed by a complete presentation of visualizations used in the VisMeB framework.

The concept of granularity, which is based on the idea of a semantic zoom (see 4.3), is described in Chapter 4. Zoom variants as well as zooming behavior are described to communicate the ideas that are hidden behind the concept of granularity. The semantic zoom takes a special place in this context because of its characteristics which exceed the simple magnification that zooming in general can produce. This is followed by a description of different zoom variants that are implemented in the approach presented here.

In Chapter 5 possible synchronization implementations will be specified in general, followed by a detailed description of the interactions realized between the VisMeB visualizations, including the effects caused by the usage of the granularity concept and its consequences on the views' structure. The possibility of interacting with all visualizations

---

1INVISIP: Information Visualization for Site Planning, funded by EC, 5th Framework of the IST Program, Project No. IST-2000-29640, www.invisip.de
using filters must not be neglected and is presented by means of various filter variants that were accomplished.

The user-centered design process that supported the complete development of the VisMeB framework can be found in Chapter 6. A general introduction to lay the basis for the field of interaction design is given, followed by the different development stages VisMeB went through. This ranges from simple paper-based mockups to the fully implemented Java system.

Chapter 7 enlarges on the preceding chapter with a detailed portrait of the evaluation of the finalized system or, more precisely, the adapted table visualization in comparison to a list presentation of search results that is usual nowadays. After a brief overview of statistical fundamentals, the single test steps are explained and the results obtained are presented.

The outlook in Chapter 8 provides ideas for further activities and developments that could lead to an improvement of the current system. This refers to new visualizations as well as evaluations that are indispensable for user-friendly and helpful applications.

Chapter 9 closes this thesis with a summary of the work presented.

A short overview of the main structure of this thesis, divided into Introduction, Information Visualization, Usability Engineering, and Outlook and Conclusion is displayed in Figure 1.1.
Figure 1.1: Main structure of the thesis
2.1 Information Visualization - An Introduction

Presenting search results for a specific query in a way that helps the user to explore the result set and find relevant data is the main task for information-seeking systems like VisMeB. The obvious question is: “Why use information visualization instead of pure text presentation when the base data are provided as text?” To answer this question, the term "Information Visualization" should first be defined:

Definition 2.1 (Information Visualization) is the use of computer-supported, interactive visual representations of abstract data in order to amplify cognition.

[CM99]

To avoid misunderstandings, a differentiation has to be made between "Information Visualization" and "Scientific Visualization", the latter being closely linked to Information Visualization, but the application domain is engaged in scientific i.e. mostly physically-based, not abstract, data. An example is the three-dimensional model of an engine block, rotatable in all directions, which is especially important for an engine-development process. In contrast, Information Visualization deals with abstract data, e.g. the content of a database containing business data that do not have a natural visual representation. This is one of the most difficult tasks in Information Visualization - finding a good and intuitive visual mapping for the respective kinds of data. The familiar proverb: "A picture says more than a thousand words!" seems to give a hint of the reason for using visualization instead of text. To confirm this assumption, a set of justifications are given [Car03].

Visualization amplifies cognition by:

1. increasing the memory and processing resources available to the users,
2. reducing search time for information,
3. using visual representations to enhance the detection of patterns,
4. enabling perceptual inference operations,
5. using perceptual attention mechanisms for monitoring, and
6. encoding information in a manipulable medium.

The variety of ideas that have been implemented to use visualizations to support the information-seeking process is enormous. Researchers all around the world try to find "the best visualization" to solve specific problems, resulting in hundreds of systems. Many of these inspired the realization of the VisMeB framework. Some ideas seemed to be a perfect fit in the created scenarios, others did not. Nevertheless, most of the tools made us consider a possible application in our work, although the original ideas were tailored to completely different application domains. An adaption was necessary, sometimes entailing a complete concept redesign. This chapter gives an overview of systems, techniques, and data types that influenced or are used in the framework developed. The number of techniques and data types used is manageable; in contrast, the number of systems in which they are employed is quite large. There is a natural restriction in the number of examples that can be cited. However, the outline will give a good insight into the field of visual information-seeking systems. To structure the visualization, a lot of possibilities are available. [Shn98] proposes a Data Type by Task Taxonomy (TTT) of information visualization. He differentiates diverse data types of the TTT that are organized by the current problems. Tasks in the TTT are defined as information actions that users want to accomplish, dependent on the task domain. Data Types and Task are displayed in Tables 2.1 and 2.2.

**Table 2.1: Data Type by Task Taxonomy (TTT) to identify visualization data types [Shn98]**

<table>
<thead>
<tr>
<th>DATA TYPES</th>
<th>EXAMPLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D Linear</td>
<td>Textual documents, program source code, alphabetical list of names.</td>
</tr>
<tr>
<td>2-D Map</td>
<td>Geographic maps, floorplans, newspaper layouts.</td>
</tr>
<tr>
<td>3-D World</td>
<td>Real-world objects such as molecules, the human body, buildings.</td>
</tr>
<tr>
<td>Temporal</td>
<td>Timelines for medical records, project management, historical presentations. Distinction from one-dimensional data: items have a start and finish time, items may overlap.</td>
</tr>
<tr>
<td>Multidimensional</td>
<td>Relational- and statistical-database contents</td>
</tr>
<tr>
<td>Tree</td>
<td>Hierarchies or tree structures; each item (except the root) has a link to one parent item</td>
</tr>
<tr>
<td>Network</td>
<td>Graph containing items linked to an arbitrary number of other items</td>
</tr>
</tbody>
</table>
Another way is to differentiate between diverse visual structures as can be found in [Car03]. He breaks them down into:

1. Simple Visual Structures,
2. Composed Visual Structures,
3. Interactive Visual Structures, and

As examples of simple visual structures we can identify e.g. lists, pie charts, box plots, 2D or 3D scatterplots, information landscapes, trees, or networks. Permutation matrices, parallel coordinates, graphs, scatterplot matrices, Keim spirals, or worlds within worlds belong to composed visual structures. Interactive visual structures are e.g. dynamic queries, magic lenses or techniques like overview + detail, brushing and linking, or extraction and comparison. Among the last group we can count filtering, selective aggregation, highlighting, or perspective distortion. A complete overview can be found in 2.3.

A lot of information can be encoded by simple structures like points, lines, areas, or volumes. Additional information can be assigned by the retinal properties [Ber83] impinging on a

- **Change of color**: This technique is not restricted to graphical display. The foreground or background of text can be varied as well as the color of a glyph or a specific area. If color is not available or not wanted, gray scale can be used.

- **Change of shape**: To emphasize the fact of changing an object’s state (e.g. from unfocussed to focussed) the shape can be varied. A typical example would be to change a circle into a square.

### Table 2.2: Tasks concerning the TTT [Shn98]

<table>
<thead>
<tr>
<th>TASKS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>Gain an overview of the entire collection.</td>
</tr>
<tr>
<td>Zoom</td>
<td>Zoom in on items of interest.</td>
</tr>
<tr>
<td>Filter</td>
<td>Filter out uninteresting items.</td>
</tr>
<tr>
<td>Details-on-demand</td>
<td>Select an item or group and get details when needed.</td>
</tr>
<tr>
<td>Relate</td>
<td>View relationships among items.</td>
</tr>
<tr>
<td>History</td>
<td>Keep a history of actions to support undo, replay, and progressive refinement.</td>
</tr>
<tr>
<td>Extract</td>
<td>Allow extraction of subcollections and of the query parameters.</td>
</tr>
</tbody>
</table>
Table 2.3: Another Way to Differentiate Between Visual Structures [Shn98]

<table>
<thead>
<tr>
<th>Structure</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Visual Structures</td>
<td>lists, pie charts, box plots, 2D or 3D scatterplots, information landscapes, trees, networks.</td>
</tr>
<tr>
<td>Composed Visual Structures</td>
<td>permutation matrices, parallel coordinates, graphs, scatterplot matrices, Keim spirals, worlds within worlds.</td>
</tr>
<tr>
<td>Interactive Visual Structures</td>
<td>dynamic queries, magic lenses, techniques like overview + detail, brushing and linking, extraction and comparison.</td>
</tr>
<tr>
<td>Focus + Context Attention-</td>
<td>filtering, selective aggregation, highlighting, or perspective distortion.</td>
</tr>
<tr>
<td>Reactive Visual Abstractions</td>
<td></td>
</tr>
</tbody>
</table>

- **Change of texture**: Simple textures can be modified to range from a non-filled appearance via dotted, ruled, or checkered, to completely-filled ones. However, the kind of texture is not limited to fill modes, but can also include images.

- **Change of size**: The change of size can result in e.g. a larger font, a circle with amplified radius, a heightened row, and so on. Important is the magnification factor, which again can encode different states.

- **Change of orientation**: A classic example is the change of stock price at the stock exchange. If there is an uptick an upward arrow can visualize that fact, whereas a downtick is represented via a downward arrow.

All the possible visualization structures introduced above are meaningful and could be used here. But now a complete introduction to this topic should be given, which leads to a slightly adapted approach. In this thesis the focus will be on systems that are directly connected to the own work, i.e. scatterplots, maps showing semantic similarity, table-based visualizations, semantic zoom realizations, and Multiple Coordinated Views. Therefore, systems can appear more than once, because of the wide distribution of Multiple Coordinated Views and their property of unifying more than one visualization technique in a single application. To build a base for a good introduction, the Visualization Reference Model (Figure 2.1) will be presented first, followed by a short introduction to the field of meta-data that act as the base input, and an overview of the different interaction techniques used.
2.1 INFORMATION VISUALIZATION - AN INTRODUCTION

2.1.1 Visualization Reference Model

Despite the variety of visualizations, we can describe the process of Information Visualization in a simple, but meaningful way. The Visualization Reference Model [CM99] specifies the mapping from so-called Raw Data to the final visualizations, the Views. Terms used are explained in Table 2.4.

![Visualization Reference Model](image)

**Figure 2.1: Visualization Reference Model [CM99]**

<table>
<thead>
<tr>
<th>RAW DATA</th>
<th>Idiosyncratic formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA TABLES</td>
<td>Relations (cases by variables) + meta-data</td>
</tr>
<tr>
<td>VISUAL STRUCTURES</td>
<td>Spatial substrates + marks + graphical properties</td>
</tr>
<tr>
<td>VIEWS</td>
<td>Graphical parameters (position, scaling, clipping, ...)</td>
</tr>
</tbody>
</table>

Human interaction plays a very important role in this context, as recognized in the definition of Information Visualization by the phrase "computer-supported, interactive visual representations". The advantage of Information Visualization in contrast to a simple drawn picture is the possibility of actively participating in the display. If an image is simply presented you can certainly draw some conclusions but you are not able to see what happens if you adjust parameters. Principally, this possibility of interaction leads to a strongly improved insight. Different techniques can be used to allow interaction, e.g. panning and zooming or filtering. These techniques will therefore be analyzed later on in this thesis. However, the focus will now be on the single phases described within the Visualization Reference Model. Let us assume we want to analyze and visualize websites containing information about a topic that a user is currently interested in. The first step is to transform the raw data (i.e. the website itself or rather the source code) into data tables via a set of data transformations. This can lead to e.g. an XML-file describing the content of this website. Possible meta-data (i.e. data about data) are e.g. "date", "language", "format", and so on (if the creator of the website followed the Dublin Core Meta-data
Standard, see Table 2.6). The resulting data tables can now serve as the base structure to visualize the single meta-data. Via visual mappings, every meta-data is assigned to a visual structure. In some cases e.g. by drawing a scatterplot to display the data, this results in a simple point for every kind of meta-data in the two- (or three-) dimensional space. Nevertheless, other structures like "bars" or "relevance curves" can be an adequate visualization for e.g. the relevance of a data set concerning a specific, previously defined measurement. These mappings are strongly dependent on the application domain and the base data. The Data Type by Task Taxonomy 2.1, the visual structures broken down by [Car03], and the retinal properties by [Ber83] have already been introduced. These can help to come to a decision about which structure fits which metadata. For instance, the retinal properties sometimes suit one type of data better than another, which is explained in Table 2.5. The underlying data are based on [Mac95].

**Table 2.5: Relative effectiveness of retinal properties.** Q = Quantitative data, O = Ordinal data, N = Nominal data. A + indicates the property is good for that type of data, a 0 indicates a marginal effectiveness, and a - only a poor one [Mac95]

<table>
<thead>
<tr>
<th></th>
<th>Q</th>
<th>O</th>
<th>N</th>
<th></th>
<th>Q</th>
<th>O</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SPATIAL</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>OBJECT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EXTENT</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Grayscale</td>
<td>0</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Size</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DIFFERENTIAL</strong></td>
<td>0</td>
<td>0</td>
<td>+</td>
<td>Color</td>
<td>0</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Orientation</td>
<td></td>
<td></td>
<td></td>
<td>Texture</td>
<td>0</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Shape</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

View transformations such as "distortion" or "camera movement" (viewing the same scene from another view angle e.g. in a three-dimensional representation) constitute the last step in obtaining the final view that users are confronted with. We can distinguish between three view transformations:

1. Location probes
2. Viewpoint controls
3. Distortions

Additional information can be obtained while using location probes. They use location in a visual structure to enrich the information already given. Examples are probing a point in a scatterplot to open a pop-up window with further meta-data, or magic lenses (see 2.1.3.3). To make details more visible, viewpoint controls can be used. Transformations like zooming, panning, or clipping the viewpoint belong to this category, as well as overview & detail (see Section 2.1.3.4). This results e.g. in a different viewpoint or in magnification of items, which can even result in a pop-up window displaying detailed
information. In contrast to the overview and detail technique, distortion uses a single window to display both views. The bifocal display [SA82] (see Figure 2.7) or the perspective wall [MRC91] are classic examples for this technique. A focus is set to the item(s) of interest and thus magnified whereas the surrounding area is shrunk.

All these techniques and properties presented are still dependent on a single object - the data itself. Therefore, a deeper insight into the field of meta-data, which build the data base for the visualizations used, will now be given.

### 2.1.2 Meta-data

Meta-data establish the core source for the visual meta-data browser VisMeB. The data to be visualized consist exclusively of meta-data - with a single exception: HTML documents that build one source of information can be displayed in their original, web-based form with the exception of any included images. The representation is restricted to text, which can possibly be seen as one kind of describing data. All further investigations will follow a strict line of ”meta-data only”.

**Definition 2.2 (Meta-data)** Meta-data are data about data. They provide information about or documentation of other data managed within an application or environment. For example, meta-data would document data about data elements or attributes, (name, size, data type, etc) and data about records or data structures (length, fields, columns, etc) and data about data (where it is located, how it is associated, ownership, etc.). Meta-data may include descriptive information about the context, quality and condition, or characteristics of the data.  

To unify the various kinds of meta-data, different meta-data standards are defined. The most important standards for the scenarios presented in this thesis are the ISO 92115 standard for geo-meta-data (see Figure 2.2 displaying a small cutout) and the DublinCore standard for web documents (see Table 2.6). More information concerning the meta-data used in the VisMeB framework is presented in the thesis of [Kle05]. Therefore, just a brief overview of the different meta-data types that are based on the standards mentioned will be given.

During the development process of VisMeB the development team was confronted with a lot of meta-data that should have followed a standard. Unfortunately, the reality was very different. Gaps and outliers are quite normal and regrettably not very rare. Some of them can be found easily, even without a visual display, others can not. Finding these hard-to-detect items in an efficient and effective way is one of the main advantages of visual seeking systems. These systems are therefore supported by interactive techniques.

---

Table 2.6: The 15 elements of the DublinCore core meta-dataset, divided into categories

<table>
<thead>
<tr>
<th>CONTENT</th>
<th>INTELLECTUAL PROPERTY</th>
<th>INSTANTIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>· Title</td>
<td>· Creator</td>
<td>· Date</td>
</tr>
<tr>
<td>· Subject</td>
<td>· Publisher</td>
<td>· Language</td>
</tr>
<tr>
<td>· Description</td>
<td>· Contributor</td>
<td>· Format</td>
</tr>
<tr>
<td>· Source</td>
<td>· Rights</td>
<td>· Identifier</td>
</tr>
<tr>
<td>· Relation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>· Coverage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>· Type</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

that make cooperation between user and system possible and effective. A short overview of techniques implemented in the VisMeB framework will now be presented.

2.1.3 Interactive Techniques

Interactivity is a feature of Information Visualization that stands out from fixed scientific graphics. The possibility of interactively engaging in the process of visualization - during data transformation, visual mapping, or view transformation - enables the user to simulate situations that would take a lot of time to replicate in any other way. Nowadays, a lot of techniques are used in realizing a system consisting of visualizations. Thus, the focus will be placed on the ones used in the VisMeB framework, i.e. Dynamic Queries, Brushing & Linking, Movable Filters, Overview-Plus-Detail, Focus & Context, and Panning & Zooming.

2.1.3.1 Dynamic Queries

Dynamic queries allow the user to directly manipulate the visual display. This technique was introduced in the early 90’s by [WS92] and implemented in the Dynamic Home-Finder (see Figure 2.3). The idea was to help a homebuyer in finding a home that conforms to his wishes. A map of the Washington D.C. area is displayed on the left side of the screen, while controls are located on the right. Yellow dots on the map mark homes that fulfil the criteria chosen by the user. Different interaction controls like buttons or sliders are available to restrict the data set to a convenient subset. Operating the controls has to change the current display in a tenth of a second ( [CMN86]). This is the time in which a system must respond to a direct manipulation of the visualization. By allowing these extremely fast and reversible modifications, it is possible to provide direct feedback to enhance the exploration process within a few fractions of a second. The technique of Dynamic Queries is close related to Tight Coupling. Different components of a system are intimately connected in such a way that changes in one component also result in changes in other components.
2.1.3.2 Brushing & Linking

Brushing & Linking deals with the connection of various views of the same data set. A highlighting or selection of a data subset ("Brushing") in one view affects other views ("Linking"), depending on the defined interrelationship. This is an extremely important technique used in the field of Multiple Coordinated Views. The term "Coordinated" refers to exactly this correlation between different sights. Many possibilities for realizing the highlighting in the corresponding view(s) are available, like changing the color, size, shape, background, or labeling ([EW95]). Figure 2.4 shows a scatterplot matrix realizing the brushing and linking concept. The base data are taken from [Lub62], describing three species of flea-beetles: Ch. concinna, Ch. heptapotamica, and Ch. heikertingeri, and six measurements on each, such as width of the first joint of the first tarsus in microns or the maximum width of the head between the external edges of the eyes in 0.01 mm and so on. The idea is to brush some points in one plot which leads to an effect like e.g. highlighting to be applied on those points in the other plots that represent the same data items, as can
be seen in Figure 2.4. The user initiates a selection that has been applied to the second plot of the first row, marked by the red rectangle. The act of selection highlights the brushed glyphs by changing their color and shape. This results in red crosses that can be found in any displayed plot. In this way, conditional dependencies can be found and an analysis can be made over more than two dimensions.

2.1.3.3 Movable Filters

Interactive filters like the "Dynamic Queries" described above provide a good opportunity to support the so-called "what if..." activity. Users can adjust settings and reverse the action immediately without fear of changing the display irreversibly. Another example of such an interactive filter is the "Movable Filter" ([FS95]), also known as the "Magic Lenses". This filter can be moved across the display to modify a view. Moreover, modification is not restricted to a specific function, but can be realized by diverse actions like e.g. filtering, adding additional information for underlying objects, or enlargement. Figure 2.5 displays an example using US census data from a database storing data like population, crime rate and so on. Each city is represented as a box and mapped in an intuitive way onto the 2D plane - to the physical location on a map. In the case presented,
2.1 INFORMATION VISUALIZATION - AN INTRODUCTION

2.1.3.4 Overview-Plus-Detail

The Overview-Plus-Detail technique connects two views in a direct and intuitive way - a multi-window. A part of the overview is magnified and displayed off to the side to avoid overlapping. Different information levels can be displayed. Typically, a visual marker

---

**Figure 2.4: Brushing and Linking in a scatterplot matrix** [Lub62]

the ”1991 crime index” is associated with the lens, as can be seen in the window’s title. A slider is used to define a threshold value for a query, which is displayed to the left. The two buttons ”i” and ”¿” determine if the user wants to highlight items lying in the area below or above the chosen threshold. In Figure 2.5 the cities with a crime rate above the defined threshold value are colored red, all others stay white. By adapting the value for specific questions, it can be very quickly ascertained whether a city fits to a grid pattern or not.
Figure 2.5: Movable Filter by Fishkin and Stone [FS95]

(e.g. a colored rectangle) highlights the position of the detail view within the overview. This makes the Overview-Plus-Detail technique appear as a kind of Magic Lens, where the lens works in the overview and the effect is an enlargement, presented in the detailed view. One disadvantage of this technique is the simultaneous representation in separate displays which is likely to cost more in user time. Context switching becomes necessary, which can sometimes be avoided by other techniques like “Focus & Context”.

Figure 2.6 shows a very widespread application, the usage within a geographical map. A larger part of a specific area (in this case a part of the USA) is displayed in the upper left corner. It is only possible to recognize the state borders of states located in this region e.g. Virginia, or Ohio. To get more information, the part of the map which is colored in a darker green is magnified and displayed in a size that makes e.g. highways, rivers, or airports visible. In the present case the selected area shows the North Bend Rail Trail State Park. This allows a detailed view of specific regions without losing the overview.

2.1.3.5 Focus & Context

The Focus & Context idea is very similar to Overview-Plus-Detail. An overview (context) is given, as well as a detailed view (focus), which seems to be a simple naming change. Nevertheless, a difference exists in the number of displays used. While Overview-Plus-Detail needs two different displays, Focus & Context unifies these in a single one. To realize this situation, distortion is used. Thus, problems can arise when any task requires precise decisions about distance, scale, or alignment. A very successful concept is the Bifocal Display [SA82]. Figure 2.7 shows the principle of the concept. An even better impression is given by one of the most famous realizations of this idea - the Perspective Wall [MRC91]. Figure 2.8 shows an example. In this case, a three-dimensional effect is simulated by expanding the small part of the wall at the front to both sides. The centered and thus readable part, clarified by a red rectangle, displays files, distributed depending on date and kind of file. Although no detail information is available for the outer parts of the
2.1 INFORMATION VISUALIZATION - AN INTRODUCTION

Figure 2.6: Overview plus detail view for the North Bend Rail Trail State Park, WV (see: http://www.wvstateparks.com/northbendrailtrail/index.html)

It can be seen that more files are located in the months after October 96 than before it. Moving the focus point, i.e. "scrolling the wall" makes it possible to see the parts currently located in the background. The three-dimensional effect is used to emphasize the fact that a wall is being displayed, but it is not necessary for bifocal displays in general.

Figure 2.7: The principle of the Bifocal Display [SA82]

2.1.3.6 Panning & Zooming

Panning and Zooming is a widespread technique in the two-dimensional space. Zooming in or out lets the user change his viewpoint, i.e. have a closer or more distant look at a
Figure 2.8: The Perspective Wall as an example of a bifocal display; developed by Inxight Software Inc. [MRC91]

specific image fraction. The viewing frame retains its constant size whereas the image fraction is scaled up or down. In contrast, panning moves the viewing frame over the image without the size changing. When adding a third dimension, [CM99] introduced the term "camera movement" instead of panning. While in two-dimensional space panning is restricted to the x- and y-dimension, camera movement makes it possible to look behind the display or view it from the side. A special kind of zoom is given by the "Semantic Zoom". Zooming in gives additional information that would not be visible by performing a simple magnification. Because of its close connection to the granularity concept described in this thesis, it will be explained in detail in Chapter 4. A schematic representation is shown in Figure 2.9 while an example of panning and zooming is given in Figure 2.10 with the aid of the SYNTH system [Lab02].
2.1 INFORMATION VISUALIZATION - AN INTRODUCTION

**Figure 2.9**: Panning describes the smooth movement of a viewing frame over an image (left), whereas zooming is a magnification of a decreasing fraction of an image or vice versa (adapted from [Spe01a]).

**Figure 2.10**: Panning and Zooming in SYNTX - A Gamma-Ray Spectrum Synthesizer. Moving around the blue rectangle in the upper right corner (panning) can be used to explore small areas of the overview, shown in the gray colored overview rectangle. A larger or smaller area can be investigated by changing the blue rectangle’s size (zooming) [Lab02].
2.2 Scatterplots

Scatterplots are a widespread technology for visualizing multidimensional data. The different variants can be categorized according to their dimension (2D or 3D) and their interaction facilities. First of all, a definition should be given.

**Definition 2.3 (Scatterplot)** A (2- or 3-dimensional) scatterplot or scatter graph is used [...] to visually display and compare (two or three) sets of related quantitative, or numerical data by displaying only finitely many points, each having a coordinate on a horizontal and a vertical axis.

(Adapted from http://encyclopedia.thefreedictionary.com/Scatter%20plot)

The possibility of adding more than two (or three) dimensions can be realized using e.g. different shapes, colors, or size for the commonly displayed points. In particular, this technique is provided for highlighting selected or focussed points. Brushing and linking capitalize on this by e.g. changing the points’ color to a darker saturation. Using glyphs (a carved figure or character, incised or in relief) instead of simple points are another way to encode additional information. As distinct from a mere difference in shape (e.g. circle, square, rectangle in the 2-dimensional space or spheres, cubes, cuboids) a direct conclusion can be drawn on the basis of their appearance. A CD- or a book-icon in a library catalogue immediately gives a hint as to the underlying source; a circle and a square could not. Problems arise if no natural mapping is available or meaningful. Imagine the situation of representing the title, or the location in the library. In this case, another technique can be applied - the tooltips. Tooltips are small textfields that appear when the mouse pointer is moved over an object. The tooltip can provide different information, dependent on the particular situation. Take the example of the library. Provided that the meta-data "title" has not already been assigned to an axis, then showing the title (and in particular in an easily readable manner) would be meaningful information, while displaying the exact position in a location map could be reasonable, too.

A great variety of techniques can be implemented to enhance the interaction between user and Scatterplot. Zooming and panning are a prominent example that is especially used for large data sets that produce a very crowded layout. A zoom to an interesting subset can mask out disturbing points and discover cluster or patterns that were not visible before. Panning allows the user to move the viewpoint all around the display without changing the zoom factor. This can lead to a fast and improved exploration of the whole data set. An overview and detail realization of panning and zooming can avoid the risk of being "lost in space".

The opportunity to filter out uninteresting data points - whether by using dynamic queries or simple non-dynamic filters - is another very interesting and necessary feature, particularly if we recall the information-seeking mantra: "Overview first, zoom and filter, then details on demand". Restricting the data set by fading out redundant subsets allows
a more detailed and less expensive exploration because of decreased stimulus overload. A very simple but efficient - and not to be discounted - feature is the axis assignment. The possibility of changing the allocation enables the user to adapt the representation to his own preferences. Detectable problems are easier to detect and they can be analyzed from different viewpoints. Relationships between diverse meta-data can be visualized and recognized.

Pop-up windows for displaying detailed information are an advancement from tooltips. Whereas tooltips usually contain a single sentence or word, pop-up windows can visualize images as well as text. Their information content is higher, resulting in a larger space usage. Some systems include a specific area on the screen to display these details. On the one hand this avoids the possible occluding of information, on the other hand a fixed location can help the user to orientate better.

Because of the great variety of scatterplots used nowadays, only a short description will be given here. Most of the applications providing scatterplot visualizations use multiple coordinated views. Therefore, a number of scatterplot visualizations can be found in Section 2.6, dealing with multiple coordinated views.

### 2.2.1 2D-Scatterplots

We have to differentiate between the diverse versions of 2-dimensional scatterplots. For clarity, we will distinguish between the following three situations:

1. Scatterplots displaying geographical data (i.e. longitude and latitude are mapped to the x- and y-axis),

2. Scatterplots displaying abstract data, and


Systems providing the possibility of scatterplot matrix visualizations can easily be reduced to one of the preceding groups by restricting the number of displayed scatterplots to one only.

### 2.2.1.1 Geographical Scatterplots

The Dynamic Homefinder [WS92] (see Figure 2.3 and Section 2.1.3.1) is a very early example of an interactive geographical scatterplot. It displays a map of the area of Washington, D.C. and therefore maps longitude and latitude to x- and y-axes. Dynamic queries are used to adapt the display to the user’s wishes.

SeeNet (See a Network) by [BEW95] is a system that visualizes telephone traffic inside the USA. The inbound and outbound calls can be visualized as rectangles (represented by the horizontal and vertical dimensions of the rectangles) as well as by lines
drawn from source to destination. In Figure 2.11, the rectangles represent inbound and outbound calls from the switches in the AT&T network. The wider a rectangle is, the more inbound calling is taking place at the current time stamp. An analogous statement applies to height and outbound calls. This picture is part of an animation sequence and displays the telephone traffic at 11:05 as indicated by the time stamp on the lower left corner.

**Figure 2.11: Visualization of telephone traffic in the USA using SeeNet [BEW95]**

Spotfire [Ahl96] is one example of applications that combine various views in a single system. Visualizations like scatterplots (2D and 3D), histograms, bar-charts, line-charts, pie-charts and tables are integrated and can work as multiple coordinated views (see Chapter 3). A special advantage of Spotfire in contrast to e.g. the Dynamic Homefinder is its data independency. The user is no longer limited to a specific application domain. This independence includes the possible use of geographical maps. The example in Figure 2.12 shows a map of Sweden and the deposits of heavy metals.

Like Spotfire, the Open Visualization Data Explorer (OpenDX ²) [Res04] is a visualization framework that allows users to apply diverse visualization and analysis techniques

²http://www.research.ibm.com/dx/
2.2 S C A T T E R P L O T S

Figure 2.12: Spotfire showing the deposits of heavy metals in Sweden [Ahl96]

to their data. It is the direct successor of the Visualization Data Explorer introduced by IBM in the early 90s. The application provides a set of interactive tools, as well as visualization artefacts like points, lines, areas, volumes, images or geometric primitives in any combination. Furthermore, it is not restricted to a specific domain, which makes it usable with almost any kind of data. One example of the diverse visualizations is a geographical scatterplot. It is displayed in Figure 2.13, using data concerning the initial purchase intent in the US.

2.2.1.2 Scatterplots for Abstract Data

A very early work that had a strong influence on the field of interactive, 2-dimensional scatterplots is the FilmFinder [AS94b] . It can be seen as an advancement of the Dynamic Homefinder, presented two years earlier. The FilmFinder highlights the relationship between popularity and the period of movies (see 2.14). Dynamic queries allow the user to adapt the display using direct manipulation. This means an immediate adjustment of the interaction tools (sliders, buttons) and visualization.

The Interactive Timeline Viewer (ItLv) [MFM03] is an application that uses a two-dimensional scatterplot display to present the content of a digital library. Data are delineated with respect to their temporal context, i.e. the events are time-based. Multiple and
Figure 2.13: OpenDX displaying the initial purchase intent in the US within a map [Res04]

Figure 2.14: FilmFinder showing movies with Sean Connery [AS94b]
interlinked views of the entire data set, including meta-data, can be displayed at the same time. Figure 2.15 shows an example concerning Miguel de Cervantes’ life.

![Figure 2.15](image)

**Figure 2.15:** The Interactive Timeline Viewer presenting an overview of events in Miguel de Cervantes’ life (background) and a pop-up window with detailed information about a specific event (foreground, yellow color) [MFM03]

The Envision system [FHN+93] uses glyphs instead of simple geometric shapes to visualize information that is additional to the two scatterplot dimensions already displayed. It was a prototype digital library of computer science literature developed at Virginia Tech under a cooperative agreement with ACM and NSF Grant. Approximately 200,000 documents were included. The greater part of the documents consisted only of meta-data, often with abstracts, but some full-text and some multimedia documents were included. The user-controlled system facilitates examining very large data sets, displaying multiple aspects of the data simultaneously and efficiently, and interactive discovery of patterns in the data. The color, type and size of the single items encode additional information to impart a faster insight into the data set. Figure 2.16 shows the result of a query containing the query terms “Card, Stuart K.” for author and “human-computer interaction” for words in the title. This information is displayed in the Envision Query Window on the left side. Results are presented on the right side in the Envision Graphic View, where author and year of publication are assigned to the y- and x-axis, respectively. The shape clarifies whether the item found is a book, a journal article, or a proceedings article. The importance is signified by two characteristics, the label (providing a relevance rank from 1 to the number of items found) and the color, which implies a high (orange) or a low (light blue) relevance. Thus, the most important documents can be found very quickly
and easily, in this case a proceedings article from 1986 and a book from 1983.

**Figure 2.16**: Envision system for visualizing the content of a digital library [FHN+ 93]

The Search Result Explorer of the xFIND system [ASL+ 01] is another example of encoding information in the shape of data points. The data presented in Figure 2.17 originate from a collection of 44,878 documents related to the topic of knowledge management. The query uses the word "agent" and leads to a result set of 314 documents. Relevance is mapped to the y-axis, the document's size to the x-axis. In the present case, additional information is mapped to the color and size of the displayed items. Larger objects visualize a higher relevance, whereas the color ranges from white (recent documents) to yellow (older documents). The effect of mapping relevance to size is a redundant encoding and becomes clearly visible in Figure 2.17 - the most relevant documents are larger and towards the top of the display. If items are to be drawn too close to each other, a group icon is used to represent that subset. The size and color of this subset is determined by the maximum, minimum, median, or average value of the group’s members, depending on the choice the user has made. The possibility of zooming made it necessary to add an overview window, located in the lower left corner, to help maintain the context.
2.2 Scatterplots

2.2.1.3 Scatterplot Matrices

Scatterplot Matrices can be seen as extended scatterplots, where the overview is given a greater importance than the detailed view. More than one pair of characteristics can be compared simultaneously, which makes it easier to detect outliers or patterns. However, because of the reduced space available for each single scatterplot, detailed information can be lost in this view - depending on the kind and size of data. An additional zoom function can reduce this drawback by maximizing the scatterplot of interest. In this context brushing and linking becomes very important because connections between the different views are the main advantage of this kind of visualization.

An extension of the Pivot Table interface spread by Microsoft Excel is the Polaris interface [CSH02]. The main task for this system is to explore large multi-dimensional databases. The configuration of fields on shelves, which can be generated by dragging fields from the database schema onto shelves throughout the display, is called visual specification. As a consequence the user is able to construct visual specifications of graphical displays and generate relational queries from these. Visual feedback during the development process of the visual specification allows the construction of complex queries and
visualizations. Figure 2.18 shows the Polaris interface, including an explanation of how to use it. The constructed scatterplot matrix shows sales versus profit for different product types (in this case coffee and tea) in different quarters. Using different shapes for the icons displayed allows a quick assignment of an item to the corresponding market (west, south, east, central). As is usual for scatterplot matrices, brushing is possible but was not done for this example.

The statistical software ClusCorr98 [HJMB02] combines a large variety of visualizations to illustrate the raw data to the user. It allows the use of internal and external databases that can be accessed from the Excel environment. Clustering plays an important role in this context, which results in mainly cluster-based visualizations, as can be seen in Figure 2.19. The underlying data originate from a snapshot of the monitoring of phytoplankton (flow cytometry measurements). Flow cytometry provides the possibility of obtaining two different kinds of information - the number of cells (in this case algae) per unit of sample-volume and the optical characteristics of each cell, i.e. parameters of light scatter and of fluorescence. This allows a differentiation between the different pigment groups that the cells belong to. To identify the corresponding groups and assign the organisms to them, classes have to be determined. Because in this case five parameters have to be taken into account a manual building of clusters is difficult or even impossible. A cluster algorithm is therefore applied. Different clusters are identified by color, cluster centroids are marked by large black crosses. The five parameters used in the flow cytome-
ter are the forward scatter (FSC, to measure the cell sizes) and the side scatter (SSC, for information about the structure of the cells’ surface) as parameters of scattered light, and FL1, FL3, and FL4 as parameters of fluorescence. All axes-combinations are displayed within the scatterplot matrix in figure 2.19.

**Figure 2.19:** The ClusCorr98 system displaying an extract of a scatterplot matrix of cluster memberships [HJMB02].

Figure 2.20 displays a scatterplot matrix [Voi02] developed at the VRVis research center in Vienna, Austria. In contrast to conventional scatterplot matrices, improvements were made in the areas of the **adjustable point size**, the **free choice of plots**, and the **tile mode**. This can lead to an improvement in the field of effectiveness with large datasets and categorical data. In this case, the underlying dataset concerns the field of market research.

---

3 [http://www.vrvis.at/vis/resources/DA-RVoigt/masterthesis.html](http://www.vrvis.at/vis/resources/DA-RVoigt/masterthesis.html)
The aim is to find out whether certain groups of the population prefer certain brands. Two groups are identified: (1) Households with one or two adults and no children, and (2) households with one adult and one or more children. The scatterplot presented uses the so-called tile-mode to solve the problems of overlapping datapoints. Color is an indicator for the number of points that would lie in the area of the tile in a conventional scatterplot. The mapping ranges from cyan (zero) via green (one third), yellow (two thirds) to red (maximum number in the plot). When brushing is used, the tiles have two colors: the outline, which is a measure for the total number of cases in the area of the tile, and the color of the inner quadrant to represent the number of brushed cases. In the preceding case, the households with one adult (second row in the upper left plot, counting starts at row zero) and one or more children (columns 1-3, starting with column zero) are brushed, as is indicated by a blue rectangle. The corresponding tiles are filled in all plots, all others are not. By an investigation of the contrast between outline and filling, any patterns and outliers can be found.

Figure 2.20: Scatterplot Matrix developed at the VRVis research center in Vienna, Austria [Voi02].

Systems like HyperSlice [WCB96], the Spreadsheet Framework by [CBRK97], ArcMap as a component of ESRI ArcView [ESR04], or the Influence Explorer [TSDS99] are further applications providing scatterplot matrices and are mentioned in order to give a brief overview of the large number of systems implementing this idea. Because it will not be possible to provide a complete overview, we will now focus on the area of three-dimensional scatterplots.

http://www.esri.com/software/arcgis/about/desktop.html
2.2 SCATTERPLOTS

2.2.2 3D-SCATTERPLOTS

It is still debatable if the use of 3-dimensionable scatterplots for abstract data visualization adds an advantage over 2-dimensional ones with regard to contrast. In scientific visualization, a 3-dimensional representation is reasonable because of the "natural" mapping from the real to the virtual world. But one property of abstract data is its abstraction, i.e. no natural mapping is available that helps the user to recognize a familiar representation. The increased cognitive and mental workload for the user, which is caused by a complicated navigation through, and a confusing presentation of, a 3-dimensional space, is the major disadvantage of this visualization. To reduce this workload a more simple interaction has to be used, e.g. no "fly around" mode should be available. There is an unavoidable necessity of providing a reset mechanism that enables the user to return to an initial position in an efficient way.

The previously described OpenDX 5 is an application that provides diverse visualizations, as described above. Although two-dimensional visualizations are included, the three-dimensional ones prevail. Scientific as well as abstract data can be visualized. Figure 2.21 shows a 3D scatterplot displaying results obtained from a credit card application. Approved card holders are represented as colored spheres. Clicking on a sphere provides historical information on the card holder. The three axes show applicant information, in this case work duration, debt ratio, and net worth. Color indicates the credit limit of card holders (see upper left corner for a caption), the size of the spheres indicates the salary. New applicants are displayed as white spheres. To determine the credit limit of an applicant, the 3D position of the sphere relative to the historical data base information can be used.

The three-keyword axes display of the NIRVE system (The NIST Information Retrieval Visualization Engine) [CLS00], which was developed at the National Institute of Standards and Technology (NIST), maps the keyword strength to the axes (see figure 2.22). It is not restricted to three keywords as would seem to be the case because of the number of axes i.e. three. Any subset of keywords can be assigned to an arbitrary axis. To control this assignment, a separate keyword window with a column of checkboxes for the X, Y and Z axes is used. Axes are labelled with their associated keyword(s). To find the correct position, the average of the constituent keyword strengths is computed. If the axis choices are changed an update is performed dynamically. Nevertheless, the individual icons encode the strength for all the keywords. Each item is represented by square icons, aligned to the origin. Each icon consists of a bar chart to present information about the relative frequency of query terms. The connection of query terms to retrieved documents can be interactively explored by the user, which is the main advantage of this visualization. For example, if the same keyword is assigned to all three axes, a list (linearly ordered by keyword strength) would be the result.

---

5http://www.research.ibm.com/dx/
Figure 2.21: IBM Open Visualization Data Explorer presenting data from a credit card application [Res04]
2.2 SCATTERPLOTS

As the direct successor of LinkWinds (the Linked Windows Interactive Data System) Webwinds\textsuperscript{6} [Lab05] provides tools for visualizing two- and three-dimensional data. The XYZPlot as a realization of a 3D scatterplot offers the ability to rotate, zoom, and change colors among other interaction tools. This division in specific interaction sections avoids a cognitive overload for users and makes it easy for them to adapt the visualization to their own vision. Figure 2.23 shows an example of tabular data with latitude and longitude as independent variables. The dependent variable is plotted on an axis perpendicular to latitude and longitude. Color is used to reinforce the impression of a three-dimensional representation.

The Voxelplot\textsuperscript{7} [Sah02] is another application developed at the VRVis research center in Vienna, Austria. Of special interest is the fact that a bridge has been built between scientific visualization and information visualization. A real world object can be displayed simultaneously with specific abstract data of interest. Figure 2.24 displays a catalytic converter in the upper left corner and additional data like pressure and velocity in the other scatterplots. The respective axis assignment can be found in the lower right corner after

\textsuperscript{6}http://www.openchannelfoundation.org/projects/WebWinds
\textsuperscript{7}http://www.vrvis.at/vis/research/voxelplot/index.html
Figure 2.23: Webwinds XYZPlot, developed at the NASA’s Jet Propulsion Laboratory [Lab05]
marking the view of interest by means of the red square. A synchronized movement (e.g. rotation) and the possibility of marking areas of interest by color enables the detection of unknown relations. This brushing technique is supported by different brushing modes - the range brush (uses alphasliders to limit the marked data), the beam brush (defines a cylinder by center and radius, orthographic to the current view), and the cluster brush (tries to find and mark clusters).

Figure 2.24: Voxelplot developed at VRVis research center, combining scientific and information visualization [Sah02]

2.3 Semantic Similarity Maps

Document visualization is usually made by semantic maps showing the relationship between documents within a document corpus. Similarity between items is indicated by spatial closeness. In general, many schemes for document visualization include three stages [Spe01b]:

1. Analysis
2. Algorithms

3. Visualization

The first step is to extract the essential descriptors of a collection of texts, in general a set of key words. The output from this stage is a compressed representation of the original document. The descriptors in this case may be first-order statistics, higher-order statistics, or semantic aspects including the meaning of the documents. In processing the documents, so-called ”stopwords” like ’is’, ’a’, or ’the’ are removed. Vectors with maybe 10,000 numbers corresponding to the frequency of occurrence of words represent single documents.

The task of the second stage is to generate a usable representation of the document corpus. The vectors have to be transformed because the visualization in a very high-dimensional space is totally unsuitable. This can be done by clustering and projection into the two- or three-dimensional space. The main task is to place similar documents close to one another i.e. form clusters and project these clusters by once again expressing similarity by means of spatial closeness.

To solve this problem, many different algorithms are known. The following section will describe three main approaches - Latent Semantic Indexing (LSI), Self-Organizing Maps (SOM), and Force-Directed algorithms. A detailed and mathematical description of these algorithms is given in appendix A.

1. **LSI:**

   The initial situation for applying a Latent Semantic Indexing (LSI) algorithm is a term-document matrix [DDL+90], [BYRN99]. It includes all the documents of the document collection and all terms included in any document, except words that are deleted by a stopword elimination. This matrix is usually very large and has therefore to be reduced to define a meaningful layout. For this purpose, each document and query vector has to be mapped from this high-dimensional to a low-dimensional space. The technique for implementing this process is called singular-value decomposition (SVD) and leads to a particular latent semantic structure model. The initial term-document matrix is therefore decomposed into three specific matrices containing singular vectors and singular values. These matrices represent a breakdown of the initial data into linearly independent factors. The result of the breakdown is a model that approximates to the original model but works with far fewer dimensions. All possible similarity relations, i.e. term-document, term-term, and document-document, can be represented by values within these fewer dimensions.

2. **Self-Organizing Maps:**

   The Self-Organizing Map (SOM) is a neural-network algorithm and supports unsupervised learning. It is used to analyze and visualize high-dimensional data. SOM was developed by Teuvo Kohonen (see [Koh97]) and is therefore sometimes
named the "Kohonen-Map" or "Kohonen-Layer". The main idea is to reduce the complex relationships between high-dimensional input data into simple geometric relationships on a low-dimensional display, the map. This map consists of a two-dimensional regular grid of points which are called neurons, units, or nodes. Each of these items represents a single dataset which is provided in the form of a vector that is associated with a model of some observation. The task of the SOM algorithm is to determine a set of models that optimally describes the domain of observations. As explained above, similarity is indicated by proximity, i.e. similar models are closer to each other than less-similar ones.

Unsupervised learning plays a very important role in this context. With each input vector the "winner-neuron" or "best matching unit (BMU)" (i.e. the most similar one to the input vector) is searched for. A learning process is started by the BMU and its neighbors by adapting their data vector step by step to the input vector. Figure 2.32 shows a typical layout.

3. **Force-Directed Algorithms:**

Force-directed algorithms are normally used to compute a layout for graphs. These graphs often simulate physical or chemical models. A layout of a document collection is another possible application scenario that can be used. The main idea of these algorithms is to utilize attraction and repulsion to construct a good layout. Documents can be stored as nodes and weighted edges can define the similarity between the documents, for example. One therefore often uses an \( n \times n \) matrix \( A \) containing the values \( a_{ij} \) describing the similarity between document \( i \) and document \( j \). In the preceding case a good layout results in a positioning of nodes close to one another if documents are similar, and further apart if they are not. This results in vertices being well spread in the available space and a close placement of adjacent vertices. The Spring Embedder is a typical and widespread model that deals with exactly this topic. To illustrate this model, we can imagine charged balls being the vertices and springs taking the place of the edges. Springs are chosen because extension as well as compression is possible, which is not case with sticks or ropes. While the springs connect the particles and provide an attraction, the charged balls repel each other. Two possible ways to express this model are:

- Forces that act on the physical objects, and
- Potential energy, which reflects the current stage of a configuration and how it matches the modeled design goals.

In this case a stable state is sought where all forces cancel each other, or the energy level has to be minimized in a direct way. Both approaches, the force-directed and the energy-based placement, are described in detail in appendix A and in [KW01]. To compute a local minimum of the objective function, simulated annealing is used. This model provides an analogy between the physical process of annealing, i.e. the
way in which a metal cools and freezes into a minimum energy crystalline structure and the search for a minimum in a more general system. The advantage of simulated annealing is the fact that a captivity in a local minimum is avoided. Two main steps characterize the algorithm: change of temperature and computation of the energy level at this temperature. If a candidate solution is given, a new one is obtained by modifying the current one. For the case in which the new solution results in a lower value of the objective function, it becomes the new candidate solution, otherwise it becomes the new candidate solution with only a specific probability \( e^{-\Delta U / T} \), where \( \Delta U \) is the amount the objective function increased, and \( T > 0 \) is a temperature parameter. \( T \) is used to enable the algorithm to transcend energy boundaries to find a solution that is likely to be better “behind” this boundary. The temperature is under the control of a temperature scheme or temperature scheduling. Thus, the optimization starts at a high temperature \( T_0 \) which means that a large percentage of random steps resulting in an energy increase is accepted. After a sufficient number of steps has been completed, the temperature is decreased until the final temperature \( T_{\text{final}} \) is reached.

In many cases a combination of different algorithms is used to compute a meaningful layout. Additional algorithms are included to reach a better fine-tuning of the end result. A possible application scenario is given by a hierarchically ordered document collection. Thus, a general layout is provided by the different hierarchy levels and usually only documents of the current level have to be positioned. To visualize the different clusters, polygons sometimes bound the specific area. In this case, the area’s size directly corresponds to the number of contained documents. The partition of an area into sub-areas including the next hierarchy level can be accomplished by Voronoi diagrams.

The main task of Voronoi diagrams [OBS92] is to divide a space filled with geometric objects into cells, each of which consists of the points closer to one particular object than to any others. This can be described as follows:

**Definition 2.4 (Voronoi Diagram:)** A Voronoi Diagram \( V = V(p_1), ..., V(p_n) \) is a partition of the plane into \( n \) polygonal regions. The calculation of these regions proceeds by sites in the euclidean plane, \( P = p_1, p_2, ..., p_n \) by assigning every point in the plane to the nearest site.

**Definition 2.5 (Voronoi Cell:)** A Voronoi Cell or Voronoi Region \( V(p_i) \) is defined by all points in the plane that are assigned to this site \( p_i \), i.e.

\[
V(p_i) = \{ x : \| x - p_i \| \leq \| x - p_j \| \text{ for } i \neq j \} \tag{2.1}
\]
Voronoï diagrams tend to be involved in situations where a space is to be partitioned into "spheres of influence", as can be seen in e.g. the InfoSky system (see Figure 2.28).

The third and only occasionally last stage defines the mapping to a specific layout, concerning the kind of representation as well as the interaction features. This is the most interesting part of the scheme for this thesis and will therefore be explored in more detail. Various schemes are available for document visualization. Examples will be shown, including galaxies, themescapes and Kohonen maps.

The Galaxy Of News [Ren94] is a specific visualization with a layout based on similarity, but it does not fit exactly into one of the three categories mentioned above. It will therefore be described first. The idea of the Galaxy of News is the visualization of a network of related news articles. To arrange these articles, a hierarchy of topical keywords from general to more specific has to be generated. This can lead into article headlines and into full articles. All these are arranged in a three-dimensional space without discrete and predetermined steps, but with a smooth and continuous transition. Thus, it is very hard to maintain one's orientation while navigating through this space. Figure 2.25 displays an example of a Galaxy of News.

### 2.3.1 Galaxies

As mentioned above, the second stage for document visualization creates clusters. If these clusters are projected onto the two-dimensional space, the resulting visualization is called Galaxy. A zoom into a cluster allows the user to get more information about this specific cluster. This can be done with or without animation. Typical examples can be found in the IN-SPIRE [Lab04] and the SPIRE system [TCK+01] displaying a galaxy (Figure 2.26).

The xFIND system [ASL+01] includes different visualizations like the above mentioned Search Result Explorer (see figure 2.17). A version of a galaxy visualization is realized within the xFIND system by the Visualization Islands (Figure 2.27). Important regions are displayed as islands that are integrated in an "ocean of information". The information to be visualized belongs to the field of knowledge management, containing a database of about 45,000 documents.

InfoSky [AKSG04] provides a representation very similar to the SPIRE galaxy described above. Prerequisite for this visualization is a hierarchically structured document collection. Documents are visualized as stars and similar documents form a cluster. A collection consists of clusters and stars, bounded by polygons. This structure is realized by three algorithms. First, a similarity placement algorithm is used to position cluster
centroids as well as documents themselves, realized by an optimized force-directed placement algorithm. The geometric transformation has to define the final placement of documents and collection centroids within a bounding polygon. Therefore, the normalized coordinates have to be transformed into a multi-axis coordinate system which is determined by the bounds. As a last step, an area partition is necessary. Because of the hierarchical structure dividing the information space in parent and sub-collections, the sub-collection centroids are used to divide the parent collection into polygonal sub-areas whose size directly corresponds to the number of documents included in the respective sub-collection. This is accomplished by a recursive Voronoi subdivision of the available space. Variable magnification is possible, which results in an adaption of the available space depending on the size of the respective sub-collections and the magnification factor. The placement of individual documents that are realized as stars depends, as explained above, on their similarity (see Figure 2.28).

2.3.2 Themescapes

In contrast to the galaxies, themescapes display an abstract three-dimensional landscape of information. The higher an elevation, the higher the theme strength is rated. Additional
2.3 Semantic Similarity Maps

Figure 2.26: IN-SPIRE system (left) and SPIRE (right) displaying a galaxy [Lab04].

Figure 2.27: Visualization Islands included in the xFIND system [ASL+01].
encoding by various colors supports the recognition process. Peaks are easy to detect and interesting characteristics become visible in ridges and valleys. The SPIRE system includes galaxies (see figure 2.26) as well as themescapes (see Figure 2.29). In this system, the themescape visualization is based on the previously mentioned galaxy, where relevant keywords from the documents visualized in the galaxy are extracted and visualized as topics in a landscape.

A very early but typical example of thematic landscapes can be found in the Bead system [Cha93], [Cha96]. In contrast to hierarchically-based systems like InfoSky, the Bead system works on flat document repositories. The information space is arranged on the basis of inter-document similarity to form a 3D landscape in a very simple implementation, which makes it easy for users to navigate around the information landscape. An example is shown in Figure 2.30.

Another example of themescapes is offered by the Cartia company [Car04]. At this company, Relational Topic Mapping (RTM) software is developed. This describes the technology used to automatically extract the content of unstructured text and organize it onto interactive maps of information. Figure 2.31 gives an insight into the work of Cartia.
Figure 2.29: Themescape displayed within the SPIRE system [Lab04]
Figure 2.30: Thematic Landscape within the Bead system, constructed from articles of an HCI conference, CHI’91 [Cha93]
Figure 2.31: Cartia Themescape [Car04]
2.3.3 Kohonen Maps

Kohonen maps are computed using a different algorithm from the ones described above. The basis for this computation is a neural network. The algorithm used is called SOM (Self-Organizing feature map algorithm) and was developed by T. Kohonen [Koh97]. As a result of the Kohonen algorithm, the entire document set is broken down into its main contents (e.g. keywords taken from titles and abstracts) and can be mapped onto the nodes of a two-dimensional grid. Figure 2.32 shows a Kohonen map displaying 32,627 articles from the Usenet newsgroup collection sci.lang (from June 95 to March 97) [Koh99].

![Kohonen map displaying 32,627 articles from a newsgroup collection](image)

**Figure 2.32:** Kohonen map displaying 32,627 articles from a newsgroup collection [Koh99]
2.4 Table-Based Visualizations

The categorized visualization of text in tables is a well-known and familiar technique. In contrast to a simple list presentation, where all information is usually provided without any particular order, the clearly structured and obvious layout of a table improves the possibility for comparing data sets with respect to single meta-data. This comparison capability is supported by the sort feature as well as the possibility of changing column locations. Although business graphics make extensive use of tables, their use in the field of information visualization is still limited - dependent on the very low application of visualizations. Nevertheless, combining the advantages of a table with information visualization techniques can lead to a high performance by the user in fulfilling his task. Focus and context, distortion, or semantic zoom are only a few catchwords that describe the opportunities. However, only a small set of visualizations are known that work with this approach. A selection will be shown with the following examples.

Figure 2.33: Infozoom system from humanIT Software GmbH [SBB96]

As the commercial successor of the research project Focus [SBB96], Infozoom \(^8\) is a system that strongly influenced the development of the SuperTable, which is included in

\(^8\)http://www.humanit.de/deu/infozoom/index.htm
the VisMeB framework. The preparation of data within a table using focus and context enables the presentation of a large number of datasets in a small space. Contents of cells are merged and sorted and therefore can be explored by zooming in and out using a simple mouse-click. Thus, scrolling is not necessary. Figure 2.33 is an example from the area of e-commerce, showing an eCatalogue Management.

A project very similar to the Infozoom system is Inxight’s TableLens [RC94], included in the VizServer framework. In contrast to the horizontal fusion in Infozoom, the TableLens merges rows instead of columns, i.e. it implements a vertical fusion of cells. Again, focus and context is used to avoid the need for scrolling or making all the data fit into a single screen. Figure 2.34 shows data about the Superbowl 2001. It can be used to predict the winner of the Superbowl.

![TableLens with data from the Superbowl 2001](http://www.cs.umd.edu/hcil/date lens/)

An obvious application for a table-based layout is calendar software. To achieve a good overview, weekdays are organized in columns, time or time ranges in rows. This enables a very fast access to important data. The DateLens ⁹ [BCCR04] combines this common paradigm with a zoom function that enables the user to magnify important data without losing the context. Adapting the amount of information to the available space can be done independently from the device used, i.e. whether a PDA, a Tablet PC, or a desktop computer. Figure 2.35 shows the months June, July, and August 2002 with a focus on the 17th of July. The use of this semantic zoom technique leads directly to the next section, which deals with this subject.

### 2.5 Semantic Zoom

Zooming in a visualization system is a common and almost unalterable technique. But the simple effect of enlarging a chosen area is just one side of the coin. The second one is

⁹[http://www.cs.umd.edu/hcil/date lens/]
semantic zooming. Additional information is provided, far removed from a mere change of objects’ size. Items can modify their shape, or even their very representation in the display. Depending on the zoom factor, information is displayed or not. This can extend from a small label or text passage to a completely new layout of the object. Obviously, a combination of geometric and semantic zoom is possible and meaningful in specific situations, e.g. while using a map. A more detailed description of this topic is given in Chapter 4.

Common systems providing the opportunity to include a semantic zoom are Pad [PF93], Pad++ [BPM+04] (see also 4.3), Jazz [BMG00], [BGM04], and Piccolo [BGM04]. They allow the user to create zoomable user interfaces and thus offer great versatility. The possibility of e.g. Jazz or Piccolo to create robust, full-featured graphical applications in Java and C# enables users to build their own visualizations, including visual effects such as zooming, animation and multiple representations.

Silver2 [LMC+04] is a video-editing software that includes a semantic zoom feature.
to explore more than one location of the same timeline in detail. A combination of scale-based and semantic zooming is used to enable the user to view video at multiple temporal resolutions as well as at multiple levels of detail. To change these properties, a slider providing a continuous sliding mechanism has to be moved (see Figure 2.36).

Whenever different levels of detail are given, the use of semantic zoom seems to be self-evident. We saw examples from the field of video editing, website design or simply hierarchically-clustered data. Another important application domain is genomic research. With Sockeye [MAB°04], a system is provided that enables users to assemble, visualize and work with complex comparative genomic datasets. All this is done in an interactive 3D graphical workspace. Long genomic regions can be displayed, as well as short base pairs. Sockeye can automatically switch from an “individual features” view to a ”compressed semantic zooming” view for this given feature. Figure 2.37 shows two different views at different levels of detail.

Another application from the field of bioinformatics is TrendDisplay [BG°03]. It was built to analyze large amounts of data from high-throughput screening runs as an approach to drug discovery. The combination of so called micro and macro views implements the focus and context idea. It enables the display to present the data at multiple levels of detail, as well as in context. Distortion-oriented magnification as well as semantic zooming is used to implement the idea. Figure 2.38 shows a typical timeline visualized within the TrendDisplay system.

DENIM [LNHL°00] provides a similar mechanism for navigating website designs and it served as an inspiration for the Silver2 system. A study of website designers at their daily work showed that designers use different levels of refinement, site map, storyboard

---

10 http://www-2.cs.cmu.edu/silver/publications.html
2.5 Semantic Zoom

Figure 2.37: Sockeye system showing all genes on a 151MB chromosome (a) and the individual genes (b) in a 200Kb region around one of the four histogram peaks in the image to the left. As the new region is shorter than the 300Kb threshold for genes, Sockeye automatically queries in and shows individual genes [MAB+04]

and individual page. Sketching at an early development stage is therefore an important part of the work and is performed at all stages. DENIM supports this sketching behavior and provides a semantic zoom function to change the level of refinement. Figure 2.39 shows a screenshot.
Figure 2.38: TrendDisplay presenting a timeline with different levels of detail [BG03]

Figure 2.39: The DENIM system supporting web site designers by providing different levels of refinement as can be seen on the left border [LNHL00]
2.6 Multiple Coordinated Views

Multiple Coordinated Views (MCVs) play a large and important role in the context of this thesis. Because of their great significance, the topic is split into three different parts. The first part, described in this chapter, gives an overview of actual systems implementing MCVs. The second part, presented in chapter 3, provides the formal background of MCVs. It includes a definition of MCVs and introduces an enhanced three-phase model specifying the design process. So far, only two phases are mentioned in the literature (see [WWK00]), but an important step is missing and will therefore be added in this thesis. Finally, all the different views used in the VisMeB framework are presented in detail. The last part, which is the topic of Chapter 5, deals with the interaction between the views used. The relationship categories introduced below are explained in detail. Afterwards, all interactions implemented within the VisMeB framework (including interactions between different views as well as the use of filters) are investigated. Because of the strong connection between the granularity concept used and the relationship of views, the granularity concept has to be introduced first, i.e. in Chapter 4, which leads to a further reason for splitting the topic of MCVs.

The comprehensive scope of the MCVs made it necessary to divide the content into these three parts: First, give an overview of which systems are currently used i.e. a good introduction. Then, provide the theoretical background for understanding how and why views are used. And last, investigate the implementation in the current context, i.e. the views and their relationship within the VisMeB framework. This makes it easier to focus on specific aspects.

Nowadays, a hybrid form of the visualizations presented so far is widespread and meaningful. Depending on the complete environment, described e.g. by the 5-T environment (see Chapter 3), one or other of these visualizations is better for reaching the desired goal. However, a combination seems to be unavoidable. The large number of systems using multiple coordinated views (MCVs) bears out this assumption. Because of this mass of applications, only a small insight into the field of MCVs can be given. To achieve a better overview, the systems presented will be distributed into different categories, depending on the kind of interaction that is mainly used:

1. **Selection**: Users select (or highlight) an item to show their interest. This usually leads to a further action.

2. **Navigation**: By navigating the visualization (e.g. scroll, pan, zoom, etc.), the user is able to focus on specific data or display other data.

A detailed insight into this topic is given in Chapter 5, so no detailed description will be given at this point to avoid redundancy.
2.6.1 MCVs Realizing The Select ⇔ Select Relationship

A very common application domain for MCVs can be found in the field of medical care. Aigner and Miksch [AM04] present the CareVis \(^{11}\) system to support protocol-based care in medicine via multiple coordinated views. It provides three different views: a Logical View, a Temporal View, and a QuickView Panel (see Figure 2.40). Interaction between the logical and the temporal view is implemented via brushing and linking, i.e. selecting an element in one view selects the corresponding element in the other views. Additionally, a coordinated navigation is implemented but without automatic synchronization. The developers did not use this degree of automation, which means that synchronization is via drag and drop, triggered by the user. Therefore, the system does not fit in the category Select ⇔ Navigate.

Robert et al. [HPR04] propose a combination of two- and three-dimensional scatterplots as simultaneous visible views \(^{12}\). Furthermore, 2D and 3D views are linked together to allow interaction in 2D with feedback in 3D. Additional support is provided by depth cues (color and point size) to impart a better depth impression for the user. Further information is given by 2D and 3D histograms to highlight the point distribution and density inside and outside the spatial focus. Problems in terms of perception and interaction like overplotting, which makes it impossible to see all points, or displaying point clouds in 3D, will be reduced by this system (see Figure 2.41).

Combinations of scatterplots are very common - to give an overview of the data space - and visualizations that provide a detailed view, whether they include textual or graphical

---

\(^{11}\)http://hildegard.asgaard.tuwien.ac.at/projects/carevis/

\(^{12}\)http://www.vrvis.at/vis/research/2d3dscatterplots/
Figure 2.41: The combined 2D and 3D scatterplot with user interface, implemented in the VRVis project [HPR04]

parts. Craig and Kennedy [CK03] present an application for the visual exploration of microarray time-series data. The system enables the user to visually formulate and modify measurable queries with separate time-period and condition components. Users can formulate and modify the queries with rapid reversible displays in the traditional “value against time” graph format. As an example, scatterplot and graph views are coordinated via the Select ⇔ Select relationship, which results in labeling and highlighting the corresponding elements. A complete overview of the interface is given in figure 2.42.

Another example from the domain of microarray visualization is the TimeSearcher [SBZ03], see Figure 2.43. Researchers are often interested in finding genes with similar expression profiles over time. TimeSearcher can be used for finding patterns in linearly ordered sequences. In a practical application at the University of Maryland’s Department of Cell Biology and Molecular Genetics, TimeSearcher has been used to explore the occurrence frequencies for short nucleotide sequences in large sets of aligned sequences. TimeSearcher was successfully involved in identifying patterns that occur frequently in some positions, and infrequently in others. The system was implemented using the [BGM04]
Piccolo Toolkit for Zoomable User Interfaces.

Systems like the XmdvTool [MW95], XGobi [SCB98], the Exploratory Data Visualizer (EDV) [Wil96], Visage VQE [DRK97], the Attribute Explorer [TSWB94], LinkKit [Nor98], or the Navigational View Builder [MFH95] provide very similar approaches while connecting views by brushing and linking. Because this is a standard technique nowadays it is implemented in nearly all systems using multiple coordinated views. Very often the differences lie in the application domain and the corresponding visualizations that are integrated. Mirage [Ho04] is an application using multiple coordinated views to visualize images and multi-dimensional numerical data from an arbitrary domain of study. It includes projected images of points, point classes, or proximity structures in one, two, or higher dimensional subspaces, in linked views of tables, histograms, scatter plots, parallel coordinate plots, graphs, and trees.

The last system presented in this context is the Hierarchical Cluster Explorer [SS02]. It implements four general techniques for supporting the exploration of clustering algorithms:

1. Overview of the entire dataset, in combination with a detail view,
2. Dynamic query controls to limit the number of clusters,
3. Coordinated displays, and

Figure 2.44 shows a typical layout of the Hierarchical Cluster Explorer.
2.6  **Multiple Coordinated Views**

![Figure 2.43](image1.png)

**Figure 2.43:** TimeSearcher visualizing time-series for genetic research [SBZ+ 03]

![Figure 2.44](image2.png)

**Figure 2.44:** The Hierarchical Cluster Explorer [SS02]
2.6.2 MCVs Realizing The Navigate ⇔ Navigate Relationship

The possibility e.g., for working with synchronized scrolling enables the user to compare documents, source code pieces, or even images in an intuitive but effective way. While considering two different views of the same data and being able to explore and interact with it simultaneously, there is an opportunity to find either similar or completely dissimilar patterns. Very often this kind of synchronization is implemented via scroll bars that are tightly coupled. The Logos Bible Software [Kos05] provides an electronic library that includes diverse bible translations as well as commentaries concerning every single passage. Navigation is possible by a hierarchical structure that divides the entire text into book, chapter, and verse. By selecting the corresponding source, the user is able to scroll simultaneously through e.g. one bible translation and one commentary on the respective part, as can be seen in Figure 2.45.

![Figure 2.45: The Logos Bible Software providing a hierarchical structure for navigating on the left, a bible translation on the top right, and the corresponding commentary on the bottom right [Kos05]](image_url)

Corel’s WordPerfect [Cor05] allows the user to see a formatted text at the same time as the document’s formatting code, and to scroll through both displays concurrently. Thus, it is easy to keep the relationship between the two representations without being

---

13 http://www.logos.com/
forced to scroll the windows one after the other. An example is shown in Figure 2.46.

Figure 2.46: Corel’s WordPerfect displaying a formatted text (top) and the corresponding formatting code (bottom) [Cor05]

With DEVise (Data Exploration and Visualization) \(^{15}\), a data-exploration system is implemented that enables users to pan and zoom diverse 2D scatterplots with common X and Y axes [WL00]. Hence, a synchronized navigation through various plots is possible and eases the exploration of visual presentations of large tabular datasets from various sources (see Figure 2.47).

The Augur visualization tool [FD04] is an extension of the Seesoft application, introduced by Stephen Eick and colleagues [BE96]. It provides a line-oriented view of documents like source code files. Different colors encode specific properties of the respective lines and make it easy to compare various versions of the same file. A typical example is the CVS Activity Viewer included in Augur which enables the comparison of different CVS revisions. The development of software in a team that is distributed over a wide area is a very realistic case that exemplifies the significance of such a helpful appli-

\(^{15}\)http://www.cs.wisc.edu/devise/
Figure 2.47: DEViSe system displaying two scatterplots that can be navigated synchronously [WL00]

Figure 2.48: CVS Activity Viewer of the Augur system, visualizing differences by the use of light and dark blue color [FD04]
2.6.3 MCVs Realizing The Select ⇔ Navigate Relationship

Typical examples using the Select ⇔ Navigate relationship can be found in many common user interfaces. For instance, two combined views showing a table of content in one frame and a detailed view in the other e.g. in online help applications, file browsers or websites. The most widespread and familiar application is probably the Windows Explorer. However, this scenario is only one possible realization of the concept. Apart from the pure text-based approach, a large variety of visualizations are used to implement this concept.

A very prominent example is the Visible Human Explorer [NSP97]. The human body can be explored via diverse views that allow a quick navigation through the whole body. Cut lines in the overview determine the 2D cross-section visualization in the neighboring view. An additional table of contents (see Figure 2.49, left side) gives an overview of the human parts that it is possible to visualize.

![Figure 2.49: The Visible Human Explorer using a combined textual and visual approach [NSP97]](image)

In most cases, all views are assigned to a single main window that includes the application. Another approach is to open up a pop-up window to present the details for a chosen item. This can be found e.g. in the FilmFinder [AS94b]. Clicking on a data point in the scatterplot opens up an additional window containing all the information corresponding to the selected item. The Generic Genome Browser [SMS+02] is a Web-based application for displaying genomes and genomic annotations. It includes the possibility
of zooming and scrolling through arbitrary regions of a genome; even a semantic zoom is implemented. In the upper part, an overview is given (emphasized by a light blue background), whereas the lower part (white background) presents the detailed information appropriate to the section selected above (see Figure 2.50). A user’s private annotations can be uploaded and published to the community to support working together in a fluent, interactive way.

Figure 2.50: Generic Genome Browser displaying the detailed view after zooming out to 200 kb, showing semantic zooming [SMS+02]

Systems like the Information Mural [JS98] create a reduced version of an information space like documents or source codes in order to navigate through the original source. Visual attributes like color, intensity and pixel size in combination with anti-aliasing compression techniques are therefore used. This enables a broad and quick overview of the source’s real appearance.

Instead of traditional sequential menus, [HS00] use Simultaneous Menus as an alternative arrangement. Users are able to select items from diverse overviews, which leads to the display of the results in a detailed view. The results of a between-subject comparison between traditional sequential and simultaneous menus suggest "that appropriate use of

simultaneous menus can lead to improved task performance speeds without harming subjective satisfaction measures”. A good application domain for this application may be the one of Web design. Figure 2.51 shows an example screen.

Figure 2.51: Simultaneous Menus to improve the task-performance speed [HS00]

2.7 Summary

This chapter gave an overview of the field of information visualization. Important terms such as Information Visualization, Scientific Visualization, and Meta-Data were defined to create a basis for understanding the context. Visualization techniques that are implemented in the VisMeB framework were described, as well as systems that use similar approaches. To limit the amount of information to manageable proportions the State-of-the-Art analysis was restricted to a selection of relevant applications that are directly related to this thesis. Tables 2.7 and 2.9 provide a complete overview of the techniques and applications presented.
Table 2.7: Interaction techniques used in the VisMeB framework

<table>
<thead>
<tr>
<th>TECHNIQUE</th>
<th>SHORT DESCRIPTION</th>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DYNAMIC QUERIES</strong></td>
<td>Manipulate the visual display directly by e.g. sliders</td>
<td>2.1.3.1</td>
<td>14</td>
</tr>
<tr>
<td><strong>BRUSHING &amp; LINKING</strong></td>
<td>Connect various views of the same data set. A highlighting or selection of a data subset (&quot;Brushing&quot;) in one view affects on other views &quot;Linking&quot;</td>
<td>2.1.3.2</td>
<td>15</td>
</tr>
<tr>
<td><strong>MOVABLE FILTERS</strong></td>
<td>Move a filter across the display to modify a view. This can result in e.g. filtering, adding additional information for underlying objects, or enlargement</td>
<td>2.1.3.3</td>
<td>16</td>
</tr>
<tr>
<td><strong>OVERVIEW-PLUS-DETAIL</strong></td>
<td>Connect two views by displaying an overview to navigate, and a detailed view to increase the level of information</td>
<td>2.1.3.4</td>
<td>17</td>
</tr>
<tr>
<td><strong>FOCUS &amp; CONTEXT</strong></td>
<td>Similar to Overview-Plus-Detail, but the two views are unified in a single one. To achieve this, distortion is used</td>
<td>2.1.3.5</td>
<td>18</td>
</tr>
<tr>
<td><strong>PANNING &amp; ZOOMING</strong></td>
<td>Zoom in to reach a magnified view and pan by moving this magnified clip area around to explore the surroundings</td>
<td>2.1.3.6</td>
<td>19</td>
</tr>
</tbody>
</table>
### Table 2.8: State-of-the-Art Scatterplot Applications

<table>
<thead>
<tr>
<th>2D Scatterplots</th>
<th>Geographical Scatterplots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Homefinder</td>
<td>[WS92]</td>
</tr>
<tr>
<td>SeeNet</td>
<td>[BEW95]</td>
</tr>
<tr>
<td>Spotfire</td>
<td>[Ahl96]</td>
</tr>
<tr>
<td>Open Visualization Data Explorer (OpenDX)</td>
<td><a href="http://www.research.ibm.com/dx/">http://www.research.ibm.com/dx/</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scatterplots for Abstract Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>FilmFinder</td>
</tr>
<tr>
<td>Interactive Timeline Viewer (ItLv)</td>
</tr>
<tr>
<td>Envision</td>
</tr>
<tr>
<td>Search Result Explorer of XFind</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scatterplot Matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polaris</td>
</tr>
<tr>
<td>ClusCorr98</td>
</tr>
<tr>
<td>VRVis Scatterplot Matrix</td>
</tr>
<tr>
<td>HyperSlice</td>
</tr>
<tr>
<td>Spreadsheet Framework</td>
</tr>
<tr>
<td>ArcMap</td>
</tr>
<tr>
<td>Influence Explorer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3D Scatterplots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Visualization Data Explorer (OpenDX)</td>
</tr>
<tr>
<td>NIRVE</td>
</tr>
<tr>
<td>Webwinds</td>
</tr>
<tr>
<td>Voxelplot</td>
</tr>
</tbody>
</table>
Table 2.9: State-of-the-Art Applications for Semantic Similarity Maps, Table-base Visualizations, and Semantic Zoom

<table>
<thead>
<tr>
<th>Semantic Similarity Maps</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Galaxy of News</td>
<td>[Ren94]</td>
<td>2.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Galaxies</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IN-SPIRE</td>
<td>[Lab04]</td>
<td>2.3.1</td>
</tr>
<tr>
<td>SPIRE</td>
<td>[TCK+01]</td>
<td>2.3.1</td>
</tr>
<tr>
<td>Visualization Islands</td>
<td>[ASL+01]</td>
<td>2.3.1</td>
</tr>
<tr>
<td>InfoSky</td>
<td>[AKSG04]</td>
<td>2.3.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Themescapes</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SPIRE</td>
<td>[TCK+01]</td>
<td>2.3.2</td>
</tr>
<tr>
<td>Cartia</td>
<td>[Car04]</td>
<td>2.3.2</td>
</tr>
<tr>
<td>Bead</td>
<td>[Cha96]</td>
<td>2.3.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KohonenMaps</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Websom</td>
<td>[Koh99]</td>
<td>2.3.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table-based Visualizations</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Infozoom</td>
<td>[K]://www.humanit.de/deu/infozoom/index.htm</td>
<td>2.4</td>
</tr>
<tr>
<td>TableLens</td>
<td>[RC94]</td>
<td>2.4</td>
</tr>
<tr>
<td>DateLens</td>
<td>[K]://www.cs.umd.edu/hcil/datelens/</td>
<td>2.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semantic Zoom</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pad</td>
<td>[PF93]</td>
<td>2.5</td>
</tr>
<tr>
<td>Pad++</td>
<td>[BPM+04]</td>
<td>2.5</td>
</tr>
<tr>
<td>Jazz</td>
<td>[BMG00], [BGM04]</td>
<td>2.5</td>
</tr>
<tr>
<td>Piccolo</td>
<td>[BGM04]</td>
<td>2.5</td>
</tr>
<tr>
<td>Silver2</td>
<td>[LMC+04]</td>
<td>2.5</td>
</tr>
<tr>
<td>DENIM</td>
<td>[LNHL00]</td>
<td>2.5</td>
</tr>
<tr>
<td>Sockeye</td>
<td>[MAB+04]</td>
<td>2.5</td>
</tr>
<tr>
<td>TrendDisplay</td>
<td>[BG03]</td>
<td>2.5</td>
</tr>
</tbody>
</table>
Table 2.10: State-of-the-Art applications using Multiple Coordinated Views

<table>
<thead>
<tr>
<th>MULTIPLE COORDINATED VIEWS</th>
<th>SELECT ↔ SELECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CareVis</td>
<td>[AM04]</td>
</tr>
<tr>
<td>2D3D-Scatterplots</td>
<td><a href="http://www.vrvis.at/vis/research/2d3dscatterplots/">http://www.vrvis.at/vis/research/2d3dscatterplots/</a></td>
</tr>
<tr>
<td>Microarray Visualization</td>
<td>[CK03]</td>
</tr>
<tr>
<td>TimeSearcher</td>
<td>[SBZ^03]</td>
</tr>
<tr>
<td>XmdvTool</td>
<td>[MW95]</td>
</tr>
<tr>
<td>XGobi</td>
<td>[SCB98]</td>
</tr>
<tr>
<td>Exploratory Data Visualizer</td>
<td>[Wil96]</td>
</tr>
<tr>
<td>Visage VQE</td>
<td>[DRK97]</td>
</tr>
<tr>
<td>Attribute Explorer</td>
<td>[TSWB94]</td>
</tr>
<tr>
<td>LinkKit</td>
<td>[Nor98]</td>
</tr>
<tr>
<td>Navigational View Builder</td>
<td>[MFH95]</td>
</tr>
<tr>
<td>Hierarchical Cluster Explorer</td>
<td>[SS02]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAVIGATE ↔ NAVIGATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logos Bible Software</td>
</tr>
<tr>
<td>WordPerfect</td>
</tr>
<tr>
<td>DEVise</td>
</tr>
<tr>
<td>Augur</td>
</tr>
<tr>
<td>Seesoft</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SELECT ↔ NAVIGATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visible Human Explorer</td>
</tr>
<tr>
<td>Generic Genome Browser</td>
</tr>
<tr>
<td>Information Mural</td>
</tr>
<tr>
<td>Simultaneous Menus</td>
</tr>
</tbody>
</table>
3.1 Introduction

By presenting actual systems using MCVs, we have taken a first step into this domain. To provide a better understanding of the "How?" and "Why?", a formal background has to be created, which will now follow.

Result-set presentation with the help of visualizations is becoming a more and more common technique (see e.g. [Shn98]). [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 and improve the possibilities offered by a single visualization. In recent years MCVs have become more and more popular. A lot of systems use this approach to provide a better access to the mass of data users are confronted with. Nevertheless, there are drawbacks that have to be weighed up. The use of MCVs can be significant, but only if the advantages clearly outweigh the disadvantages. The design decision to use MCVs for search result visualization is a very important and far-reaching one. Not only do the visualizations have to be chosen, but also the coordination between them, which has wide-ranging consequences for the design of the final system. [WWK00] present eight guidelines for the design of multiple coordinated views. These can be taken as a basis for a process of deciding whether to use MCVs or not. The guidelines are organized in two main sections; the first part deals with the topic of when multiple views are preferable, i.e. it supports the designers in coming to a decision. Part two engages in the use of MCVs, i.e. the costs (like space used, cognitive attention, etc.) that arise when working with MCVs should be minimized. Unfortunately, trade-offs exist among the rules; this is common in reality but it is even more important that it be investigated here. To avoid misunderstandings, a definition of the phrase "Multiple Coordinated Views" is given.

**Definition 3.1** 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 done by selecting and/or navigating.

[WWK00] divide the design process of MCVs into two phases, describing (1) When to use MCVs and (2) How to use MCVs. In this thesis the process is expanded to a three-phase model:

1. Decide if MCVs should be used or not
2. If MCVs are to be used, decide which visualizations have to be chosen
3. Define the layout and the interaction of the multiple visualizations.

In this chapter the focus will be on the first two phases, which are strongly related. The process of choosing an MCV approach and the interrelated visualizations should be made clear. The reasons for choosing exactly those visualizations that we selected are given and explained. Different views can exploit their advantages, whereas their drawbacks are corrected by the coupled view(s). Phase number three is only covered briefly and will be explained in detail in Chapter 5. This is because it has a very close relationship to the granularity concept, which has to be introduced before it in Chapter 4.

3.2 Phase 1: Acceptance Decision

In the first phase you have to decide if MCVs should be used or not. To come to this initial decision, the first set of guidelines previously mentioned can be of help:

- **Rule of Diversity**: Use multiple views when there is a diversity of attributes, models, user profiles, levels of abstraction, and genres.
- **Rule of Complementarity**: Use multiple views when different views bring out correlations and/or disparities.
- **Rule of Decomposition**: Partition complex data into multiple views to create manageable chunks and to provide insight into the interaction among different dimensions.
- **Rule of Parsimony**: Use multiple views minimally.

The problem with a single view is its limitation to only a few specific domains or tasks. Even if the data can indeed be applied to the visualization, its expressiveness can be very low or the cognitive overhead to comprehend and handle the output may be very high. Thus, the rule of diversity is one of the most important rules in choosing an MCV system or not. The diversity can be present in various characteristics, as there are:
3.3 **Phase 2: Choice of Visualizations**

- **attributes** e.g. abstract of document vs. measured relevance
- **models** e.g. hierarchical structure of nested links vs. thumbnail view of webpage
- **user profiles** e.g. non-expert users, expert users, company role (e.g. developer, manager, etc.)
- **level of abstraction** e.g. overview of the entire and detailed information
- **genres** e.g. block diagram vs. pseudo-code views of a source code

The simultaneous consideration of the same data from different views should be governed by the *rule of complementary*. Relationships that were not visible before when using a single view can be detected by a comparison of different visualizations. Switching from one visualization to another without being able to see both at the same time is cognitively demanding. Thus, a visual comparison in contrast to a memory-based comparison eases the work for the user. The concept of "divide and conquer" is well known in computer science. It is therefore not surprising that it is also relevant to information visualization. A division of complex data into multiple views allows the creation of manageable chunks to help understand the reduced mass of data more easily. Up to now, only rules that plead for the use of MCVs have been presented. But there is still one fact that must not be forgotten. Although the use of multiple views adds advantages to a system, it can lead to drawbacks concerning complexity. To learn a new visualization may be relatively easy for users, but to learn a variety of visualizations makes it harder to comprehend all the features and characteristics. The effort of context-switching increases, which likewise leads to a higher cognitive load. However, it is not only the user that can be confronted with disadvantages but also the system itself. Just think of the higher computational costs, or the display space to be provided. This has to be taken into account as well.

All these rules are very closely related to the task environment in which the finalized system is working, as well as the visualization itself. For example a specific task can be responsible for the meaningful use of MCVs, or the type of data, or the combination of various factors. Therefore it is important to take into account the environment, which leads to another supporting technique that is introduced in the second phase - the 5-T environment.

### 3.3 Phase 2: Choice of Visualizations

If the decision to use MCVs has been made, the second phase can start: choosing a specific set of visualizations to be used. This can be done applying the so-called "5-T environment" [Man02]. It deals with:

- Type of data
- Typical user
• Task
• Technical environment and
• Training

The type of data has a very strong influence on the choice of graphical representations. Numerical data e.g. can use completely different visual structures to be displayed than textual data. Or imagine hierarchical data, data concerning a date which can be presented using timelines, and so on. Another important fact is related to the number of data. If only a few items have to be visualized, additional information can be encoded e.g. by a glyph. Displaying thousands of items on a limited screen makes it very hard to detect single items because of clustering, overlapping, etc.

In most cases the typical user who will work with the application will differ from the developers. Therefore, the typical user has to be determined and the system has to be adapted. Although a specific user group can perhaps be defined, individual differences will still be there. Further information concerning this point can be found in e.g. [Shn98], [CY00], or [NS00].

The task is certainly also a very important factor influencing on the kind of visualization that has to be chosen. Some specific visualizations fit better to specific tasks than others, although it is very difficult - perhaps impossible - to define a general classification. Nevertheless, the effect of a visualization on a task strongly influences its success. To further explore this topic please see e.g. [Tud03], or [AA02].

Factors that are often neglected concern the technical environment. Most developers work an a high-end personal computer that provides a good platform for running the application fluently. However, there is a major difference between such PCs and the computers where the system will be installed for the end-user. Other restrictions depend on e.g. the network speed. Using a 56k modem will slow down the program execution in contrast to a T1 LAN connection.

Applications that seem complex on first sight often become more and more manageable if the user spends some time working with them. Therefore, a training period is absolutely essential. Depending on the degree of complexity, this can take from several minutes to months. So the designer has to assess if the effort needed to learn a new system is justifiable in terms of efficiency. Users tend to work faster with a familiar application than with a new one, although work could be done more efficiently after a training period that would compensate for the initial difficulties.

Depending on all these decision criteria, we have to find the appropriate visualizations.
3.3 Phase 2: Choice of Visualizations

The different 5-T environments that are used for the VisMeB system that has been introduced will therefore now be described (see table 3.1).

Table 3.1: Three different scenarios (5T₁–5T₃) and the corresponding requirements

<table>
<thead>
<tr>
<th>5T₁</th>
<th>TYPE OF DATA</th>
<th>TYPE OF USER</th>
<th>TASK</th>
<th>TECHNICAL ENVIRONMENT</th>
<th>TRAINING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Geo metadata available as XML- or HTML-files</td>
<td>Site planner; expert user familiar with business graphics and search behavior</td>
<td>Find a map providing information about a building site, using metadata</td>
<td>Standard office PC</td>
<td>A training period is reasonable, but use familiar visualizations to minimize its length</td>
</tr>
<tr>
<td>5T₂</td>
<td>Metadata for library media including administrative metadata</td>
<td>Administrator</td>
<td>Administrate and analyse metadata; find gaps, outliers, etc.</td>
<td>Standard office PC</td>
<td>A training period is reasonable</td>
</tr>
<tr>
<td>5T₃</td>
<td>Metadata describing media (e.g. DVD, CD-Rom, etc.)</td>
<td>Library user</td>
<td>Find a medium (e.g. DVD, CD-Rom, etc.)</td>
<td>Standard office PC</td>
<td>No training period desired; system has to be intuitive (ease of learning)</td>
</tr>
</tbody>
</table>

The first environment (5T₁) arose from the INVISIP project in which VisMeB was partially developed. A corresponding scenario was given in Chapter 1, Introduction. The latter ones (5T₂ & 5T₃) have their source in the MedioVis project, which is currently running in coordination with the University Library of Konstanz. Typical scenarios could look like this:

Scenario 2:

Maria Kramer is a 42 years old administrator working in the library of the University of Konstanz. Her job deals with the allocation, maintenance, and supervision of data describing the content of the media center (Mediothek) of the University library. Because data are often still entered by hand, the consistency is sometimes very poor. Today she has to check the entries that a student worker has created for newly-delivered DVDs. The student’s lack of familiarity with the field of library administration, resulting from his short working time (he has worked there for about three weeks) can lead to entries that are correct in form and content correct but located in the wrong category, or to simple careless mistakes like a missing meta-data. Thus, gaps and permutations can arise. Maria
starts the MedioVis application and enters just the search term "DVD" in the form fillin interface. As a result she gets 347 hits. By clicking on the table header "year" she sorts the entries by their date and is now able to explore the newest ones. While she skims over the data sets, she notices an empty cell in the table for the DVD "Metropolis": the language is missing. This can be explained by the fact that Metropolis is a silent movie. Nevertheless, the subtitles and menus are available in different versions e.g. in English, German, French, and so on. Maria inserts the missing meta-data there and then, and thus completes the fragmentary dataset. Because no other faults are detected, Maria closes the MedioVis application and ends here database session.

Scenario 3:
Daniel Beck is a student of media science in the third semester at the University of Konstanz. As homework, he has to write an essay about Charlie Chaplin and his most famous movies. He walks into the media center (Mediothek) of the University library, takes a seat in front of a desktop PC and starts the MedioVis application. The search terms he uses are "Charlie" and "Chaplin", no further restrictions are made so far. The first information he gets is an overview of all 20 movies in the Mediothek in which Chaplin is involved as actor or director. While he browses the result list, he notices "The Great Dictator", which he is very interested in because he has already heard about this movie. Daniel zooms in with the help of the "levels of detail" buttons. Thus, he will get more and more information, e.g. within the first step, a short description; on the next level, all the actors, the year of origin, and a poster; and on the highest level, even a short trailer that he looks at for a moment. Now he changes from the table view to the graphical view in the form of a scatterplot. To be able to get an impression of the movies’ importance, he changes the y-axis assignment to "Rating". Four movies seem to be very interesting because they have a much higher rating than the others. When Daniel explores these hits, he sees the following titles: City Lights, The Gold Rush, The Pilgrim, and The Great Dictator, which confirms his assumption of "The Great Dictator" being a very famous film by Charlie Chaplin. Fortunately, all four titles are currently available in the Mediothek. Daniel selects all four titles, sends the resulting list by mail to his own email-address and prints it directly from a printer in the library. With this information, he walks to the corresponding shelves, takes the movies and borrows them for a more detailed investigation during a comfortable video evening.

As a result of these varying environments, a modified table was created as a granularity-based visualization. Three (or even four) different versions are available: a LevelTable, a GranularityTable, and a MediaGrid. The fourth version, the GridTable, is an adaption of the LevelTable to the media center scenario for which the MediaGrid was originally implemented. Because of some small differences in layout and interaction it must be mentioned, but the MediaGrid is the real advance. A detailed description of layout and interaction opportunities is given in Section 3.5 and Chapter 5.
The connection between the first set of guidelines for MCVs and the 5-T environments leads to the following conclusions:

- Use different visualizations to give the user the chance to choose the appropriate one for each individual task.
- If a user is familiar with table presentations and business graphics, use a combination of both to support the information-seeking process.
- Use different levels of information to provide the whole data variety and reduce the number of visualizations to minimize the cost of context-switching.
- Display some visualizations sequentially (as alternatives), some simultaneously (e.g. to provide overview and detail at the same time)

In the VisMeB approach, the visualizations for presenting the result set are chosen from the following heap of visualizations, dependent on the corresponding 5-T environments:

- A modified table named "SuperTable", in different variants, dependent on the kind of granularity used, including task-constrained visualizations like bar graphs, tile bars, stacked columns, and relevance curves,
- A scatterplot, enhanced by a movable filter, available in a 2D and a 3D version,
- A document universe to show semantic similarity,
- A pie chart view, named "CircleSegmentView", to be used as a filter,
- A BrowserView, and
- A LocationMap.

The SuperTable and the tightly coupled (2D-) Scatterplot can be seen as the main visualizations created in the VisMeB framework. These two views are displayed simultaneously, dividing the available screen space into an upper part (SuperTable) and a lower (Scatterplot). By request, one of the two visualizations can fill the whole area, however the advantage of this combination is their coexistence. We will explain the added value in Chapter 5, when the coordination between the different views is described. Reasons for choosing exactly these two visualizations can be seen in Table 3.2.

The idea of the combination of these two visualizations is to give an overview (by the ScatterPlot) as well as a detailed view (by the SuperTable) at the same time. The previously mentioned visual information-seeking mantra [Shn98]: "Overview first, zoom and filter, then details on demand" is followed strictly. Users can very quickly find interesting data sets by choosing the related axis assignment in the Scatterplot. It is possible to zoom into specific areas or to filter specific data points. If one or more data sets seem to be
interesting, they can be explored in more detail in the SuperTable. Therefore, the granularity concept provides another advantage. The level of detail determines the amount of information in order to keep the data volume as small as possible.

The Document Universe can be used as an alternative to the Scatterplot, but focussing on another purpose. Every data set is drawn as a point in the 2-dimensional space. Position i.e. proximity to other data sets allows a conclusion to be drawn about the semantic similarity between them. In the Scatterplot, you can compare data sets by means of two dimensions. It can be seen as a 2-dimensional cut in the n-dimensional space of metadata. The Document Universe provides a more global view of the correlation between all data sets. A semantic similarity is computed by an LSA (Latent Semantic Analysis) algorithm. This leads to a layout in 2-dimensional space where the x- and y-axes are not assigned to a specific metadata, which is characteristic of the Scatterplot.

A closer look at the CircleSegmentView is given in Chapter 5 during the system’s filter description. For further exploration and a detailed description, see the applicable thesis of [Kle05].

The BrowserView is used to display large text passages. In the different granularity realizations, explained in the following chapter, 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, we can ignore this view.

The decision to use multiple coordinated views in dependency on the 5T-environment has been taken, the visualizations have been chosen, leaving the last step undecided - how to connect the different views.

### Table 3.2: Advantages and disadvantages of SuperTable and Scatterplot

<table>
<thead>
<tr>
<th>Advantage SuperTable</th>
<th>Disadvantage SuperTable</th>
<th>Advantage ScatterPlot</th>
<th>Disadvantage ScatterPlot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well structured</td>
<td>Only part of all data sets visible</td>
<td>Overview function because of representation of data sets as points</td>
<td>Small amount of information</td>
</tr>
<tr>
<td>Users are familiar with tables</td>
<td>Correlations between data sets hard to detect</td>
<td>Detect outliers, gaps and clusters easily</td>
<td>Very abstract</td>
</tr>
<tr>
<td>Goal directed search for single information possible</td>
<td>Overview in level 1 can be handled by ScatterPlot</td>
<td>Fast recognition of overall correlations</td>
<td>Comparisons possible only by means of two dimensions</td>
</tr>
</tbody>
</table>
3.4 Phase 3: Usage, Interaction and Layout

Now it is time to turn to the third phase, which often poses a hard challenge: *How will the multiple views be used? How will they interact? Where should they be placed?*

The second set of guidelines can lead us to a solution:

- **Rule of Space/Time Resource Optimization**: Balance the spatial and temporal costs of presenting multiple views with the spatial and temporal benefits of using the views.

- **Rule of Self-Evidence**: Use perceptual cues to make the relationships among multiple views more apparent to the user.

- **Rule of Consistency**: Make the interfaces for multiple views consistent, and make the states of multiple views consistent.

- **Rule of Attention Management**: Use perceptual techniques to focus the user’s attention on the right view at the right time.

In information visualization space is precious, as well as time. So the costs and the benefits of these properties have to balanced, and this balance is described in the *rule of space/time resource optimization*. The limitation of screen space and the patience of users while waiting for the results of a computation are factors that strongly influence the design of multiple coordinated views. Sometimes a more complex algorithm would lead to a better result, but it would take too long to execute it. The same situation can be found concerning the visual display, where a larger view might improve the insight, but the space is too small. Thus, we have to live with trade-offs. One decision associated with this topic is whether to show visualizations side-by-side or sequentially. An example in VisMeB is the simultaneous presentation of SuperTable and Scatterplot, whereas the DocumentUniverse or the 3D Scatterplot are only included as alternatives and are therefore distributed to different tabs.

Interactions between multiple views are very important and provide one of the bonus values of MCVs. Unfortunately, this correlation is not automatically visible to the user if no perceptual cues are given. A selection of e.g. a point in the Scatterplot without a selection of the respective row in the SuperTable would weaken the concept. Thus, it is very important to show the user how the visualizations are interrelated and what effect one action has on another one. The *rule of self-evidence* deals with exactly this topic.

Ease of learning is a very important feature, when using MCVs. To facilitate this process, the interfaces as well as the states of the views have to be consistent, as described in the *rule of consistency*. A display with e.g. a small region in one view and a different one in another must be deliberately created by the user (for instance overview and detail, or different views presenting different states), or should be avoided. Other mistakes can be
made by using inconsistent descriptions or labels, which leads to enormous confusion for the user.

Users react to perceptual cues, like a change of color, size, playing a sound, using an animation or a simple movement. These techniques should be used to channel the user’s attention to important situations encapsulated in the rule of attention management. In critical systems like safety-related control in a nuclear power station, or in the field of life-saving this point is crucial. A responsible person has to be able to detect critical situations with one view, enabling a very fast reaction.

All these rules are constructed using a very practical approach. Therefore, a short introduction to the theoretical background, describing some coordination models, will now be provided.

3.4.1 Coordination Models

Before the diverse VisMeB visualizations are described in detail, an overview of coordination models will be given. These models try to provide a formal and consistent approach to coordination. While many systems use coordination, in many cases they do not provide a coordination model. The following part will concentrate on three models, the Snap conceptual model [NCIS02] (Section 3.4.1.1), the View Coordination Architecture [PP01] (Section 3.4.1.2), and the Coordination Model for exploratory visualization [BR03] (Section 3.4.1.3), which are very commonly used in this context. The influence on and the design of, the conceptual model used in the VisMeB framework (Section 3.4.1.4) concludes this section.

3.4.1.1 Snap Conceptual Model

The Snap Conceptual Model ([NCIS02], [Nor00]) approach is focussed on the data itself. It is based on the relational data model. Components of a relational database are connected to display the effect of an interaction on changes to the related components. Therefore, a relation from the database (i.e. a table or query result) is displayed by a visualization. The linkage between two visualizations is realized via a join relationship of the underlying relation. Database concepts can be translated directly in user-interface concepts via the following functions:

- Relation $\rightarrow$ Visualization
- Tuple $\rightarrow$ Item in Visualization
- Primary Key $\rightarrow$ Item ID
- Join $\rightarrow$ Coordination
3.4 Phase 3: Usage, Interaction and Layout

Coordinations can be built without writing a line of program code. Therefore, the user first has to load relations into the visualizations. The different coordinations can be constructed between the respective visualizations afterwards. This can be called the "snap-them-together" step, from which the model gets its name. When establishing a linkage between two visualizations, the user is now able to define the action in each view that implements the tight coupling. This can be e.g. a selection or navigation action, both of which are addressed in Chapter 2 and will be explained in detail in Chapter 5. A typical example is the linkage within Windows Explorer - click a folder in the overview to display the files in another window. A schematic view of this example can be seen in Figure 3.1.

![Figure 3.1: Snap components: First, the relations are loaded into visualizations, then snapped together (adapted from [Nor00])](image)

3.4.1.2 View Coordination Architecture

The View Coordination Architecture is based on the Model-View-Controller pattern, known from the area of software engineering (see [GHJV93]). Specification and presentation, as well as the view model and the data model, are separated (see Figure 3.2). Coordination between distinct views is achieved via so-called "coordination components". If changes arise in an observed view, the corresponding effects in dependent views are initiated. To describe a coordination, at least the following facts are the minimum that are required:

- the source view to be observed,
- the action that invokes an effect in another view,
- the target view, and
- the reaction of the target view as a result of the source view’s action.
Coordinations can be composed and as a consequence are implemented as a set of directional coordinations. This includes the situation of a view being the source and the target view at the same time, within a composed coordination. Consequently, the effort of implementing and debugging expensive coordinations increases when the complexity of links and diversity of views rises. Figure 3.2 illustrates the cooperation of the different view coordination components.

![Figure 3.2: The View Coordination Model Architecture (adapted from [PP01])](image)

### 3.4.1.3 Coordination Model for Exploratory Visualization

To explain the coordination model for exploratory visualization in detail, a common language has to be used. Therefore, the *rudiments of coordination* have to be introduced [BR03].

- **Coordination entities**: They provide a detailed description of the coordination regarding actual window, view, data, record, tuple, attribute, parameter, process, event, function, graphic, or time.

- **Type**: The type is responsible for the link method between entities. Simple coordination is usually implemented by primitive types (e.g. integer, float, etc.), others by more complex data structures.

- **Chronology (lifetime of scheduling)**: The lifetime (or persistence of the coordination) determines how long the entities are coordinated, sometimes permanently, sometimes limited by their scope. Possible coordination modes are: synchronous, asynchronous, reactive, or proactive.
3.4 Phase 3: Usage, Interaction and Layout

- **Scope**: Depending on the scope some links can only be used in local areas, others are not restricted and are thus embedded in a global scope.

- **Granularity of links**: This specifies the number of entities in one coordination [2,...,n], the number of views in one coordination [1,...,n], and the number of links of an entity within a coordination [0,...,n] are specified.

- **Initialization**: Defines the method of creating a coordination, e.g. automatic, user-specific, scheduled, etc.

- **Updating**: Displays can be updated in different ways, e.g. by eager or greedy update, lazy update or user-initiated.

- **Realization (link realization, user control)**: This point covers different aspects. First, how does the user recognize the coordination? Is highlighting used, simple lines, or a formal layout mechanism? Second, how does the user control the linked information? Is it by direct manipulation or indirectly?

![Figure 3.3: Abstract model for coordination in exploratory visualization (adapted from [BR03])](image)

The main objects of the model for coordination in exploratory visualization are the **coordination objects**. They are responsible for the entity combinations and they monitor the coordinated views. Every coordination in the system is assigned to a special coordination object. Views that share the same coordination object can be called coordinated.
The complete set of coordination objects forms the *coordination space*. To describe a coordination, *translation functions* $f_{ij}$ are used to link the coordination object $CO_i$ and the view parameters $V_j$. Views have to be advised of a changed object and therefore have to be registered. Such an event - which is usually invoked by a user interaction such as selection - causes a change of the coordination objects. The coordination objects inform all coordinated views to adapt to the current situation. This is achieved with the information given in the coordination object and a new view is created using the translation function. A schematic view is shown in Figure 3.3.

### 3.4.1.4 The VisMeB Conceptual Model

All models introduced so far possess properties that are included in more than one model. The idea remains very similar, but the implementation differs in small ways. As an example, the underlying fundamental design of the View Coordination model, as well as the model for Exploratory Visualization, is based on the Model-View-Controller pattern. In addition, a strict separation of data and the corresponding visualization is one of the main aspects that is implemented in the VisMeB framework. The architecture of Snap and the model for Exploratory Visualization are both event-based and the coordination between views is achieved by the association of actions. To control the coordination and if necessary provide a translation mechanism for heterogenous data, an additional component has to be used. While the interaction with Snap and the construction of MCVs is very easy, the coordinations are restricted to the abstract navigation and selection actions defined in the Snap API. In the Exploratory Visualization model, the user is able to specify existing coordinations and integrate novel coordinations formally. This leads to the possibility of an early testing of the coordination design by programmers or users without implementing the coordination, if the construction is fulfilled visually.

In the VisMeB framework the underlying concept is implemented by the Model-View-Controller pattern [GHJV95]. Usually, this leads to a tripartition into a *data model* that administrates the data, a *view model* to visualize the data, and a *controller* that reacts to user input. Because of the strong relation between user input and visualization (direct manipulation), the view model and the controller were combined in a single component, the *view model*. Thus, the stringent separation between the data model and its visual presentation is still maintained. A complete overview of the implementation’s structure is provided in [Grü04]. The central package for the conceptual model of VisMeB is the *Views*-package. Apart from the single views like ScatterPlot or BrowserView, it includes the main window in which to run the complete program, the interface *View* which is implemented by all visualizations, and the *ViewNotifier* as the central component to implement the Multiple Coordinated Views concept.

In the current version, all visualizations can appear only once in the system, i.e. it is not possible to use e.g. two scatterplots. Thus, the *Singleton Pattern* [GHJV95] is used to guarantee that there exists only one single instance of an object. It enables a direct and
controlled access to a guaranteed constructed resource all over the program, without using global variables. If a singleton is used, you can be sure the object is valid for use - this contrasts with global variables where you cannot be sure if they are set or fully constructed. To implement the concept of Multiple Coordinated Views another pattern was used, the Observer Pattern [GHJV95]. The idea of this pattern is to notify and update all dependent objects if the state of a single object changes. Consequently, we speak about a one-to-many dependency of objects or - in the preceding case - more precisely, of views.

In dependence on the View Coordination Architecture, the coordination implemented in the VisMeB framework can be described by the following factors:

1. the source view, where an action is initiated,
2. the action that leads to a reaction in other views,
3. the administration object, which enables the correct processing,
4. the target view(s), that react(s) to the invoked action and
5. the kind of effect on the target.

Any action is carried out in a specific view, which directly suggests the combination of these two items into a single object, named StartObject later on. Analogously, the target view and the effect invoked by the StartObject are connected in the DestinationObject. The coordination between these two objects is managed by the AdministrationObject. Because of the use of the observer pattern, all views are automatically notified of any action, so the target space always stays the same, regardless of the kind of action. The StartObject is defined by user interactions like selection or focus. The space of available actions consists of the following possibilities:

- **repaint**: Is invoked if e.g. data in the table are sorted or general options are changed.
- **update**: Updates the number of visible data sets if e.g. a filter function is called.
- **focus**: Repaints and displays the focus initiated by a mouse-over effect.
- **selection**: Repaints and displays the selection initiated by a selection.

The central component of this coordination is the AdministrationObject, implemented in the system by the ViewNotifier. Changes in a view as described above are reported to the ViewNotifier, which informs all views about the current state. In the present architecture, actions and their effects are hard-coded in the system to obtain a consistent concept in accordance with the Rule of Consistency introduced above. On the one hand, this restricts the user in his decision freedom (What effect in View\textsubscript{i} takes place if an
Figure 3.4: Coordination model for coordination of multiple coordinated views in the VisMeB framework

Action$_j$ is carried out in View$_k$?), on the other hand unreasonable effects can be avoided. Figure 3.4 shows the coordination model for the VisMeB framework.

To ensure a consistent process, the registration of new views at the ViewNotifier is indispensable. This proceeding enables the views to be updated if any changes are made in other views. Therefore, the method ViewNotifier.attach(this) has to be called in the constructor of the corresponding view. If this registration is completed, changes in the view can be reported to, and changes in other views to react on can be received by, the ViewNotifier. As an example, we can imagine a selection of documents in an arbitrary view SourceView. As a consequence the function ViewNotifier.notifySelection(SourceView, SelectedDocs) is called to inform the ViewNotifier that a selection of the documents SelectedDocs in view SourceView took place. This results in a notification of all attached views by the function repaintSelection(id).

This concludes the theoretical background to multiple coordinated views. The focus will now shift to the visualizations implemented in the VisMeB framework. A short description of any view is given, as well as explanations of why only these are used.

3.5 The VisMeB Framework

Several visualizations are combined in the VisMeB framework (see also [LRKM03a]). Each of these has its own advantages and drawbacks and equally they work best for specific tasks. The combination allows a highly user-adapted work. This means that the user can choose the visualization that seems to fit best for him for the forthcoming job. For
3.5 THE VISMEB FRAMEWORK

instance, an overview is provided by the Scatterplot in a more intuitive way than by the SuperTable, but for detailed information the reverse is true. This clarifies the reasons for choosing exactly these visualizations. However, it does not force the user to use these visualizations in the anticipated manner. The MedioVis realization of VisMeB e.g. allows the direct movement of visualizations inside the screen. So if the Scatterplot is to be placed in the lower part, it just has to be dragged from the top and dropped onto the bottom. The different implementations will be explained in more detail later on: for the moment we will focus on the single views. Figure 3.5 gives an overview of the various visualizations and interrelations.

Figure 3.5: VisMeB Architecture

VisMeB evolved from a project called INSYDER \(^1\) (see [RMMH00], [RMM01], [Man02]). INSYDER was built to support small and medium sized enterprises in their daily work by finding business-relevant information in the Web. Depending on the typical users for this application, the decision was made to include business graphics in this visual information-seeking system. As a result a scatterplot, a bar chart, a table, a seg-

\(^1\)The project was funded by the European Commission under the Fourth Framework of the ESPRIT Program, Project No. 29232. www.insyder.com
ment view and a relevance curve visualization were included. All these graphical facil-
ities were adopted by the VisMeB framework, but in a modified way. In contrast to the
(mainly) sequential display of these views in INSYDER, VisMeB presents the visualiza-
tions integrated in a modified table, the SuperTable, or simultaneously as achieved by the
ScatterPlot. The bars found their way into the LevelTable (Levels 1, 2) and the Granulari-
tyTable (Levels 1-4). They represent the global document’s relevance and the relevancies
of single keywords (see Figure 3.6).

Figure 3.6: INSYDER visualizations integrated in VisMeB: The BarCharts (left) and the
RelevanceCurve (right)

Stacked columns as used in one SegmentView version of INSYDER are integrated
into the LevelTable (Level 4), and the TileBar version of the SegmentView was imple-
mented in the GranularityTable (Level 5) (see Figure 3.7).

The Relevance Curve can be found in Level 3 of the LevelTable (Figure 3.6), whereas
the Preview Window of INSYDER for showing html webpages (without images, thus
"preview") was implemented by the BrowserView (Figure 3.8). As a last visualization
the scatterplot could be integrated as a whole and is used in all SuperTable variants, the
LevelTable, GranularityTable, GridTable, and the MediaGrid (Figure 3.8).

Different visualizations, chosen in dependence on the application domain, were used
in the system. These include

- the ScatterPlot in a two- and a three-dimensional version,
- the MultiDataPointView, to handle the problem of data point overlapping in the
  ScatterPlot occurrences,
3.5 The VisMeB Framework

Figure 3.7: INSYDER visualizations integrated in VisMeB: The SegmentView, implemented with StackedColumns (left) and TileBars (right)

- the BrowserView to provide a view for displaying larger text parts,
- the DocumentUniverse, realizing a semantic similarity map,
- the LocationMap, to show a direct connection between the media and their physical location in the library,
- the SuperTable in its different versions, and
- the Visual Configurator for adapting the assignment of detail levels and corresponding visualizations realized in the SuperTable.

For the remaining chapter, the main focus will be set on the various SuperTable versions. Thus, detailed descriptions of the other visualizations are given in appendix A.
Figure 3.8: INSYDER visualizations integrated in VisMeB: The ScatterPlot (left) and the BrowserView (right)
3.5.1 SuperTable

Tables are a very common method for displaying metadata. This is undoubtly due to the fact of their clear structure. One data set is assigned to one row, all metadata are assigned next to each other. A comparison of data sets by means of a specific characteristic i.e. metadata can be done extremely fast if sorting of rows is done by means of selected occurences. The possibility of sorting and movinge columns into an order that fits the current task is another advantage. However, visualizations are rarely included and are often displayed simultaneously to add a higher level of information. The SuperTable tries to combine the familiar, but clinical, table presentation of text with attached graphical items that enable the user to gain more information. Different levels of detail admit a classification of abstraction layers. A granularity concept (see Chapter 4) defines these degrees. As an independent program, the Visual Configurator (see appendix A) allows the user or, in general, the system’s administrator to distribute the database contents to the available levels. Visualizations have to be assigned to metadata to define the cooperation between table and graphical items. Hence, the realizations presented below are only examples of possible assignments that seem to be meaningful. Nevertheless, they can be changed using the configurator.

Diverse versions of the granularity concept are implemented depending on the environment and the stage of development. The User Centered Design Process (see Chapter 6) that built a base for the system’s evolution enabled a continous improvement and further development of the VisMeB framework. Three real variants of the SuperTable (and the GridTable as an adaption of the LevelTable to a new scenario) are currently available and presented below; the LevelTable, the GranularityTable, and the MediaGrid as the most current and technically mature version. Before describing the individual visualizations, a short development history of the MediaGrid is used to give an insight into what was bequeathed from version to version and what was adapted. A more detailed description is given after the SuperTable presentation in Section 3.5.2.

3.5.1.1 Short History of the MediaGrid

Because a detailed delineation of the development history of the MediaGrid uses terms that are introduced in the following sections 3.5.1.2 to 3.5.1.5, this will be just be a short introduction to describe the interrelations.

The base system for the development of the MediaGrid was the INSYDER system, a visual information-seeking system for the Web. Diverse visualizations were offered, but only in parallel, not in combination. The first version of the SuperTable, which was named LevelTable, included a part of these visualizations in the table itself. This was implemented within the INVISIP project, which is strongly related to the domain of geographical meta-data (see also [SJFH02b], [SJFH02a], [GHG02]). Information was presented in different levels of detail to provide all information without leaving the context of the table. As a second version, the GranularityTable was introduced, including even more
visualizations in the table itself. Again, various levels of detail were used, but the way to change from one level of information to another changed slightly. The next step was to adapt these two tables in a way that allows the use of arbitrary databases, not restricted to web documents (like in INSYDER) or geographical meta-data (like in INVISIP). Thus, the project VisMeB was brought into being. Visualizations did not change, but the underlying data model did. To investigate the data independency, a new application domain was searched and found in the MedioVis project, which deals with the multimedia data of the media center of the University of Konstanz. Therefore, the LevelTable was adapted with small changes. To differentiate between these two LevelTable versions, the name GridTable was introduced. This was the last step before the MediaGrid came into being. Figure 3.9 shows an overview of the various development stages.

Figure 3.9: Overview of the development history of the MediaGrid

Before going into more depth, we should first consider the individual steps i.e. the various SuperTable versions, to provide a deeper knowledge and to make it easier to understand the reasons for creating the different visualization variants. Afterwards we will return to the history and give a detailed description of the single development steps.

3.5.1.2 LevelTable

The first version of the SuperTable was implemented as LevelTable. Four levels of detail are available to represent the range from overview to detail (see Figure 3.10).
3.5 **The VisMeB Framework**

![Figure 3.10: The four levels of the LevelTable](image)

Level 1 is restricted to a graphical display where only bars are shown. On one side, these bars indicate the relevance for the data set itself and on the other side for the entered keywords that usually start the information-seeking process. If no keywords are given e.g. if the system is used to explore the entire set of data without a limitation, no relevance can be computed. Every keyword receives its own color to serve as a unique identifier appearing in the different visualizations used in the framework. Column headers are given but, depending on the number of result data sets, the height of single rows is usually too small to show text.

In level 2 the situation changes. The rows are enlarged to display one line of text. Overlapping text is truncated to fit into the available cell. Bars are still visible and the additional information given by the denoted value confirms the bars’ length, standardized to a range from 0 - 100.

Level 3 introduces a new visualization, the “Relevance Curve”. It serves as a representation for the whole data set (e.g. a web document) that carries out a fragmentation into segments. Linked to the segment’s relevance, computed on the basis of keywords, a standardized value is calculated. If all these values are connected by a line the relevance curve is produced. Thus, a high or low curve amplitude at a specific position i.e. segment reflects the importance of this section. To recognize these height variations more easily, the rows are enlarged. This automatically leads to more space for displaying textual information in several lines, e.g. a short abstract. By then, truncated text often becomes visible in
Level 4 extends the relevance curve to a *Detailed Relevance Curve*. So far, only a basic segment importance is indicated by a high curve amplitude. Now a detailed specification can be done by splitting the relevance in partial relevances for the single keywords. Again, the colors initially assigned are used to draw colored bars representing the keywords’ relevance. This version of the detailed relevance curve uses stacked columns to visualize the importance. Another one, used in the GranularityTable, will use tile bars.

### 3.5.1.3 GranularityTable

The second SuperTable design variant, named GranularityTable, differs slightly from the LevelTable. It tries to achieve a smoother change of levels to give the impression of a steady process. In order to emphasize the continuous transition, the visualization is manipulated and adjusted by a slider. The granularity concept will be implemented as six different steps that you can choose between (see Figure 3.11). Four columns are used to show all the information: selection, visualization, text, and granularity. The visualization as well as the text column change their display from level to level, always providing more information than the previous level. The current selection is clarified by a checkmark in the first column.

Again, the first level will give an overview. Therefore, only a bar representing the data set’s relevance is drawn for each line. No further information is given. In level 2 the row’s height is enlarged to show a single line of text per data set. The data set’s relevance bar is split into the single keyword relevances, a check mark is set if the data set is selected, the text column displays the first written information (usually a title), and the slider for modifying the level of detail becomes visible.

Level 3 changes the fragmentation of the differently colored relevance bars to a vertical display, i.e. the length of the single keyword bars can now be compared to each other in an easier way than before. The text column is filled with more information, possibly another metadata. The kind of metadata is always defined by a prefixed descriptor, e.g. “title”, “URL”, or ”language”. This eases the allocation of text to the corresponding metadata. In level 4 the bars are extended by adding a label to each bar displaying the numerical value. More text is inserted into the respective column to provide more information.

Level 5 displays the previously introduced detailed relevance curve in a slightly modified form. The segments are arranged vertically instead of horizontally as seen before, implemented as *TileBars*. Each row represents a single segment, whereas the colors again indicate the relevance concerning the search terms. The more saturation, the more importance is implied. Usually, the whole data set, or at least an abstract, is now shown in the text column to get an almost complete overview of the data set.

In level 6 the two columns ”Visualization” and “Text” are merged to obtain more space. The complete data set can now be displayed in the combined columns.
3.5 The VisMeB Framework

Level 1

Level 2

Level 3

Level 4

Level 5

Level 6

Figure 3.11: The six levels of the Granularity Table
3.5.1.4 GridTable

The GridTable does not differ very much from the LevelTable presented above. It is adapted to the Mediothek scenario and thus introduces no significant changes. The intention was to meet the non-expert user group’s need to find information about data stored by the Mediothek of the University of Konstanz, i.e. comparable to a conventional catalogue search. Layout and features were restricted to an easy-to-handle interface without highly sophisticated features or visualizations. These differences reflect in e.g. the way the user can change the levels of detail. Numbering the levels would not make any sense to users when they do not know what is hidden behind these numbers. Therefore simple “+” and “-” buttons, supported by a graphical presentation as a kind of stairway, seemed to be more helpful. As a further modification, the names were translated from "LevelTable" to "Table View" and from "ScatterPlot" to "Graphical View", to give users a hint of the underlying visualization in words they are familiar with. Indeed, the different levels provide different information about the data sets in a manner comparable to the original LevelTable, but additional visualizations like the SegmentView or BarCharts were omitted to avoid possible confusion for the user. Figure 3.12 displays the GridTable in its original version.

![GridTable](image)

**Figure 3.12**: GridTable version of the SuperTable, implemented within the MedioVis project

3.5.1.5 MediaGrid

The latest implementation of the SuperTable is called MediaGrid. Again, diverse levels of detail are implemented. Depending on the underlying library scenario, the metadata are currently distributed to at most four levels. In contrast to the former table versions, the drill-down into detail levels can be done cell by cell, not just line by line or for the table as a whole. This leads to a distortion of table cells in the x- and y-dimensions instead of a linear distortion in the y-dimension resulting in a heightened row. In contrast to the Level- and the GranularityTable, it was decided not to incorporate built-in visualizations
like bars or relevance curves. This is due to the typical user working with the system within this scenario. A non-expert in the field of information visualization just wants to search for a specific media or to explore the whole data set to find an appropriate item. Usually, he/she is neither familiar with business graphics, nor does he/she want to pass through a long training period. For that reason, an intuitive and easy to learn interface has to be provided.

Unlike the previously introduced overview levels that display only bars without textual explanation, a visual presentation is abandoned in level 1 in the presented prototype. The most important information is therefore shown as text, restricted to one line per data set and at most seven columns. Information has to be clustered to fit to corresponding columns, e.g. "general information about the movie", "kind of media", "administrative information", and so on. Table 3.3 describes the metadata used and their distribution for the example shown.

<table>
<thead>
<tr>
<th>Table 3.3: Granularity Levels for the MedioVis Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LEVEL 1</strong></td>
</tr>
<tr>
<td>Language</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>LEVEL 2</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>LEVEL 3</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>LEVEL 4</td>
</tr>
<tr>
<td>-</td>
</tr>
</tbody>
</table>

In this case, the order and clustering of metadata is fixed. Changes could be made in different ways, e.g. by an administrator, or - to adapt the system to every single user - by observing a users’ behaviour, resulting in an automatic rearranging of cells. This leads to the so-called "Aspect of Interest (AOI)" that is described in detail in Chapter 4. We have to clarify the fact that the presented prototype is only an example of how the system could look. As we will see in Chapter 4, we are not restricted to a static layout. Quite the contrary, in fact; any metadata or visualization can be assigned to any column and any level.

A second version of the MediaGrid, named MovieVis, was implemented for research purposes. It was implemented as a standalone version, i.e. without any additional visualization. The idea was to combine the preceding LevelTable and GranularityTable variants
into a single system to test its potential. A lot of the ideas that had been developed were transferred and adapted to the movie scenario, which was intended to present an alternative to existing movie information systems like e.g. the Internet Movie Database \(^2\). The interface is very similar to the one used in the original MediaGrid version. As a first view, all data sets are displayed in a single line format (see Figure 3.13).

![Figure 3.13](image-url)

**Figure 3.13:** MovieVis first view after entering the query term "james"

Six columns are used, containing "Seen", "Title", "Direction", "Country", "Year", and "Length". Different levels of detail are available but without a strict separation by the addition of one extra piece of information per level. The transition is very smooth and thus the user can cope with the amount of data. Additional features include a history function, a table filter, and the possibility of starting a new query by just clicking on a link provided in the text passages (see Figure 3.14, third column, blue underlined link "Sean Connery" on top).

The result can be seen in Figure 3.15. This feature shows a very good integration of queries within the table context. A user can start a query, explore the results and directly formulate a new query without leaving the context of the table. It eases and accelerates the information-seeking process by making a context switch between query and result presentation unnecessary.

A table filter is provided as an additional row in the table located immediately below the table header. It is directly connected to the underlying column and affects only this type of metadata. If letters or words are entered (e.g. "Bond", Figure 3.16) all rows that do not contain the term are removed.

This enables a very fast filter mechanism without leaving the context of the table or introducing a highly sophisticated filter handling. The history function can be operated by two buttons on the upper left corner, displaying a "left arrow" (to go back in the history) and a "right arrow" to move forward in the history. The buttons are followed by a textfield to enter the query term. The search button "Suche" directly to the right or a simple "Enter", starts the query. Further interaction widgets deal with the level of detail.

\(^2\)http://www.imdb.com
3.5 The VisMeB Framework

Figure 3.14: The level slider is set to the extreme right and thus all rows are increased to the maximum. By moving a mouse over a term (e.g. "Sean Connery" in the third column) and clicking it, a new query can be started.

Figure 3.15: As a result of the new query, only movies with Sean Connery are displayed. Although there are still James Bond movies, additional ones are added.

management. The “up arrow” and “down arrow” buttons (as known from the LevelTable) enable the table row magnification or reduction respectively. Clicking the “down arrow” shrinks all rows to a single line presentation, whereas the “up arrow” effects a maximum magnification of all rows to make the complete content readable (see Figure 3.14). The small square in the upper right corner, as the last entry in the table header, provides the same effect as the “down arrow”, i.e. a kind of reset function to the start view. A slider between the two buttons directly influences the rows’ height and enables the user to magnify all rows without being bound to specific discrete levels (see Figure 3.17).

In addition, the magnification of single rows is possible, as it was in the GranularityTable. The only difference is the existence of just two extreme levels - a minimization to a “single line per row” presentation and a maximization to the highest level where the information is displayed as a whole. No intermediary steps are made by the buttons, only by the slider. Figure 3.18 displays such a situation.
Figure 3.16: The result set of MovieVis can be restricted by using a table filter. In the current situation the title is filtered by the term "Bond"

Figure 3.17: The slider located above the table enables a stepless magnification of all rows simultaneously. Thus, the rows’ height is completely dependent on the user and his slider settings.

3.5.2 Detailed History of the MediaGrid

Before turning to further implemented visualizations, it is appropriate to describe in detail the history that lead to the current version of the SuperTable in the form of the MediaGrid. As we saw above, a very long development phase preceded this state, including the two EC funded projects INSYDER and INVISIP. The first step was taken within the INSYDER project. Different visualizations, chosen in dependence on the application domain, were used in the system. These include

- the ResultTable (Figure 3.6),
- the RelevanceCurve, included in the ResultTable (Figure 3.6),
- the ScatterPlot (Figure 3.8),
3.5 The VisMeB Framework

The advantage of visualizations in supporting the information-seeking process is currently well known. Nevertheless, the way of presenting the information still differs from system to system. This variant was a first step in a direction that leads to a very efficient implementation of an information-seeking system. The proof of this statement is shown by evaluations which assessed the idea as good. Further improvements entailed an ongoing development phase, again including a lot of tests. With the exception of the relevance curve, which was included in the ResultTable directly, all visualizations coexisted in INSYDER. The screen was divided into two main parts, where the upper one contained the ResultTable, the BarChart, and the SegmentView visualizations, the lower one the BrowserView and the ScatterPlot. This variety helped in viewing the same data from different perspectives, but a hard context switch was necessary when changing from e.g. the table to the BarChart view. Thus, the main shortcoming of INSYDER was detected and an initial solution was sought.

All visualizations achieved within INSYDER could be adopted into the INVISIP project. Nevertheless, adaptions had to be made. The idea of providing various visualizations was retained, but the kind of presentation underwent a radical change. Although the application domain changed from a web search to a search on geo meta-data, the main concept stayed the same. The VisMeB framework, which became the internal project name for the system implemented for INVISIP, went one step further. While INVISIP was conceived for the visualization of geographical meta-data, the idea of VisMeB was to allow arbitrary databases and application domains to be examined. Therefore, the former IN-
VISIP project was integrated as one possible scenario for the VisMeB framework. As a result, a systematic revision of individual program parts and conceptual adaptions had to be fulfilled to cope with the generic tasks. The main difference from the INVISIP project was the strict separation of visualization and the underlying data model, following the Model-View-Controller pattern (see [GHJV93]). This enabled a parallel development of INVISIP and VisMeB because the visualizations used stayed the same, only the data model had to be adjusted. Thus, if we talk about INVISIP or VisMeB the only difference lies in the base data model, while the concept, the appearance, and the features do not differ. Its domain-independence is another milestone in the development of the SuperTable. Because the system itself is not restricted to a specific application domain, only small adaptions have to be made for each scenario. This can be done by the Visual Configurator, which will be presented in appendix A.

The first step in improving the INSYDER concept was to create the SuperTable. Its first version, the LevelTable, included hitherto separated visualizations in a single tabular layout. Thus, the BarChart visualization for presenting the relevance value for all query terms was included in single columns in the table, as was the segment view in its stacked column version. No context switch was necessary; additional information was still visible. ScatterPlot and BrowserView coexisted in parallel. Because it was impossible to provide all the information given by the different INSYDER views in a simple table in a meaningful way, the granularity concept was introduced to let the user find all information, now displayed in diverse levels of detail, but without the necessity of leaving the context of the familiar table. This was the first main advantage of the SuperTable over the INSYDER ResultTable. Changing from one detail level to another caused two effects: The table rows’ height was magnified and in certain circumstances the number of columns altered, depending on the number and kind of meta-data to be presented. This fact of variable column numbers made inevitable the need to always move the complete set of data from level to level.

The GranularityTable describes the next step that lead to an improvement. So far, three visualizations were necessary in the LevelTable version of VisMeB: the ScatterPlot, the BrowserView, and the LevelTable itself. A further integration of the Browser in the GranularityTable reduced this amount to two (main) visualizations - the ScatterPlot and the GranularityTable (additional visualizations were implemented as alternatives, such as a 3D ScatterPlot or a Document Universe, but this had no effect on the original idea). However, there was another drawback still to be remedied. Up to this SuperTable version, it had only been possible to get more information for the complete data set (in the LevelTable) or for a single data set (in the GranularityTable). The opportunity of seeing more details of an individual meta-data was always connected with the intrusion of possibly less relevant information in all other meta-data.

This drawback lead us to the MediaGrid. Here, the selection of a single cell made it feasible to get details of just the desired item. Although an automatic magnification of cells in an orthogonal table without distortion leads to a magnification of the complete
the main focus stays on the cell and thus makes it wider. All other columns shrink as long as the focus is not moved.

The history of the MediaGrid shows the long period of development and the necessity to take one step at a time. Although the idea was conceived at the very beginning, it took more than one single stage to reach the current version of the SuperTable. A complete overview of the different development steps, and the improvements made at each stage, can be seen in Figure 3.19 and Table 3.4.

Up to now only a set of visualizations was presented. But a very important question has so far been neglected: *what is the advantage of this approach? How do these visualizations work together to reach better results?*

The coordination of multiple views is a principal focus of this thesis. This point will therefore be examined in detail in Chapter 5. The use of MCVs in combination with the granularity concept leads to further complications, but also have advantages that must be considered. Coordination becomes more complicated, though the variety of interaction possibilities increases. However, before concentrating on this aspect the granularity concept itself has first to be introduced in the following chapter.
### Table 3.4: Advantages and disadvantages within the MediaGrid history

<table>
<thead>
<tr>
<th>IDEA</th>
<th>ADVANTAGES</th>
<th>DISADVANTAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSYDER</td>
<td>Various visualizations, all used alternatively</td>
<td>Task dependent problem solution by single visualizations</td>
</tr>
<tr>
<td>LevelTable</td>
<td>Integration of formerly separated visualizations in the table, introduction of the granularity concept</td>
<td>No context switch necessary, information split into different levels of detail</td>
</tr>
<tr>
<td>GranularityTable</td>
<td>Integration of all visualizations in the table except the Scatter-Plot (and its variants), smoother change over by added levels</td>
<td>Change of details for single data sets</td>
</tr>
<tr>
<td>MediaGrid</td>
<td>Avoid visible level fragmentation by omitting level buttons</td>
<td>Direct manipulation without level buttons; details for single cells available</td>
</tr>
</tbody>
</table>

3.6 Summary

This chapter gives a general introduction to Multiple Coordinated Views (MCVs) and the visualizations used in the VisMeB approach implementing the MCV concept. After a definition and short overview, a guide for selection and use of multiple views is provided. It encapsulates the question of whether it is meaningful to use MCVs, as well as decision criteria about which visualizations and what kind of interaction between them to choose. Different models for formulating a formal and consistent approach to coordination are delineated. The second part concentrates on the visualizations implemented within the VisMeB framework. The single views are characterized without a detailed insight into the field of interaction and coordination. This will be the topic of Chapter 5, "Interaction Between Granularity-Based Multiple Coordinated Views".
One important aim in information-seeking is to set the focus to something interesting. If you have found a special item, you want to know more about it and to find out if it really meets your needs. This can be done in different ways like opening a further window to present details, linking to another webpage, etc. A base function implemented within the granularity concept is the technique of zooming. The concept’s additional features and significance is described in Section 4.3. To get an overview of the possibilities provided by zooming, a short introduction to diverse zooming techniques and behavior will be given in the following section.

### 4.1 Zoom Introduction

Zooming is a widespread technique for giving the user a more detailed insight into information. Therefore, diverse techniques exist for presenting the particular area of interest. The context of the respective zoom area plays an important role and thus has to be made visible in specific situations. Again, this can be done in various ways. Plaisant et al. [PCS95] and Rueger [Rue98] provide a classification of zooming view opportunities that serves as a basis for the following overview.

- **Detail-only view**: This method is most common in systems like Microsoft Windows or similar user interfaces. A fixed section of a larger image is displayed in a single window. Panning is used to navigate through the information space, often implemented with the help of scroll bars. The global view is not visible, mainly because of the small zoom factor. Imagine a zoom factor of two, then a quarter of the whole image is visible at one glance. Orientation by navigation is quite easy, the problem of being ”lost in information space” is very small. However, if the zoom factor increases, orientation becomes more and more difficult. In this case another technique should be used to support the user. Figure 4.1 shows a cut-out of a Microsoft Windows Explorer window.
• **Zoom and Replace:** First, a global view of the required image is presented. A rectangular area is defined by the user and is displayed in magnified form and replaces the original view. Differences can appear in the size of the cut-out. One possibility is to appoint a fixed size, i.e., the rectangle keeps its size all the time. Another method is to let the user choose the size by dragging a rectangular area that fits the user’s needs. A side-effect of the second version is the necessary adaption of the zoom area to the screen space. If width and height of the rectangle are freely selectable, they have to be scaled where appropriate to fit into the available space. Although more flexibility is provided, this might cause confusion for the user because the selected area and the resulting image do not accord exactly. Figure 4.2 provides an overview of three different variations of zoom and replace.

• **Pan and Zoom:** Provides the same features as zoom and replace. The user can select a rectangular area to be magnified which replaces the current view. As an extension, the user can relocate ("pan") the magnified area. This feature allows a fast scan of the complete area, restricted only by the zoom factor. If the zoom factor is too large, a scan will take much longer and the danger of loosing the context is much higher.

• **Overview and Detail:** This technique can be seen as a logical extension of ”simple” pan and zoom. To avoid the risk of not knowing where the current cut-out (detail) is located, an additional view is introduced that always displays the global view (overview). In general, a small part of the screen is reserved for the global view. Nevertheless, a separate window is another often-used possibility. Figure 4.3 illustrates the use of two separated windows, displaying the SVG browser Squiggle.
Figure 4.2: Variations of zoom and replace: a) zoom only, b) zoom and the possibility of scrolling, c) zoom with additional levels of magnification (adapted from [PCS95])

Figure 4.3: Overview and detail implemented in the SVG Browser Squiggle
• **Tiled Multilevel Views**: In this zoom variant three different views are used to implement diverse magnification steps. The global and the detailed view (see “overview and detail”) are extended by a third one, the “intermediate” view. This allows a multidimensional zoom using different zoom factors for different views. A good example is an application showing a map in distinct detail levels. The global view displays e.g. the complete map of a national park in the US (see Figure 4.4). Depending on the size of the park, you may be interested in a specific part that would allow a round trip within a specific, self-chosen time range. Therefore, the intermediate view shows a map of the eastern part of the park including the point where the trip would normally start - the visitor center. A large variety of intersection signposts in the south makes this particular area especially interesting and thus an even more detailed view shows this location in depth. Information that was not visible before is now readable, e.g. the Jordan Pond House. Coordination between global and intermediate view, as well as between intermediate and detailed view, can be implemented as described above in “overview and detail” with one small restriction: if the selected area in the overview (rectangle A) is moved, the selected area in the intermediate view (rectangle B) should also be moved to avoid a complete disappearance of rectangle B if rectangle A is moved to a position where B is not visible any more.

• **Free Zoom and Multiple Overlap**: An overview of the entire image is displayed in a main window. This provides the basis for the zoom action. The zoom is called free because users are enabled to define a) **an arbitrary area in the actual view** and b) **borders for a new window**. Accordingly, the marked area is presented in the newly created window, which overlaps the source window. To implement various zoom factors, this approach can be repeated using any available window. No coordination between the windows is determined, so all windows act independently. Although this leads to an advantage in specific situations, the problem of overlapping and thus obscuring windows still exists.

• **Bifocal View**: The bifocal view uses a magnification lens metaphor to emphasize the interest in a small area without losing the context surrounding this area. The idea is to present the items of current interest in readable detail whereas the surrounding items are visible in outline. A typical example is the London Underground Map (Figure 4.5).

• **Fisheye View**: The fisheye view extends the bifocal display by adding a distortion factor to the surrounding area and using a smooth transition between the two extreme zoom factors that are described in the bifocal view. Figure 4.6 clarifies the effect. A more detailed view of this topic is described in Section 4.3.

• **Translucent zooming and panning**: This technique extends the general zooming and panning and adds transparent layers that are displayed simultaneously. Focus and context are shown in combination, but coordination of detail cut-outs has to be done
4.1 Zoom Introduction

Figure 4.4: Overview and detail realized within a tiled multilevel view

by users themselves, not by the system, in contrast to e.g. a system using a fisheye technique. The Macroscope System [Lie97] uses this kind of zoom (see Figure 4.7).

As we have seen, the large variety of zoom techniques and their possible implementations can provide more than a "simple magnification". Different windows, parts of windows, zoom factors and distortion techniques can be used to support the process of zooming. The complete approach is independent of the tasks to be fulfilled. Systems like Jazz [BMG00] and Piccolo [BGM04] (see Chapter 2) provide the opportunity to include a "Zoomable User Interface" (ZUI) to other applications. Apart from the techniques available, the zooming behavior plays a role as well. A short introduction is now presented.
Figure 4.5: London Underground Map using a Bifocal Display

Figure 4.6: Map of Washington, D.C., using a Fisheye View
4.2 Taxonomy of zooming behavior

When working with systems providing a zoom function, users fulfill typical tasks. These can be separated into different classes. Tasks as well as typical users are described to give a better insight into the corresponding situation. Plaisant et al. [PCS95] describe a classification that will be described under the following headings.

- **Image Generation**: when creating an image, users are often interested in small parts of this image, which they want to modify (or establish). Nevertheless, getting an overview to look at the work just done is also an important requirement. Therefore, zoom plays a relevant role. Typical users are experts using a CAD/CAM program. First, a sketchy image is drawn, followed by a refinement of interesting parts that have a high importance. This implies a fast change from detail to global view.

- **Open-ended exploration**: a typical example can be found in the area of tourism. If a user plans to visit an unknown city, he wants to become familiar with the city and its local attractions. The space itself is unknown to the user and the risk of getting lost is quite high. Fast navigation is very important for this kind of task.

- **Diagnostic**: this describes a special case of exploration. Samples have to be compared and patterns are searched for. Panning plays a very important role. Typical users could be pathologists or VLSI circuit specialists. If a coverage is not complete it can result in a wrong diagnosis. This type of task is very time consuming because complete analysis is necessary. Panning speed and completeness of coverage are closely related.

![Figure 4.7: Visualization of a file system, using translucent layers in the Macroscope system](image)

**Figure 4.7**: Visualization of a file system, using translucent layers in the Macroscope system
• **Navigation**: in contrast to the open-ended exploration, here users are more or less familiar with the environment. What they want to know is how to get from one place to another. Typical users are truck drivers that have to make a delivery. The current location has to be known as well as the destination. Magnification is only used on a minimum level to display the information necessary for the route.

• **Monitoring**: a typical example where monitoring is used is the observation of a large network. The exact application domain (e.g., security monitoring of a set of buildings, production plants, etc.) is not relevant in this case. The important fact is that the user always has to maintain an overview. If a problem occurs, he has to be able to direct his attention to it immediately, but without losing the context. In this case, overlapping windows can obscure relevant information. This situation has to be considered as a special topic in any case.

Although a lot of zooming techniques and behaviors have already been mentioned, a specific kind of zoom has still to be described - the **semantic zoom**. This specific feature provides the basis for the concept of granularity. It stands out from the simple magnification of specific parts of an item. Additional information is displayed in combination with a variety of zooming techniques. The following section will give a detailed description.

### 4.3 The Semantic Zoom

The granularity concept is based on an idea that is well known in practice. Various names like “semantic zoom”, “semantic scaling”, “drill-down”, or “focus of interest” describe the same approach, dependent on the application domain: "Divide the mass of information into different levels of detail", i.e., always allocate as much information as is needed, desired, or possible. In contrast to common zooming techniques that simply enlarge the object, the semantic zoom provides additional information that otherwise would not be available. As an example please refer back to the Scenario 2 (see Section 3.3) based on the MedioVis approach implemented. A student is interested in a Charlie Chaplin film. The first information he gets is an overview of all movies Chaplin is involved in, as actor or as director. He sees "The Great Dictator”, which he is very interested in, and zooms in with the help of the semantic zoom. Thus, he will get more and more information, e.g., within the first step, a short description; on the next level, all the actors, the year of origin, and a poster; and on the highest level even a short trailer.

This procedure demonstrates the high capability of semantic zooming. The user is able to accommodate a large amount of information broken down into manageable chunks. The higher the interest, the more information is given. Figure 4.8 illustrates the different levels proposed in this scenario.

A typical example of a system implementing semantic zoom is Pad++ by [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 the user can get an overview to comprehend
4.4 Degree of Interest

[Fur81] laid the foundation stone for the idea of semantic zooming by introducing the Fisheye View. He addressed 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 the human visual-processing capacity. The problem arises of deciding what portion of the information to show. Therefore the "Degree of Interest (DOI)" function was established to support the decision process. Three properties have to be defined to calculate the de-
Figure 4.9: Schematic presentation of semantic zooming. The bottom slices ((1)-(3) and (a)-(d)) show views at different points [BHP+96]

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 (be it 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 the following equation:

$$DOI(x_{FP}) := LOD(x) - D(FP, x)$$  \hspace{1cm} (4.1)

The absolute value of the DOI function is of minor interest only. Nevertheless, it is a measure for comparing the importance of different objects in order to decide what should be displayed 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 has been discussed 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, whereas the implementation in practice is achieved as a combination of more than one presentation style:

- **Distorted presentation**: leads to an adaption of size, position, or shape of objects as determined by the DOI. The farther an object is distant from the focal point, the smaller it is presented.

- **Filtered presentation**: performs a comparison of the DOI with a threshold value. The result decides if an object is presented or not.

- **Decorated presentation**: leads to an adaption - as determined by the DOI - in respect of specific presentation variables, like color, transparency, font, animation, etc. If elements are in focus, they will be highlighted by the described variables.

[Pre99] introduces a zoom technique called "Zoom Navigation". In addition to the DOI he defines an AOI, an "Aspect of Interest". The idea of the AOI is to analyze user interactions and to draw conclusions for the desired information. Applying these two approaches you can define a so-called "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 by the equation

$$ AOI(\text{aspect}_k) = f(N, t_1, t_2) $$  \hspace{1cm} (4.2)

where

- $N = \text{Number of visits for aspect}_k$,
- $t_1 = \text{duration of visits}$,
- $t_2 = \text{last visit}$.

As an example, you may recall the previously introduced Scenario 2 (see 3.3): A student is interested in a Charlie Chaplin film. The first information he gets 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 Table 4.1:
**Table 4.1: Representation Matrix for Library Scenario**

<table>
<thead>
<tr>
<th>DOI</th>
<th>ASPECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td></td>
</tr>
<tr>
<td>Abstract</td>
<td></td>
</tr>
<tr>
<td>Year of origin</td>
<td>Name of actors</td>
</tr>
<tr>
<td>Poster</td>
<td></td>
</tr>
<tr>
<td>Trailer</td>
<td></td>
</tr>
</tbody>
</table>

### 4.5 Granularity Zoom Variants

In the VisMeB framework the realization of decreased levels of detail is implemented via a table visualization, the *SuperTable*. The analysis of user actions with the help of the *DROID* (Dynamic Remote Operation Incident Detection) system has recently started (see Chapter 8). The DROID system is strongly interrelated with the VisMeB framework and implemented as part of it. It logs actions like mouse-clicks and filter activation, search terms, time spent at a single view and so on, to allow inferences to be drawn about the user interface. This will lead to improvements of the system itself as well as the concept or, more precisely, the levels of detail, i.e. which information is useful in which level and which column.

Due to the table used, distortion and various aspects of interest are combined in a simple but efficient way: the DOI is assigned to the different levels, the AOI to the columns. Because of the very small amount of DROID data up to now, it is not yet possible to draw conclusions. For that reason the AOI function in the current VisMeB version is a fixed value.

The MediaGrid e.g. uses a distorted presentation to display the focal point, in this case a single cell of interest. A similar approach is chosen in the Infozoom system [SBB96], or the TableLens [RC94] for example. To retain the undistorted layout of a common table, the distortion is made in an orthogonal, not radial, manner. Different AOIs can easily be displayed in columns simultaneously. The main task remains to decide which data is to be shown at what level and in what order, which will be computed later on by the AOI function based on the DROID data. So far it is possible to differentiate three different occurrences of granularity. Step by step, you can increase the amount of information using either

1. The *TableZoom*: The whole table moves to another level (implemented within the LevelTable), or

2. The *RowZoom*: Single rows can change their level independently (implemented within the GranularityTable), or

3. The *CellZoom*: Single cells can be viewed in more detail (implemented within the MediaGrid).
4.5 Granularity Zoom Variants

This leads to decreased implementations, which will be explained in more detail in Chapter 5. The common characteristic is the implementation as an adapted table. Display as well as interaction features vary from zoom to zoom. Nevertheless, the main idea stays the same: *Use a plain table in combination with diverse visualizations and various levels of detail to present the information the user is interested in.*

The ability to build chunks of information by dividing the bulk of metadata into different levels decreases the cognitive attention required from the user. It provides the possibility of viewing as much information as needed at specific time stamps. As a result, the user is not overwhelmed by a flood of information that he has to scan for important facts. The whole extent of the advantages of this concept can be seen in their combination with the MCVs, the combination being described in Chapter 5.

### 4.5.1 TableZoom

The TableZoom was the first idea in the implementation of the granularity concept. The interaction features are therefore very restricted. The aim of this development was to test the general acceptance of the idea. Different levels of detail offer different degrees of information. The more a user is interested in a data item, the more details he will get. To implement this zoom version, the *LevelTable* (see Sections 3.5.1.2 and 5.3) was created. The degree of information is controlled by four buttons, labeled "Level 1" to "Level 4". As a result of clicking one button, the table moves to the specified level. An important fact is that one zooms in to the complete set of data items. It is impossible to zoom in to a single data item. One reason is the changing number of columns that is used within the LevelTable. If different rows are moved to different levels, the number of columns can vary and thus, the headline is no longer valid for the whole column, but only for a subset. This technique would lead to an inconsistency that should be avoided at all costs.

In Figure 4.10 all four levels are presented. Bear in mind that only one level at a time is visible - with the exception of level 1, where a mouse-over effect leads to an enlargement of the row and thus a display of the second level. The underlying data relate to the scenario describing the work with geo-metadata, and in this case they are stored as html-documents.

### 4.5.2 RowZoom

The RowZoom was developed as a derivative of the TableZoom. It is implemented by another variant of the SuperTable - the *GranularityTable*. In cooperation with Dr. Maximilian Eibl [KMRE02], the granularity concept was refined and improved. There were four main differences that influenced the new design:

1. The number of levels,
2. The way of changing from one level to another,
3. The number of columns, and

4. The influence on the table when changing a level.

To achieve a smoother transition from one level to another, the number of levels was increased from four to six. The visualizations used change from degree to degree. The additional steps were introduced to make this change more visible to the user. Granularity sliders instead of buttons control the movement inside the table. Figure 4.11 gives an overview of all six levels. Again, the underlying data come from the domain of geo-information systems.

In contrast to the TableZoom, each row has its own granularity slider. As an add-on, a global slider is used to replace the functionality of the formerly-applied buttons. In this case, the transition from level to level is possible for the table as a whole (analogous to the LevelTable version), or for single rows. This enables a stage combining multiple rows at different levels, as can be seen in Figure 4.12 using the same data as in Figure 4.11.

Various benefits are reaped from this implementation. One major advantage is the more focus-oriented view. If a user is interested in a specific item, he will not receive
Figure 4.11: GranularityTable displaying all possible levels. Again, a mouse-over effect causes an enlargement of a row to display more information.

more additional, but unnecessary information on unfocussed data sets. His main concern is the selected item. Nevertheless, the context, i.e. surrounding rows, is still visible.

4.5.3 CellZoom

The CellZoom is the most advanced approach implementing the granularity concept. The starting point was a zoom variant that moves the whole visualization, in this case the whole table, from one granularity level to another. The next step was to provide the possibility of focussing on single rows, i.e. single data sets. This emphasizes the focus of interest on a specific, although multidimensional, item (if we consider the single characteristics as discrete dimensions). In contrast, the CellZoom features an even more precise
Figure 4.12: The GranularityTable is able to display single rows in different levels of detail. This enables the user to compare several data sets using the same level without moving the entire data set to this level and wasting space (data sets in level 1 need less space than in level 4, for instance).

focus on individual properties or, in other words, a unique meta-data. Thus, the user is able to focus on the content of a single cell and can magnify it to get more information. Therefore, the cell is enlarged in the x- and y-directions. Figure 4.13 shows an example of an early MediaGrid version with four different levels that range from a simple title display through more detailed information like a poster to a short trailer. This example is based on the movie scenario that takes place within the Mediothek of the University of Konstanz.

This example clarifies the opportunities of the CellZoom. We are no longer restricted to specified levels, but every visualization to be shown can now be displayed in an arbitrary cell. Width and height are directly defined by the size of the visualization or length of the text. In the preceding versions of the SuperTable, the height of the row was defined by the visualization or the text with the largest size. This resulted in an enlargement for all cells in this row (or even the complete table) although there may have been only one item that needed the space. For the CellZoom, no maximum height of data contained in the row has to be defined. Depending on the cell’s content, the height is destined. Thus, we are able to include any visualization or text at any level of detail in any cell. A replacement of single metadata or just a change in the ordering does not have any consequences for the residual items in the same row. Figure 4.14 shows a schematic presentation. All possible cell entries are collected in the Visualization & Meta-data Pool. Any of these objects can be assigned to any level (in this case level 1 to level 4) and any column (column 1 to column 4).

Essentially the surrounding cells have to be moved (in the y-direction) or shrunk (in the x-direction) after any CellZoom, even if the cells belong to the same row. The focussed cell always takes the center stage and thus the remaining cells have to be adapted with regard to their size. Diverse techniques to solve the space problem are known and should be mentioned. The problem of space allocation is restricted to a distribution in the x-
4.5 Granularity Zoom Variants

Figure 4.13: The four levels of detail realized within an early MediaGrid version: (1) Only titles are visible, (2) additional information such as tagline or rating are displayed, (3) the movie poster appears, (4) the trailer can be played
direction, but the technique could, by analogy, certainly be extended to a distribution in the y-direction.

- **Resize Off**: keep the size of all non-changed columns. As a result, the user has to scroll horizontally to view the complete content. No other option changes the size of the table.

- **Resize Next**: the column to the left or to the right is scaled down to allow the focused column to grow.

- **Resize Subsequent**: all columns to the right of the enlarged one are shrunk. In

![Visualization & Meta-data Pool](image1)

**Figure 4.14**: Any object in the Visualization & Meta-data Pool can be assigned to an arbitrary cell in the table. The sizes of the object itself and of the surrounding objects in the same row are not restricted.
The granularity concept

general, the columns’ size defines the shrink factor, i.e. larger columns are reduced more than smaller ones.

- **Resize Last**: the width of the rightmost column is reduced as much as necessary.

- **Resize All**: all columns except the focussed one are shrunk. Again the factor is computed in dependency on the actual sizes.

In the current context the **Resize All** version is used. The focussed cell gets as much space as needed, divided in the x- and y-directions. When e.g. the specific cell is magnified by a specific factor in the x-direction, all other columns are shrunk by a factor dependent on available space and current size in order to maintain the actual width of the table. An example will clarify this computation:

Let us assume there are four columns in the table with widths of one, two, three, and four units, respectively. If the second column is enlarged by a factor of 1.5 its current size is three units. Thus, we have one unit extra. This space has to be subtracted from the other columns in proportion to their current size. All three columns together had a size of eight units, i.e. the first column had a size of 1/8 of this space, the third one 3/8 and the last one 4/8. The reduction rate is now computed by the current width minus this proportional factor multiplied by the space to be reduced. This results in e.g. a width of $1 - (1/8 \times 1) = 7/8$ units for the first column, $3 - (3/8 \times 1) = 21/8$ units for the third column, and $4 - (4/8 \times 1) = 28/8$ units for the fourth column. Thus, we can describe the process as follows:

Let $m$ be the width of the table, $x_i$ the width of the columns for $i = 1, \ldots, n$, where $n$ counts the number of columns, $j$ the number of the cell with width $x_j$ to be enlarged, and $x_{j,\text{new}}$ the new width of cell $j$. So we can compute the size of the remaining cells by the equation

$$x_{i,\text{new}} = x_i - \left(\frac{x_i}{m - x_j}\right) \times (x_{j,\text{new}} - x_j), \forall i, i = 1, \ldots, n, i \neq j$$  \hspace{1cm} (4.3)

### 4.6 Summary

In this chapter, an overview of zooming techniques and behavior that are very widespread nowadays was presented. A special form of zoom is described in detail - the semantic zoom. Its specific features lead to the granularity concept that is implemented in the VisMeB framework. The base visualization is the SuperTable, an adapted table including various additional visualizations. Different forms of use are described, these being the TableZoom to move the whole table from one level of detail to another, the RowZoom to focus on single rows and enabling the user to investigate rows on different levels of detail, and the CellZoom for the detailed analysis of single cells in the table, which can be seen as the most fully developed approach. This concept, in combination with the multiple
coordinated views, provides a good basis for making the work of users much easier when they are engaging with visual information-seeking systems.
5.1 Introduction

So far, Multiple Coordinated Views have been introduced by examples (see Chapter 2) and by a formal description, including a definition and a design process model. The visualizations used in the VisMeB framework were specified and the granularity concept was depicted. Now the point has come to combine all this knowledge and investigate how the coordination of the VisMeB views works in practice.

On the one hand the combination of various views into a single system offers new possibilities. On the other hand it often results in a more complex structure. Each new visualization, as well as the multifarious interaction opportunities, has to be understood by the user. If only one view is given, the effect of e.g. selecting, focussing, or zooming is manageable and can be comprehended in a short space of time. The more views that are introduced, the more difficult it will be to work with them. Depending on the kind of corresponding visualization, the effects can vary although the triggered action still remains the same. However, not only the visual representation itself is responsible for a reaction, but also the information used, i.e. Is the collection of information the same or different? It is possible to show the same data in various views, for example in a scatterplot, a table, a bar chart, and so on. Furthermore, it is feasible to connect views with different, but interrelated data.

In the following sections the classification of actions to coordinate multiple views (already mentioned in Chapter 2) is described in depth. Techniques that are implemented within the VisMeB framework are assigned to the respective class. The techniques selection, navigation, and filter are therefore investigated in detail.
5.2 Taxonomy of Interaction Techniques for Multiple Coordinated Views

Two main classes of actions to coordinate multiple views according to [NS97] can be differentiated:

1. **Selection**: Data items (e.g. characters, words, pixels, regions of a 2D image, etc.) can be selected and highlighted. This action expresses an interest in them and can possibly initiate other forms of manipulation. We have to be careful when using the word "selection". In specific situations a more general action is meant (including e.g. moving a rectangular area over a specific location, or the mouse pointer over a data dot), whereas sometimes the "pure selection", e.g. mark a row in a table, or a data dot is described.

2. **Navigation**: To focus on specific data items or to display other data items, the user can navigate the visualizations (e.g. scroll, pan, zoom, slice, rotate, follow link, open file, etc.).

To coordinate the views, three main relationships can be distinguished:

1. Select ⇔ Select
2. Navigate ⇔ Navigate
3. Select ⇔ Navigate

Figure 5.1 demonstrates this taxonomy of multiple view coordination.

![Figure 5.1: Taxonomy of multiple view coordination [NS97]](image)

The possibility of filtering can be thought of as a specific kind of selection. If we consider filtering as a selection of a subset, the effects on other visualizations can also be described as a selection or navigation. In general, a filter has influence on the complete data corpus. This is correct, as long as "usual" filters like Dynamic Queries are used. However, there are filter techniques that have a different effect on multiple views. This case will therefore be considered in a special section.
5.2 TAXONOMY OF INTERACTION TECHNIQUES FOR MULTIPLE COORDINATED VIEWS

5.2.1 Select ⇔ Select

This kind of coordination is the most widespread one. The user selects (highlights, brushes, focusses) an item in one view and the corresponding one in (usually) all other visualizations is selected (highlighted, brushed, focussed). This method helps the user to correlate equivalent or related items. Typical mechanisms to highlight/brush an item are the retinal properties [Ber83]: color, shape, texture, size, orientation, and position (see Chapter 2). The combination of different properties can intensify the effect and thus give the user a more obvious hint. The most typical and widespread example is a combination of size and color (or texture).

A differentiation will be made between diverse selection techniques. These are: Focus (moving the mouse over an item to express interest), Selection (the ”pure selection”, very often indicated by a mark), and Filter (removing uninteresting items). Each technique results in another effect on the corresponding view(s) and is triggered by another action. An additional dependency is created by the respective view, i.e. is it a completely graphical one like the Scatterplot or the Document Universe, or is it a table-based visualization like the LevelTable, the GranularityTable, or the MediaGrid. These relationships are therefore explained in detail in the particular sections below. Systems using this kind of interaction can be found in Section 2.6.1.

5.2.2 Navigate ⇔ Navigate

To synchronize views by navigation, actions like scrolling, zooming, panning, etc. are used. A good example is the simultaneous scrolling through two windows, where the first one displays an HTML page, the second one the subjacent source code. To find out e.g. how a specific table on the display is built, you can scroll to this section and get the corresponding section in the source view. Various systems that use this kind of synchronization can be found in Section 2.6.2.

5.2.3 Select ⇔ Navigate

In this kind of tight coupling the user selects items in one view to navigate in another one, and vice versa (i.e. navigate to select). A typical example is the use of an overview, like a table of contents, and the content itself. In contrast to distortion-based techniques, where details are shown within the context, two different windows are used, i.e. overview and detail are split.

A combination of these possible relationships is implemented in the VisMeB system (see also [MLRK03]). The use is dependent on two factors Which views are used? and Which level of detail is used?, which highlight the strong connection between the multiple views and the granularity concept. Implementing only one concept at a time would decrease the possibilities offered by the two together. Simply combining the two provides advantages that do not otherwise exist.
As a visualization that implements the granularity concept, the SuperTable is used. It is implemented in three previously described versions, unifying different characteristics (see Section 3.5). A short overview of the system’s architecture is given in Figure 3.5. To explain the different versions implemented, the scenarios introduced earlier are used to clarify their use.

### 5.3 TableZoom

The scenario introduced in Chapter 1, which alludes to the INVISIP project and describes the typical task for a site planner, serves as a starting point. To realize the TableZoom the LevelTable was implemented. Therefore, mainly simple interaction techniques are used. To change the level of detail, a button with the corresponding label is clicked and the whole table moves to the desired degree of granularity. Interaction possibilities between the SuperTable and the Scatterplot/DocumentUniverse are restricted to focus (move the mouse over an item without clicking a mouse button), selection (mark an item by clicking the left mouse button) and filtering (remove items that do not fulfill the filter characteristics), i.e. realizing the “Select ⇔ Select” relationship. These selection operations can be fulfilled independently from the granularity level. In level 1 an additional highlighting method is used to emphasize a focus; all remaining levels act in the same way, as can be seen in Section 5.3.1. The filter works in a self-contained way without a direct reference to a specific level. It can be invoked by global working filters - the dialog box or the CircleSegmentView, or the local working one - the Movable Filter. A detailed description is given in the corresponding Section 5.6 below.

The purely graphical views like ScatterPlot (2D and 3D) as well as the DocumentUniverse react very similarly to actions initiated by the table visualization. Therefore, these graphical displays are subsumed under the term “Graphical Only Views” or short “GOViews”.

#### 5.3.1 Coordination with GOViews

Level 1 displays numerical values as bars to shrink the row’s height to a minimum. Possible values are the relevance of the whole data set as well as the query terms’ relevance (see Figure 5.2), or the size of the data set. Text is not visible. In this level, the focus of a data set is achieved by moving the mouse over it - regardless of whether it is in the LevelTable (focus of a table row) or the GOViews (focus of a painted item). It results in a highlighting of the corresponding data set in the GOViews via an enlargement of the glyph and a changing fill color, and a movement to the second level i.e. a row magnification in the LevelTable. To mark a data item, the user has to click a row in the table or a glyph in the GOViews. This action results in a changing color for all visualizations, the row background color in the table, and the fill color for the glyph.

Level 2 enlarges the row and adds text to the so far unreadable columns or values to the corresponding bars. As was seen in the user tests, this level is preferred as the starting
level by a large percentage of users. The additional overview function of the ScatterPlot made level 1 unnecessary in this cooperation - at least, if the users work with the GOViews like the ScatterPlot. If not, level 1 can present a large set of data providing a short insight into relevant data. The effect of a focus or a marking action is still the same as in level 1 except for the magnification of the row. Level 3 and level 4 work analogously to level 2 with respect to the actions "focus" and "selection".

The coordination between the Document Universe and the LevelTable is assembled in a very similar way. The concrete effects are described in Section 5.4.1 below.

### 5.3.2 Coordination with Textual Views

Focus and selection in coordination with the BrowserView works in a very similar way to the GOViews. However, the principle of overview and detail behind this connection is exactly the reverse. If a data set is focussed in the table and the BrowserView is activated, the current data set is displayed as a whole in the BrowserView. In this case, the table serves as overview whereas the BrowserView displays the detailed information. Because of a more consistent general view the backgrounds of table and BrowserView use the same colors. As a result, this color gets more saturation when focussing and more still when selecting.
Furthermore, a new interaction between LevelTable and BrowserView based on "Navigate ⇔ Navigate" is introduced by the third level. For that reason a new visualization appears - the Relevance Curve. It divides the complete data set (e.g. HTML- or XML-document) into segments, depending on the overall length. The maximum number of segments is restricted, but can be increased or decreased if necessary. Consequently, a segment’s length is determined by this maximum number and the text size. The importance of a segment is visualized by a curve whose spikes indicate the importance with reference to the query terms. As a result, it is very easy to find important text passages without scanning the whole text. Moving the mouse over segments highlights the corresponding segments in the BrowserView (see Figure 5.3).

Figure 5.3: LevelTable Level 3 with BrowserView. One segment with a maximum spike is highlighted.

Level 4 offers the highest granularity degree. It provides the same interaction possibility as level 3, but with one small difference. The relevance is split into stacked columns that display the single query terms’ relevance accentuated by color. Each query term is assigned to a specific color that will stay unchanged for all levels and views. This implies keyword highlighting as the next consequence in the textual view, using the assigned colors (see Figure 5.4).

A further interaction feature between LevelTable and BrowserView is implemented by a popup-menu. The user is able to add a data set to the BrowserView permanently. Thus, the focus effect disappears in this case. This can be assigned to the "Select ⇔ Navigate" relation. The fact is marked by a small "x" in the upper right corner to give a hint of this permanent assignment. As an extension of this idea, it is possible to add more than one data set to the BrowserView. This enables the user to compare items without changing
the order in the table. Different alignments are available, from a completely horizontal one via a rectangular to a completely vertical arrangement. Depending on the type and structure of data respectively, one of the alignments is relevant and applicable (see Figure A.4 in Section A.2.3).

To get an overview about the various selection-technique relationships between the LevelTable and other views, the possible actions and reactions are displayed in table 5.1.

<table>
<thead>
<tr>
<th>Effect/Action</th>
<th>Focus</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoked by</td>
<td>Mouse-over</td>
<td>Left mouse-click</td>
</tr>
<tr>
<td>LevelTable</td>
<td>Change background color, enlarge row (if level 1)</td>
<td>Change background color</td>
</tr>
<tr>
<td>ScatterPlot</td>
<td>Enlarge dot, change fill color</td>
<td>Change background color</td>
</tr>
<tr>
<td>3D-ScatterPlot</td>
<td>Enlarge cube, change fill color</td>
<td>Change background color</td>
</tr>
<tr>
<td>Document Universe</td>
<td>Enlarge square, change fill color</td>
<td>Change background color</td>
</tr>
<tr>
<td>BrowserView</td>
<td>Display data set</td>
<td>Change background color</td>
</tr>
</tbody>
</table>

Table 5.1: Relationship between corresponding views realizing the select - select coordination
5.4 RowZoom

To create the RowZoom variant, the GranularityTable was implemented (see Figure 5.6). Its features vary, but the same scenario as with the TableZoom can be used. This version differs in several ways. First of all, the number of columns is fixed. Whereas the LevelTable is able to deal with a different number of columns depending on the level of detail, the GranularityTable provides four of these. One is for selection ("data set is marked or not"), one for included visualizations (bars, segment views, etc.), one for text, and one for the granularity slider (replacing the level buttons) to assign a single row to a specific level. This shows another aspect of granularity. While changing the level of detail for a single line, the remaining lines stay in their current stage. Additionally, a global granularity slider can adjust the whole table to a certain level. Within this implementation, the change from one grade to another is made more smoothly, which results in six levels instead of only four. Levels 1 to 4 present bars, values and text, similar to the first two levels in the LevelTable. In Level 5 the stacked columns are replaced by tile bars that divide the data sets into segments, though using a vertical, not horizontal alignment. The BrowserView is no longer necessary because of a wide text column that takes over its function.

5.4.1 Coordination with GOViews

Focussed segments in the visualization column are highlighted in the text column. Level 6 uses even more space by unifying visualization and text column in a single one to display the maximum degree of information, i.e. the complete data set. In this combination with the Scatterplot, another adjustment of the granularity is possible. Opening the context menu for a data point in the Scatterplot provides a slider to directly assign a specific level in the table. There is no necessity for context switching (select a point in the Scatterplot, move the mouse to the corresponding table row, change level for this row), and interesting points can be manipulated immediately in a common but efficient way, demonstrating another advantage of the MCV and granularity concept combination (see Figure 5.5).

The Scatterplot provides other features that will be mentioned shortly. One function is the zoom - independent from the semantic zoom. The user is able to define a rectangular area and have a closer look at it. This is especially helpful if one is interested in a small part that is densely populated. A Movable Filter, as an additional tool working as a temporary sieve, will be presented in Section 5.6.3.

The Document Universe (see Figure 5.7), as an alternative to the Scatterplot, behaves in a very similar way to its role model. Focus and selection in the table, or Universe, highlights the corresponding data point in the coupled view. The use of 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. The zoom function enables the user to get more details for a specific area. Additionally, panning is possible and helps to move around the complete data space.
Figure 5.5: Interaction via a pop-up menu in the ScatterPlot. The item of interest can be found in the ScatterPlot and explored for further details by using the slider to change the level

5.4.2 Coordination with Textual Views

Because of the large text column provided in this SuperTable version, the BrowserView almost loses its function as an overview of the whole data set. However, the advantage of being able to compare several data sets using different alignments is still there. Although the GranularityTable enables the user to shrink uninteresting rows to a minimum (level 1) and magnify interesting rows to a maximum (level 6), it is still hard to compare them. Either there are too many shrunken rows between the interesting ones or they are ordered by interest, but only a vertical comparison is possible, independently of the kind and amount of data.
### Table 5.2: Relationship between corresponding views realizing the select - select coordination

<table>
<thead>
<tr>
<th>EFFECT/ACTION</th>
<th>FOCUS</th>
<th>SELECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVOKED BY</td>
<td>Mouse-over</td>
<td>Left mouse-click</td>
</tr>
<tr>
<td>GRANULARITYTable</td>
<td>Change background color, enlarge row (if level 1)</td>
<td>Change background color</td>
</tr>
<tr>
<td>SCATTERPlot</td>
<td>Enlarge dot, change fill color</td>
<td>Change background color</td>
</tr>
<tr>
<td>3D-SCATTERPlot</td>
<td>Enlarge cube, change fill color</td>
<td>Change background color</td>
</tr>
<tr>
<td>DOCUMENT Universe</td>
<td>Enlarge square, change fill color</td>
<td>Change background color</td>
</tr>
<tr>
<td>BROWSER View</td>
<td>Display data set</td>
<td>Change background color</td>
</tr>
</tbody>
</table>

**Figure 5.6: GranularityTable levels 1 to 6**
Figure 5.7: Document Universe displaying semantic similarity. The selected points are colored blue whereas the focussed one uses a red highlighting.
5.5 CellZoom

This last version of the granularity concept is implemented within the MediaGrid as part of the MedioVis system, one of the previously mentioned variants within the VisMeB framework, which uses the application domain of the University library. Because of a different scenario, it differs slightly from the former ones. The scenario has already been introduced in Chapter 4; the task was to find an appropriate movie in the library database. In this implementation of the granularity concept, the focus is on single cells, no longer on rows or the whole table. If you are interested in a specific fact described in a single cell, you are able to get more information by zooming in on this particular cell. The technique used is described above in Section 4.5.3, and corresponds to the "Fisheye View". An orthogonal distortion is used to enlarge the focal point ("the cell of interest") and shrink as many surrounding items as necessary. Because of the orthogonal distortion it is possible that neighboring cells become enlarged, too. Figure 5.8 demonstrates this effect.

The graphical displays provided within this variant are a ScatterPlot and a LocationMap. A slight difference from the previously introduced ScatterPlot can be found in this version with respect to its appearance and interaction possibilities and will be described below. The LocationMap is a completely new visualization introduced for this specific scenario. Thus, the GOViews in this context consist of these two visualizations.

![Figure 5.8: MediaGrid and CellZoom](image)

5.5.1 Coordination with GOViews

The synchronization between the MediaGrid and the ScatterPlot resembles the coordination described for the Level- and the GranularityTable. In the table as well as in the plot, Focus and selection is indicated by colored highlighting. Differences can be found in the kind of visual point presentation, where glyphs take the place of circles and where the handling of zooming is changed. The additional encoding by shape and color enables the user to recognize the kind of meta-data, i.e. DVDs can be easily distinguished from VHS video tapes as can be seen in Figure 5.9.
5.5 **CellZoom**

Figure 5.9: The MediaGrid of MedioVis in combination with the ScatterPlot. Original locations are changed; the ScatterPlot is normally positioned in the upper part of the window. A flexible assignment technique allows the movement of single visualization windows.

Zooming in the ScatterPlot can be done by simply clicking the mouse button (**left** to zoom in, **right** to zoom out) or turning the mouse wheel (**forward** to zoom in, **backward** to zoom out). By enlarging the cells, more information is visible and this is not restricted just to the focal point, but also applies to the surrounding cells. In this case, the interaction to focus on a specific point of interest leads to a higher DOI for the complete row.

The LocationMap is a scenario based view that is introduced for this specific application domain. In the scenario (defined by $5T_3$) the focus lies on the "Mediothek", the “multimedia” part of the University library in Konstanz containing DVDs, CD-Roms, VHS, and further multimedia items. These media are all located very close together in a manageable room which makes it possible to draw a map of all the shelves, a map that can fit onto a small screen and still be readable. The user can now move the mouse over a specific table row and the corresponding medium will be highlighted in its exact position in the shelves of the Mediothek. This eases the search process when an interesting medium is found on the screen, but its physical location in the library has not yet been found. Another difference exists in this scenario - the views can be dragged and dropped almost
arbitrarily. In our example, a version of the LevelTable, a Scatterplot and the MediaGrid are positioned on the upper part of the screen (see Figure 5.8), whereas the BrowserView and the LocationMap share the lower part (see Figure 5.10).

```
Figure 5.10: MedioVis with BrowserView and LocationMap
```

### 5.5.2 Coordination with Textual Views

The completely different scenario used in the MedioVis approach leads to a new configuration of visualizations, including the ones integrated in the SuperTable. Therefore, the SegmentView implemented by stacked columns (LevelTable) or TileBars (GranularityTable) was left out. The BrowserView gave way to the TitleView which indeed displays a more detailed insight into the corresponding data set, but not a complete overview; rather, an accurate selection of meta-data. Selection and focus still work in the same way, made clearer by changing background color, as can be seen in Figure 5.11.

A further textual view is introduced, a list of selected data sets. It enables the user to collect the interesting data in a separate window in order to send it by email, to save the list, or to print it out. This adaptation was made because of the application scenario and the practical benefit for users. The reason was the fact that a typical action for library users is to search for specific media and then to have a closer look at the item itself, i.e. by walking to the corresponding shelves, taking the media and then probably borrowing it. Thus, the user needs the exact location and identification, which one normally gets by making a note of this important data.

A further advantage of the MedioVis approach implementing the MediaGrid is its completely closed concept of an information-seeking system. All steps are taken into account, starting from the query formulation via the result set presentation, the selection
5.5 CellZOOM

of relevant data sets and their subsequent processing. The MedioVis framework enables the user to fulfill all these necessary tasks in a single system without changing the context, as can be seen in Figure 5.11. The buttons to the far right provide a direct processing feature facilitating the sending of the list of selected items as an email, saving it to disk or printing it directly. Thus, all steps can be done within this single system. Figure 5.12 shows the concept. The green arrows indicate the possibility of taking one step, or even more steps, backwards in the process. Some steps are not possible if preceding steps are not executed, such as a selection without a result set, or subsequent processing without a selection. But turning back is possible at any time.

This emphasizes the general applicability of the MedioVis framework. Dependent on the underlying scenario and data, the steps can differ slightly but the concept will still cope with the demands made on it.

Because the development of VisMeB follows the information-seeking mantra "Overview first, zoom and filter, then details on demand" that we have already mentioned, the different filter variants that the VisMeB framework provides have to be introduced. Zoom is available in the Scatterplot as well as in the SuperTable, where it is realized as semantic zoom. Therefore, the focus will now be on the diverse filter features built into the
5.6 Filter

The possibility of filtering the result set is one main advantage of interactive systems. A large amount of data can be reduced to a manageable size by restricting specific characteristics. Using direct manipulation [Shn98] enables the user to see immediately the consequences of the filter action. Visualizations can support this process by providing appropriate interaction techniques to fulfill this task. In the VisMeB implementation different types of filters are implemented. They vary in their appearance as well as in their function, but nevertheless are based on direct manipulation. While one kind of filter follows a global influence and restricts the whole set of data, another kind works only temporarily. These variants and their influence on the set of views will be explored. Table 5.3 at the end of section 5.6 gives a short overview of the different versions and their advantages and disadvantages.

5.6.1 Dialog Box

It is possible to filter the data set by restricting any kind of metadata. Depending on the character (e.g. nominal or ordinal) the interaction widgets used will vary slightly. Categories (e.g. languages) can be filtered in or out by selecting a checkbox 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 enable direct choice of the domains (e.g. ".de", ".us", or ".co.uk") is attributable to the application area of geographical information systems, seen in the scenarios based on $5T_1$ and $5T_2$.

Filters can easily be activated by a checkbox connected to the corresponding metadata. This makes it much easier to use a filter at a later stage, when the filter settings were established earlier on (see Figure 5.13). In addition, it is possible to activate all filters or to invert the current selection to enhance the usability of the selection activity. This kind of filter affects all the visualizations i.e. filtered data are removed from every view in the complete set.
5.6 Filter

5.6.2 Circle Segment View

An optimized filter for categorical data is given by the CircleSegmentView (see Figure 5.14). It is used as a visual filter instead of a textual one like the Filter Dialog Box described above. Two pie charts dominate the view, limited by horizontal and vertical sliders that are able to exert a filter function. The data sets - visualized as points - are positioned by three properties: The *pie segment* (dependent on the category), the *distance from the centre* (dependent on the metadata assigned to the horizontal slider) and the *angle* inside the corresponding segment (dependent on the metadata assigned to the vertical slider). A more detailed description can be found in [Kle05]. The CSV is able to work in two different modes. These are a *selection mode* and a *filter mode*. When used in selection mode a single click on a segment selects all documents that are visible in this set. Another feature appears in this case: all selected data can now also be seen on the second pie chart. This is very helpful if one wants to see the distribution of the selection under another criteria. When used in filter mode, the segments and sliders work as assumed: they filter out everything that is not required, which will then no longer be visible. This has an instant impact on e.g. 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 rapidly explore the multidimensional space of data sets. And along the lines of the preceding filter, selecting or filtering affects all other visualizations.

5.6.3 Movable Filter

The *MovableFilter*, influenced by the moveable filters by [FS95], and available in the ScatterPlot, has an effect on the SuperTable as well. If objects are filtered out by the lens, the background of the corresponding objects in the table changes to the lens color (see Figure 5.15), but no movement takes place. The reason for this implementation is the fact
that a permanent movement in the table, by removing objects, would obviously confuse the user. Moreover, the possibility of exploring the filtered documents would be taken away. Additionally, it is possible to use different lenses simultaneously, which made it necessary to add half-transparent lens colors. The boolean expression ‘AND’ is visualized by the summation of the colors. Thus, documents from Italy (meta-data “country”) are colored red, documents concerning infrastructure (meta-data “theme-code”) blue and documents having both these attributes, purple. This only holds true for items that lie underneath the moveable filters.
Figure 5.15: Two MovableFilters connected by a boolean ‘AND’. Blue-colored rows in the table arise from the meta-data “country” and the characteristic “Italy”, yellow ones from ”theme_code” “Infrastructure”, and grey ones from a combination of both. Only points underneath both filters get this combined color.
Table 5.3: Advantages and disadvantages of filters used

<table>
<thead>
<tr>
<th>ADVANTAGES</th>
<th>DISADVANTAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dialog Box</strong></td>
<td></td>
</tr>
<tr>
<td>• Users are familiar with</td>
<td>• Additional window, i.e. context switching neces-</td>
</tr>
<tr>
<td>• Easy adjustment of combined filters</td>
<td>sary</td>
</tr>
<tr>
<td>• Activating / deactivating of filters by simple check boxes</td>
<td></td>
</tr>
<tr>
<td><strong>CircleSegmentView</strong></td>
<td></td>
</tr>
<tr>
<td>• Optimized for categorical data</td>
<td>• Primarily high cognitive load</td>
</tr>
<tr>
<td>• Pie charts are well-known business graphics</td>
<td></td>
</tr>
<tr>
<td><strong>Movable Filter</strong></td>
<td></td>
</tr>
<tr>
<td>• Included in the context</td>
<td>• Combining multiple filters by a boolean opera-</td>
</tr>
<tr>
<td>• Data are marked, not removed, to reduce confus-</td>
<td></td>
</tr>
<tr>
<td>ing movements in the table</td>
<td>tor can confuse the user</td>
</tr>
<tr>
<td>• Filter applicable to specific areas, not just</td>
<td></td>
</tr>
<tr>
<td>the whole data set</td>
<td></td>
</tr>
</tbody>
</table>
5.7 Benefits and Shortcomings

The combined use of Multiple Coordinated Views and the granularity concept has a marked effect on the power of the system. A SuperTable that implements different levels of detail without an additional visualization like the ScatterPlot is not unsuitable, but is less powerful. Analogously, the same statement is correct for a single ScatterPlot visualization. As we saw above, the development of the current implementation of the MediaGrid had to pass through a number of steps until the current status was reached. The advantage of MCVs lies in two main properties:

- The variety of visualizations available to satisfy users, who need to use the "best method" for achieving a task, whether this is dictated by subjective preference or by the needs of the specific task, and
- the coordination of these different views.

The benefits of the first item were considered in Chapter 3, the second one will be discussed here.

In all system implementations, interaction is necessary and is effected. It does not differ very greatly but, dependent on the kind of implementation of the SuperTable, variations can be found. The reasons for this are based on the way the level-of-detail concept is realized. Apart from these granularity versions, all other interaction possibilities like focus or selection will not vary. Overview and detail play an important role in this context. The possibility of picking an object in the overview (e.g. the ScatterPlot) and getting a direct feedback about details located in another view (e.g. the SuperTable) enables a quick and easy exploration functionality. An example will clarify the advantages.

Let us assume that Daniel Beck (introduced in Chapter 3, Scenario 3) is looking for the name of the king in a very famous fantasy adventure. He has it on the tip of his tongue but he does not remember the exact title. The only facts he knows are: it is a fantasy movie, it is something about a ring, the movie was rated very good, is not very old, and he wants to know the name of the king featured in this movie. Therefore, he enters the query terms "fantasy", "king", and "ring". As a result he gets 76 datasets.

In the ScatterPlot, three dots in particular attract his attention, resulting from their location in the upper right corner. Dependent on the axis assignment (Rating and Year), this implies highly rated and current movies. Thus, he selects all three as can be seen in Figure 5.16. To get more information about the three movies, Daniel changes to level 2 in the LevelTable and sorts the data set by "selection". He immediately recognizes the correct title of the movie he was looking for, "The Lord of the Rings". So the first problem - the missing title - is solved. Nevertheless, he is still looking for the name of the king. There are three parts of the trilogy available, "The Fellowship of the Ring", "The Two Towers", and "The Return of the King", but only two of them include the query term "king" (see Figure 5.17), visible by means of the red bar in the third column.

To have a closer look at these, he changes to level 4, where the stacked-column version of the segment view is included and enables Daniel to find query terms in the description. The detailed description in this case (i.e. using the LevelTable) can be given by the
BrowserView, which is patched in automatically when entering level 4 for the first time. As a first step Daniel investigates part three, "The Return of the King". Unfortunately, the name of the king he is looking for is not mentioned (see Figure 5.18).

Because of this, he analyzes the second part of the trilogy, "The Two Towers". Moving the mouse over the red rectangle in the segment view (the upper part of Figure 5.19) highlights the corresponding segment in the BrowserView (the lower part of Figure 5.19). Fortunately, this time the king’s name is mentioned and Daniel obtains "Theoden" as the answer to his question.

This example clarifies the strong relationship between the different visualizations implemented as views (in this case LevelTable, ScatterPlot, and BrowserView) and the different levels of detail in this SuperTable version that provides textual information itself as well as further visualizations (BarCharts and the SegmentView). Without an interaction of these parts, the solution of the problem scenario described above would not be possible in such an easy, quick, and intuitive way. If the GranularityTable instead of the LevelTable were to be used, the stacked column version of the SegmentView would be replaced by the TileBar version and the BrowserView would appear as the fifth level, integrated in the table. In other respects the procedure would stay the same.
The combination of different filters may appear to be an unnecessary effort. Nevertheless, the small differences in handling and visual effects make the existence of all the filters worthwhile, and it is also the case that different users may prefer a specific choice. If we have a closer look at the individual implementations, we will find situations where one or other of them is a better fit for solving a current problem. All filters have their own strengths and weaknesses, which become visible in the context of the kind of task to be solved and the method of implementation. The Dialog Box uses a familiar and effective layout to provide a filter mechanism for all available meta-data. Well known functions are used and make it easy to adapt the settings to the prevailing task. All metadata can be filtered at the same time by activating the filters through simple check boxes (see Figure 5.13). Adjustments stay unchanged whether the filter is activated or not, which enables the user to keep current settings. The only drawback is the additional window, because the context is switched and the user has to leave the main MCV window.

The functionality employed by the CircleSegmentView is largely unknown for non-expert users. Although pie charts are very widespread in the field of business graphics, the combination of two charts and the possibility of interacting with them by clicking on items, segments, or using Alphasliders is new to a wide range of people. Nevertheless, it provides an optimum approach for filtering categorical data (see [Kle05]), which occur very often in the field of meta-data. Typical examples are e.g. “domain category” or “lan-
interaction between views

Figure 5.18: Level 4 of the LevelTable, including the SegmentView in its stacked-column version for finding query terms in the description that is displayed in the BrowserView in the lower part of the window

guage” in the domain of web documents, “themescape” or “country” in the area of geo meta-data, or “genre” and “media type” in the library scenario. This fact makes it meaningful to introduce the CircleSegmentView as a filter.
The Movable Filter, in contrast to the ones mentioned above is the only filter that can work on a specific part of the data, not just on the whole data set. It is located in the ScatterPlot view i.e. directly included in the context and can be used by moving around the available information space. The size is adaptable and a combination of more than one filter is possible. Filtered items are just marked in the table and not removed, as seen above. Moving the filter around and thus filtering out data sets completely would lead to an extreme and confusing movement in the table. The use of the boolean operators "AND" and "OR" is a benefit on the one hand, because additional operations can now be executed. On the other hand the arbitrary combination of multiple filters and boolean operators can lead to confusion for the user.
We can see that the combination of the different available filters is useful and supports the user, where the choice of the best filter to use always depends on the situation and the task to be solved. The variety available makes it possible to choose the one best fits - for the task or the user, who may prefer one of the technologies provided.
5.8 Summary

This chapter deals exclusively with the synchronization of multiple views. To structure these correlations, two different classes of actions were defined, selection and navigation. From these classes, three possible combinations arise, Select $\Leftrightarrow$ Select, Navigate $\Leftrightarrow$ Navigate, and Select $\Leftrightarrow$ Navigate. According to these combinations, the connections implemented in the VisMeB framework are assigned. A differentiation was made between graphical and textual views to add an even more precise structure. The different filter variants implemented provide the conclusion for the synchronization. A combination of multiple coordinated views and the granularity concept leads to advantages, that would not be possible using just one of the techniques presented. The example introduced clarifies this statement. The simultaneous use of an overview in one visualization and detailed information presented by another view makes it easy to switch between these visualizations without losing information. Without a strict coordination model the user could loose his focal point and overlook important correlations. The combined use therefore allows a simplified and efficient access for finding important information in databases containing hundreds of items. Unfortunately, at the current stage of development these benefits are mainly analytical. There are still tests to be done to prove these statements. An initial number of tests has already been performed and the results are presented in Chapters 6 and 7. They give a first hint that the approach is definitely promising. Chapter 6 gives an
introduction to the "User Centered Design Process" that strongly influenced the development of the system and the individual stages of development. Chapter 7 adds a detailed description of an evaluation performed to compare the SuperTable idea with a typical list-based visualization as used nowadays by search engines like e.g. Google.
The first part of this chapter gives a general introduction to the user centered design process and its background in the field of interaction design. Usability is one catchword that will haunt us during the whole chapter. Because of its good design and structure, this overview will closely follow the book of Preece et al.: Interaction Design - Beyond Human-Computer Interaction [PRS02]. The interested reader is therefore referred to this source for further, more detailed information. The reader who is already confident with this topic can omit this part and turn towards the second part, which describes the different development stages the VisMeB framework went through.

Measurement of a system’s quality, independent of its application domain, its design or features should always be judged by how well users interact with it. This depends heavily on **effectiveness**, **efficiency**, and of course, subjective **user satisfaction**, i.e. the system’s **usability**. Before moving into more detail, the terms should first be defined.

**Definition 6.1 (Effectiveness:)** Effectiveness represents the precision and completeness with which a user can achieve a task. This normally refers to the degree to which errors are avoided and tasks are successful, measured by ”success rate” or ”task completion rate”.

**Definition 6.2 (Efficiency:)** 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 number of keystrokes or the number of interactive steps which a user has required to complete a task.

**Definition 6.3 (Satisfaction:)** Satisfaction is a common reference to the set of subjective responses a person has when using a system. Typically, satisfaction is measured with questions that have their responses on Likert scales, e.g. ”How satisfied are you with this software? (1=very dissatisfied, 7=very satisfied)”.

In the ”ISO 9241-11: Guidance on Usability (1998)” [ISO98] standard the definition of usability is given as follows:
Definition 6.4 (Usability (2)) Usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Shneiderman [Shn92] notes five measurable human factors that describe the degree to which a system meets the users’ needs:

1. Time to learn
2. Speed of performance
3. Rate of errors by users
4. Retention over time
5. Subjective satisfaction

To optimize these factors, the design of the system, as well as the process of designing itself, has to focus on the users and their needs. This is implied by the phrase user-centered design (UCD). UCD is integrated in the field of Interaction Design and considers a wide variety of factors, such as software, hardware, environment, task, type of user, and so on. The adaption of the system to the users’ capabilities and limitations ensures an increasing usability. It makes it easier to use, increases the user satisfaction, and reduces the error rates. In particular, systems that require a high cognitive load e.g. life-critical systems like air traffic control, nuclear reactors, power utilities, or staffed spacecraft benefit from this. Nevertheless, no other field of interactive product use should be neglected. In [GL85] three principles that should lead to a “useful and easy to use computer system” are introduced. These are:

1. Early focus on users and tasks.
2. Empirical measurement.
3. Iterative design.

The first and probably most important principle for UCD, early focus on users and tasks, can be divided into five further principles to clarify the meaning [PRS02].

1. Users’ tasks and goals are the driving force behind the development.
2. Users’ behavior and context of use are studied and the system is designed to support them.
3. Users’ characteristics are recorded and designed for.
4. Throughout development, users are consulted from the earliest to the latest phases and their input is seriously taken into account.

5. All design decisions are taken within the context of the users, their work, and their environment.

First and foremost we think of users working directly with a system. But there are many more people that are affected by, or influence, the development process. To describe the whole group of involved people, they are called stakeholders.

**Definition 6.5 (Stakeholders):** Stakeholders are people or organizations who will be affected by the system and who have a direct or indirect influence on the system requirements.

[SK98]

Thus, we have to differentiate diverse classes of stakeholders [Kir91]:

- **Primary stakeholders:** People who directly use the system
- **Secondary stakeholders:** People who use the system occasionally or not directly
- **Tertiary stakeholders:** People who are directly affected by the success or failure of the system

[DFAB04] adds a further class to this enumeration, the facilitating stakeholders. These are people who are involved in the design, development, and maintainance of the system.

Projects that have benefited from the application of UCD techniques are widespread. One of the first systems that implemented this approach by using the above mentioned three prinicples of [GL85] was the 1984 *Olympic Messaging System* (OMS), a voice mail system, developed to let Olympic Games contestants and their families and friends send and receive messages [GBL+87]. A lot of evaluation activities are included in this early study, e.g. printed scenarios, iterative testing, development of early prototypes, an Olympian joining the design team, interviews, to name but a few.

As a second example we can cite *HutchWorld*, developed in cooperation with Microsoft researchers and the Fred Hutchinson Cancer Research Center as a virtual world to support cancer patients, their families, and friends. It enables cancer patients, their carers, family, and friends to chat with one another, tell their stories, discuss their experiences and coping strategies. Thus, they can gain emotional and practical support from one another [CSF+00]. Again, a lot of different evaluation techniques are used, similar to the ones described above, but partially refined.

To add a third application field, we will turn to the domain of spacecraft, considering spacecraft ground data systems at NASA’s Goddard Space Flight Center [FDM+99].
In this context we will find many other projects that employed UCD, like *EOSDIS - the Earth Observing System Data and Information System*, *SERS - the Spacecraft Emergency Response System*, or the *Hubble Space Telescope*.

Before we describe the diverse development stages that the VisMeB framework went through, a short introduction to the field of interaction design is given. The development process is explained by means of different lifecycle models. The motivation for choosing a user centered design approach is clarified.

### 6.1 Interaction Design

To start the overview of interaction design, the term itself first has to be defined.

**Definition 6.6 (Interaction Design):** Interaction Design is the process of designing interactive products to support people in their everyday and working lives.

[PRS02]

This implies no restriction to computer software. It is a more general view of products providing a possibility for users to interact with them. Nevertheless, there is a special focus on interactive software, although no limitation is necessary in many cases. The generic approach avoids this constraint.

Developing an interactive product entails a long process until a working application is created. Several steps have to be done repeatedly until a stage of satisfaction is reached. To understand how users act and react, how they communicate and interact, it was necessary to integrate people into a multidisciplinary team. A great variety of developers from diverse disciplines are involved, such as psychologists, sociologists, graphic designers, computer scientists, and software engineers, to name but a few. Figure 6.1 gives an overview of the various fields involved in interaction design.

The process of interaction design can be divided into four basic activities [PRS02]:

1. Identifying needs and establishing requirements
2. Developing alternative designs that meet those requirements
3. Building interactive versions of the designs so that they can be communicated and assessed
4. Evaluating what is being built throughout the process
6.1 Interaction Design

Obviously, these steps have to be done iteratively. If e.g. deficiencies are found during the evaluation phase, specific design changes must be made or requirements have to be adapted or introduced. Lifecycle models with a more detailed breakdown of the working steps can help to gain a deeper insight into the process of interaction design. Many models have been created, making it necessary to limit the examples to a small but representative selection. Because of the wide range of relationships between interaction design and other disciplines, lifecycle models are derived from different areas of research. Examples taken from the fields of software engineering and human-computer interaction are the waterfall lifecycle, the Spiral Lifecycle model [Boe88], or the Rapid Applications Development (RAD).

The discipline of human-computer interaction provides fewer lifecycle models than the field of software engineering. However, the strong user focus clarifies the importance of considering these models. In 1989 [HH89] introduced the Star lifecycle model, based on observing interface designers engaged in their work. In contrast to the models described above no ordering of activities is identifiable. Evaluation works as a central bridge between all activities. This allows a movement from one activity to another on condition that the evaluation activity is done first. Thus, evaluation plays the leading role in the Star lifecycle model. Every result of a completed activity has to be evaluated.

Figure 6.1: Relationships between Interaction Design and academic disciplines, design practices, and interdisciplinary fields
During the late 90’s [May99] proposed the Usability Engineering Lifecycle. Three main tasks define the model:

1. Requirements Analysis
2. Design/Testing/Development
3. Installation

Again, each stage is divided into subtasks that form the main task’s content. Task two, the “Design/Testing/Development” phase, contains the largest number of subtasks and therefore needs the largest amount of time to be fulfilled. The main difference between this model and the previously mentioned ones is the inclusion of more details. Because a detailed description would fill a complete book, just an overview is given here to gain a better insight into this area. Two main design stages will be particularly analyzed - the requirement analysis, divided into data gathering and task analysis, and the design and evaluation phase, split into conceptual design, evaluation paradigms and techniques.

6.1.1 Requirements Analysis

The usual starting point for an interaction design project is generally the replacement or updating of an established system, or a completely new development of an innovative product. Independently of this initial situation, the requirements concerning users (needs, requirements, aspirations, expectations, capabilities), tasks, and environmental variables (platform capabilities, constraints, working conditions) have to be discussed. Therefore, "identifying needs" makes up the first step. It describes the process of understanding "...as much as possible about the users, the work, and the context of work” to support the users in achieving their goals. The second step is to derive a set of requirements from these needs. This will be titled "requirements activity". Obviously, the steps are part of an iterative, not linear, process. It is impossible to give an estimation e.g. couple of weeks for finishing these activities. Furthermore, it is not unusual to return to the requirements analysis after creating and evaluating a prototype. Two kinds of requirements are differentiated in the field of software engineering: functional and non-functional requirements. Functional requirements describe what the system should do, while non-functional requirements refer to constraints on the system and its development. A typical example for a functional requirement of a graphic program may be that it should provide different colors to paint with. Non-functional requirements may include platform independency (constraint on system itself) or delivery in a period of six months (constraint on development activity). To illustrate the variety of requirements identified in interaction design, a short - and thus a not comprehensive - overview is given.

- Functional requirements: describes what a product should do.
• **Data requirements**: gather the type, volatility, size/amount, persistence, accuracy and value of the required data amounts.

• **Environmental requirements** or **context of use**: can be divided into four aspects:

  1. Physical environment (lighting, noise, dust, etc.)
  2. Social environment (collaboration, coordination, etc.)
  3. Organizational environment (user support, training resources, etc.)
  4. Technical environment (technologies used, technological limitations, etc.)

• **User requirements**: refer to the respective user group, abilities, skills, i.e. is the user a novice, an expert, a casual, or a frequent user. This has strong implications for the system requirements, which can lead to a completely different design.

• **Usability requirements**: deal with usability goals and measures for a product. These requirements are related to other requirements mentioned above, such as the degree of knowledge of users.

One main aspect of the requirements activity is data gathering. The intention of data gathering is to collect relevant, reasonable, and adequate data. Without this activity it is impossible to create a stable set of requirements. To collect data, diverse techniques can be used, e.g. **Questionnaires**, **Interviews**, **Focus groups and workshops**, **Naturalistic observation** and **Studying documentation**.

After gathering data, it is usual to consider individual tasks in more detail. This is normally done by **task descriptions**. They are used from the early phase of requirements activities through prototyping, evaluation, and testing. This implies the great importance of this activity. Three different kinds of task description are normally used and will be introduced here - **scenarios**, **use cases**, and **essential use cases**. Often, these techniques are used in combination to describe either already existing tasks, or just future ideas.

• **Scenarios**: an "informal narrative description" [Car00]. Tasks are described as stories without elaborating on the use of specific software. Stakeholders have to be able to understand the scenario, so it has to be written using their vocabulary and phrasing. Often, this can be seen as the first step in establishing requirements.

• **Use Cases**: the focus of this task description is on user-system interaction as opposed to scenarios based on the user’s task itself. However, emphasis is still on the user's perspective, not on that of the system. Scenarios can be used in this context, too, by representing one particular set of conditions. Use cases can be divided into the **main use case or normal course**, which will be performed most commonly, and **alternative courses**, which describe other possible sequences. A detailed understanding of the interaction is assumed.
• **Essential Use Cases**: represent abstractions from details, i.e. a more general case than in scenarios is described and the assumptions of use cases are avoided where possible. Essential Use Cases consist of three parts: a name, to express the user intention, a description of user actions, and a description of system responsibility.

Describing a task is one facet, analyzing it is another. Contextual task analysis has its focus on examining actual situations, not creating new products. It tries to analyze *what*, *why* and *how* users try to achieve their goals. Different techniques are known, the *Hierarchical Task Analysis (HTA)* being the most widespread one. The name gives a hint of how tasks are analyzed. Every task is broken down into subtasks, every subtask into sub-subtasks, and so on until a predetermined level of fineness is reached. Related (sub-)tasks are grouped together and describe what to do in specific situations. The main task is always the user goal, e.g. "making a call from a cell phone", or "playing a DVD in a DVD-player". If necessary, the single steps are broken down into smaller ones to get a very detailed task description. Further detailed information can also be found in [May99].

6.1.2 Design and Evaluation

When we talk about design, we have to differentiate between two types: *conceptual* and *physical* design. While *conceptual design* deals with creating a *conceptual model* to describe how the interactive product will work and what features are included, the *physical design* is concerned with design details like icons, graphics, menu or screen structures. The iterative process visualized in the lifecycle models involves users in design-evaluation-redesign loops, following the UCD approach. Evaluation plays a very important role in this context. To evaluate interactive products, it is essential to create an interactive version - a *prototype*. This can be done in different ways, from early paper mockups via clickable screenshots up to a working system. Dependent on the situation, i.e. "Is a completely new product created or is an existing one modified?", the effort for prototyping can vary. Small changes can sometimes be made to an existing product to add, adapt, or remove features without developing a completely new product. But it is still necessary to test the new features. Diverse prototyping procedures are known and used in practice. We have to distinguish between two different kinds of prototypes: *low-fidelity* and *high-fidelity* prototypes. They differ in appearance and appropriated material. While the low-fidelity versions often include paper-based versions, the high-fidelity ones look more like the final product, e.g. a Visual Basic prototype of a software system is of a higher fidelity than a paper-based mockup. To build high-fidelity software prototypes a software tool is needed. Examples are *Visual Basic*, *Smalltalk*, or *Macromedia Flash*. A low-fidelity prototype is often built using more simple artefacts. Thus, different techniques can be used, as there are e.g. *Storyboarding, Sketching, Prototyping with Index Cards*, or the *Wizard of Oz*.

Based on this comparison, decisions have to be made about which kind of prototype to use at which development stage. The idea of prototyping is to test a specific aspect of
the product being developed. Naturally, compromises have to be made e.g. functionality versus development costs. It requires more time to create a full working prototype but a paper-based one does not provide all the necessary features. Another compromise that has to be addressed is breadth of functionality versus depth. The corresponding methods are called horizontal prototyping (provide a wide range of function but with little detail) and vertical prototyping (provide a lot of detail for only a few functions). Yet another decision can influence the process of prototyping: will the prototype be included in the final product (evolutionary prototyping) or will it be thrown away after the evaluation (throwaway prototyping)? This will have a direct effect on the quality of the prototype.

To implement the requirements detected in the first phase of the lifecycle model we have to establish a conceptual model in the conceptual design step. A conceptual model can be defined as:

Definition 6.7 A conceptual model is a description of the proposed system in terms of a set of integrated ideas and concepts about what it should do, behave, and look like, that will be understandable by the users in the manner intended. [PRS02]

Therefore, some main principles of conceptual design are:

- Keep the mind open to new ideas but do not lose the focus on users and their context
- Include stakeholders as often as possible in design discussions
- To get rapid feedback on ideas and specific design aspects, use low-fidelity prototyping
- Iterate, iterate, and iterate!!!

Fudd’s first law of creativity supports the decision for low-fidelity prototypes: ”To get a good idea, get lots of ideas” [Ret94].

During the development process of a conceptual model, different design aspects arise. You have to think about

1. Which interaction mode should be used?
2. Does an appropriate interface metaphor exist?
3. Which interaction paradigm should be used?

For a better differentiation of conceptual models, we can split them into two main categories: activity-based and object-based models. Activity-based conceptual models
can be divided into different actions, like instructing, conversing, manipulating and navigating, exploring and browsing or a mixture of these.

Object-based models orient themselves on objects or artifacts, like a calendar or a book. Because of their tight connection to a particular object, these models are often more special. A prominent example of this kind of conceptual model is the spreadsheet [Win96].

Interface metaphors are another way to delineate a conceptual model. The idea is to take something known from the physical world and transfer it to the virtual world, while adding specific features and characteristics. The previously mentioned desktop or spreadsheet, or the recycle bin, are typical examples. Although there are benefits from using interface metaphors, such as the fact that users are familiar with the (physical) device and therefore are supported in understanding and learning how to use a system, there are also drawbacks. Some rules familiar from the real world are broken (e.g. no one would place a recycle bin on the desktop instead of under the desk), which can possibly - but not necessarily - lead to confusion. Interface metaphors can be too constraining, so useful tasks might be left out although they would improve the interface. Conflicts with design principles can arise that lead to bad design solutions, or to the system’s functionality beyond the metaphor not being understood. Sometimes, existing bad designs are translated literally, which is another trap designers can fall into, or the imagination is limited in creating new paradigms and models by the use of ideas that are based on well-known technologies.

Interaction paradigms are positioned on a more general level of development. A paradigm decides the general way of interacting with a system. For a long time the main paradigm in interaction design was restricted to the fixed artefacts computer, monitor, mouse, and keyboard creating WIMP interfaces (Windows, Icons, Menus, and Pointers or Windows, Icons, Mouse, Pull-down menus). Nowadays, we think in larger connections, which can lead to completely different paradigms, for example Ubiquitous computing, Pervasive computing, Wearable computing, Tangible bits, augmented reality, and physical/virtual integration, or Attentive environments.

So far, a lot information about lifecycle models, requirements analysis, design of systems, conceptual models, and so on has been introduced. However, one important task is still to be discussed - evaluation. As we saw before, iteration is indispensable. To get feedback about ideas that we try to realize in prototypes, we have to test these implementations. This can be done in many different ways. General evaluation paradigms are e.g. a "Quick and dirty” evaluation, Usability testing, Field studies, or Predictive evaluation.

These paradigms give a good overview of the way evaluations can be done. However, it is still important to consider the techniques that can be used. Dependent on the respective paradigm, some techniques can be leveraged in different ways. Observing users,
Asking users, Asking experts, User testing, and Modeling users’ task performance belong to these techniques.

A lot of different paradigms and techniques have been presented so far. To conduct an evaluation, however, we still need a simple guide - an explanation that describes the proceedings step by step. This can be done by the DECIDE framework, for instance. It provides a checklist to help inexperienced evaluators performing an evaluation. Again, DECIDE is an acronym that now will be explained letter by letter.

The DECIDE Framework:

1. Determine the goals: first of all, you have to investigate the diverse goals that are pursued. The main questions in this context are: who is the user? Why does he want the evaluation?
   When the goals are decided, they can serve as a guide through the whole evaluation.

2. Explore the questions: as soon as the high-level goals are determined, questions have to be generated. Every goal can be investigated by a great variety of questions. Let us assume that one goal is to find out why people still visit their local bank branch instead of using online banking. Possible questions could be: do users trust the system or are they sceptical about security? Is it hard to fulfill a task because of a badly designed interface? Does every user have the possibility of using online banking, i.e. is the infrastructure in place?
   Questions can be broken down into sub-questions, and these sub-questions can again be broken down, and so on. A typical example for a question concerning a badly designed interface could be: is enough feedback provided? Is it easy to navigate? Is the wording consistent or is it confusing?

3. Choose the evaluation paradigm and techniques: the next step deals with the choice of paradigm and techniques. Depending on the evaluation paradigm, diverse techniques can be used. However, there are still other issues to consider like time, money, or equipment. These can lead to a completely different selection than would have been made under other circumstances. The next item will consider this aspect in more detail.

4. Identify the practical issues: before the evaluation is conducted, you have to identify some practical issues. These include finding appropriate users, assessing the equipment, considering time and money constraints, and the know-how of the evaluators.
   Test users have to represent the complete target group (if possible). Equipment and facilities have to be in place, e.g. camera, batteries, or empty recording tapes. Schedule and budget are very important and have a strong influence on e.g. the number of users and the kind of technique. Evaluators have to be prepared for the specific kind of evaluation that is to be undertaken. It is not meaningful to use e.g. videotaping if there is no expertise and equipment to analyze the results.
5. **Decide how to deal with the ethical issues:** the privacy of test people is a point that must not be neglected. Personal information about life circumstances (like education, age, illnesses) should be confidential. No name should be associated with a specific questionnaire or collected data, unless users are in agreement. People that participate in an evaluation should be treated with respect. An "Informed Consent Form" is therefore indispensable to inform the users about their rights, and also their responsibilities. They should know what the goal of the study is, how the process will run and approximately about how long it will take. It should be made clear that the users can stop the evaluation at any time they wish. A good motto for treating the test person in the right way is: "*Do unto others only what you would not mind being done to you!*"

6. **Evaluate, interpret, and present the data:** still open are questions about what kind of data is to be collected, and how, and about the methods for analyzing and presenting it. Keywords for this approach are [**reliability**](#): Is an experiment repeatable on different occasions and does it yield the same result under the same conditions?, [**validity**](#): Does the experiment really measure what should be measured?, [**biases**](#): Are there any distortions that influence the result, e.g. the interviewers tone of voice?, [**scope**](#): Is it possible to generalize the results?, and [**ecological validity**](#): Does the environment have an influence on the results?

To reduce the number of possible inconsistencies in the evaluation, a [**pilot study**](#) should be done before starting with the main study. Inexplicit questions can set the user on the wrong track, inoperative equipment can lead to an unintentional delay. If problems arise, they can be solved in advance. This enables an undisturbed and trouble-free performance of the evaluation.

During the development process of the VisMeB system, a variety of evaluations, ranging from predictive to laboratory evaluations (see also [RLK+03]) were made. The next section will describe the different stages in detail.
6.2 Development Stages of VisMeB / MedioVis

The first ideas for the Visual Metadata Browser VisMeB and its successor MedioVis were born after the evaluation of the INSYDER system, which was developed at the University of Konstanz [Man02], [Mus02], [RTM05]. Thus, there was no need to develop a completely new system, but nevertheless the source code was structured and organized in a totally new implementation. Experiences from the INSYDER project could be integrated into this further development, not least because of the strong similarity of the application domains.

Users: again, the system has to be used by experts in their field, in this case experts in the area of geographical information systems (GIS). Therefore, they were familiar with search engines and business graphics, which made it easier to create a satisfying system that included the knowledge gained from INSYDER.

Tasks: the task to be fulfilled is divided into different steps. First, a query has to be formulated. This can be done in various ways e.g. by visual support or simple text formulation. After sending the query to the underlying database, the result set is presented and can be explored. Diverse methods are known, dependent on task and user. On the one hand the visual seeking system should support the user in browsing the data if he is not sure about what exactly he is looking for. On the other hand specific fact finding (e.g. *Who is the director of the movie Casablanca?*) and extended (e.g. *Name five movies with Charlie Chaplin as main actor!* ) has to be possible. According to these tasks, sometimes a query reformulation is necessary and therefore has to be facilitated. So we will return to the first step and go through the process again. As a last step, it is desirable to be able to process all the extracted information as the conclusion of the self-contained visual seeking system. Very often this last step is neglected and the user finds important information but has no possibility to work further with it. This is another feature of the current MediaGrid version implemented in the VisMeB / MedioVis framework.

The conceptual model of the VisMeB / MedioVis framework is based on activity. As is usual in real life projects, a mixture of different actions forms the basis for the system being developed. *Instructing, manipulating*, as well as *exploring and browsing* play an important role in this context.

The results of the INSYDER evaluation suggested a table-based layout enriched by various visualizations. Thus, the general concept of VisMeB / MedioVis is based on three main ideas. Develop a visual information seeking system which:

- uses diverse visualizations to improve the information-seeking process,
- combines some of these visualizations in a familiar table, and
- provides the information in different levels of detail.

The first idea is implemented within multiple coordinated views that allow an exploration of data from different viewpoints. Depending on task and user, one or other of the...
visualization is better for solving the particular problem.

The second rational was implemented in the SuperTable approach. A simple table, which a wide range of users are familiar with, serves as frame for a sophisticated visualization. The table has two significant new features: the granularity concept, which distributes the information to different levels of detail, and the increased use of visualization, not an addition to the table, but as an integral as part of it. This combination enables the user to get all accessible information in manageable chunks without being forced to switch the context, as was the case in e.g. the INSYDER system.

The third design rational is based on various assumptions. First, the user should be enabled to find all the information he is looking for. Nevertheless, being confronted with all the available information at the very beginning would overwhelm the user. Thus he would not be able to explore the result set in a meaningful way, but only by simple browsing of all the pieces of information. Compressing the information to get a good overview seems to enable good access, but nevertheless detailed information has to be provided if task-dependent, interesting data sets are found. Therefore, the concept of diverse levels of detail was chosen.

The physical design of VisMeB was strongly related to the one used in INSYDER (see also [LMR+02], [MRKE02]). Small variations were made but the main layout stayed the same. Nevertheless, as a result of the introduction of the SuperTable and its inclusion of visualizations and detail levels, adaptions had to be carried out. One main problem was the way of implementing detail level changes. As we will see in the following chapters, different approaches were chosen until the actual version was found. The implementations ranged from simple buttons for single levels, through a slider to change from one stage to another, to a direct intervention in the single cells of the table. Another challenge was posed by the way the interaction between the diverse visualizations was achieved. The main question was "Will the users understand the method of interaction proposed in the system?".

Following the UCD process, different implementation steps were taken, characterized by prototypes and evaluation steps. Figure 6.2 gives an overview of the different development stages.

The description of the following evaluations will go into the process of development in detail. Goals were defined and questions derived, paradigms and techniques were described, as were the additional techniques and models that were introduced and used.

### 6.2.1 Paper-based Mockup

The first prototype was an early paper mock-up to test the idea of different detail levels and interaction techniques. Therefore, screenshots were produced and printed out on DIN A3 sized paper, together with menus and interactive elements. These were then laminated and they worked as a conceptual model mockup, which represented the VisMeB system.
Figure 6.2: Timeline of development stages during the user-centered design process

on a limited-interactivity level, but a high conceptual one (see also [LRKM03b]).

Figure 6.3: First paper mock-up of VisMeB to test the initial ideas within a user test.

The main usability goal for the test setting was to evaluate if users would accept the interaction concept of VisMeB. A second goal was to evaluate the ScatterPlot visualization. The aim was to find out if and how users would work with it, and if and how they would use a combination of the result visualization in the SuperTable and the Scatterplot. Typical users in this context were experts in the field of information-seeking systems or, as a minimum, in general computer science. This implies a general understanding of business graphics like bars, but also an ability to adapt to new visualizations after a short training period. In the preceding case, eight test users from the computer and information science department participated in the test; all had experience in computer science, only a few in information visualization.

In a test setting, the test users were asked to interact with the prototype. The participants should explain how they would solve a variety of tasks with the help of the prototype, e.g. where they would click, and which mouse button they would use. Depending on the course of action, different elements representing interactions (sheets of paper) were put
into the prototype. Users again acted and reacted with them. They were guided through a series of tasks and later asked to rate the conceptual adequacy of the prototype. Furthermore, after the test a detailed questionnaire was completed and a structured interview about the look and feel of the prototype was conducted. Here, users also had a chance to give their personal impressions. Questions in this context concerned e.g. Do the users understand the connection between the SuperTable and the ScatterPlot? Does the interface provide enough information to work with the granularity concept without a long training period? Do the users understand the use of colors, i.e. does the use of the same color in different visualizations imply an intentional connection?

As an evaluation paradigm, we chose the usability testing one. An office in the computer and information science department served as a usability laboratory. The techniques used were a composite of observing and asking users. During the test, users were observed while solving the tasks. An effective questionnaire should give users the possibility to advance their own opinion concerning the prototype itself, its interaction and general concept and layout.

The results of this early usability test addressed two thematic areas. First of all the interaction concept was found to be a valuable aid in displaying large amounts of search results. All participants thought that visualizing search results was more effective than the mere textual display. Seven out of eight stated they could imagine working with the visualizations on a regular basis and all participants stated that they could work with the prototype again even after a long break. Secondly, the visualizations themselves have to be examined with regard to their value for each user-task. Therefore, scenarios with realistic user behavior should be established and form the basis for further redesign. Once non-domain-expert users have tested the interaction and general functionality, domain-experts should be consulted and involved in the user-centered design.

Keeping in mind that the interaction concept is very new and quite unfamiliar to normal users, creativity and unorthodox thinking is required. Looking beyond one’s limits and possibly beyond the limits of “common” human-computer interaction might lead to a host of radical, yet improved, satisfying new visualization and interaction concepts. These evaluation results encouraged us to proceed with our design process.

### 6.2.2 HTML Mockup

As a second step, two HTML mock-ups visualizing the design variants Level- and GranularityTable were built to enhance the user’s possibility of interacting with the system. The lessons learned as a consequence of the former evaluation were implemented in this version. By now, the users were able to get direct feedback when invoking an action. In October 2002 the html-mockups were tested by several users (n=8) from the expected
INVISIP target user group of civil engineers from traffic planning authorities and privately owned planning offices. Although there were individual differences in working and searching habits, the majority stated that they had very high expectations concerning the work with geodata and that they expected higher efficiency with a working metadata browser. The most popular characteristics of geodata were cost-effectiveness and the potential for rationalizing daily work tasks with geodata. The browser should give immediate access to geodata and provide enough flexibility to adapt to the various tasks in a spatial planning process. The most important geodata-attributes corresponded to maps, e.g. coordinate system, scale and precision, but language, year and origin of the dataset were highly relevant as well.

After the pre-test questionnaire and a video introduction, the users were handed a script with test tasks to work on. A test monitor and a recorder conducted and documented the test, while a video camera and a screen recording took footage of the user and his actions on the screen. This proved to be especially helpful (though very time-consuming) for re-evaluating critical situations in the test, where we could view and analyze the two synchronized videos.

Users approved of the two design variants. In particular, the granularity table led to some surprisingly emotional responses from those working with it. For example, when users found out that they could change the granularity of a document by a context menu in the Scatterplot, most were surprised and pleased with this feature. Comments varied from "very clever" to "unusual, but I like it" (see Figure 6.4 left side). This, and the possibility of scrolling through an extended abstract in granularity level 4 (see Figure 6.4 right side), were critical from our point of view, but approved of by users in this demonstrator’s interaction design.

![HTML Mock-Up: The Scatterplot allows the level of document detail to be changed (left), a scrollable abstract in the GranularityTable provides further information (right)](image)

**Figure 6.4:** HTML Mock-Up: The Scatterplot allows the level of document detail to be changed (left), a scrollable abstract in the GranularityTable provides further information (right)

In the posttest, users were in favor of the GranularityTable, but our detailed analysis
showed that their actual performance did not conform with this view. Users had more problems in fully understanding the interaction concepts of the granularity design than those of the level design. We assume that users were positively influenced by the clearer and more aesthetic and appealing design of the GranularityTable, and therefore gave it better ratings than the LevelTable. We can rule out learning or last-item remembering effects. Nevertheless, this bias in the test design had to be eradicated for the next user tests. Both demonstrations lacked intuitiveness concerning the keyword highlighting and relevance attributes. To our surprise three (SuperTable) and even four (Granularity Table) users had problems in matching the designated colors with the keywords and their relevance values. Even though this aspect of the metadata browsers was explained to the users in a video before the test, the conversion of this knowledge into their mental model of working with a metadata browser seemed difficult.

Another problem was the SegmentView with its stacked columns or TileBars, which in both demonstrations gives an overview of the segmented and rated document. An example of the SegmentView with stacked columns from the LevelTable is seen in Figure 6.5, top; the TileBars can be seen in Figure 6.5, bottom.

**Figure 6.5:** SegmentView in StackedColumn form at the top (each bar is a text segment, relevance is shown by size and colour) and in TileBar form at the bottom (each line is a text segment, relevance is shown by saturation. Navigation in text is by the upper left arrow)

It did not occur to users that a segmented document was being visualized. None of the eight users understood this concept within the LevelTable, and only two users managed to work with this in the GranularityTable. In the latter, users were more open to trying to work with it after we explained the interaction. Then, some also rated this interaction with positive attributes like ”clever” and ”very usable for longer texts”. This was in contrast to the LevelTable, where most users showed little interest in the feature, or wanted to work with it at all. We suggest that the vertical alignment of the TileBars of the GranularityTable seems more like a ”normal” document structure (top down, left to right) and is therefore more easily accessible by users for the reception of a text.

During the test and also at the posttest, users frequently criticized a lack of connection between the Scatterplot and the corresponding SuperTable (both design variants). A dual visual response when documents were marked or the level of detail was changed seemed to be insufficient; users requested very tight coupling of both views.
In parallel with the lab tests, we started a web-based evaluation. The target user group were people involved in spatial planning, and comprised mainly of co-workers from the project or their colleagues. Questions were asked about individual search behavior, about a virtual search with the two design variants, how the users would interact with them, where they expected which interaction, and what they would like to be different. The participants were asked to download two short introductory videos, and several screenshots correlating to the tasks. The sequence of Level-/GranularityTable was randomized to exclude learning and last-item remembering effects. 35 users completed the questionnaire, from which 31 were put into the final evaluation.

Although screenshots are even more limited than the prototypes of the lab evaluation, some results from the former evaluation were confirmed. Throughout the test, the performance (measured in correct answers regarding interaction) was higher with the LevelTable than the GranularityTable. A lack of connection between the table visualizations and the Scatterplot was also frequently criticized. An interesting result came from the analysis of search behavior and preferences in design. With five separate questions concerning typical search tasks, we wanted to classify the users as being of the "more analytical" or "more browsing search-strategy" type [Mar95]. As might be expected, a mixture of both strategies dominated the sample. Only eight users had very clear preferences; five of them were categorized as "only browsing strategy", three of them as "only analytical strategy". Interestingly enough, four of the first category definitely preferred the GranularityTable, and all three of the second category preferred the LevelTable. We assume that, at least for the first steps of an iterative search process, the LevelTable can be efficient for analyzing the result set as a whole; it can find patterns, or reformulate/discard the query in the event of unsatisfactory results. The primary goal is not content, but filtering and reduction of the result set. If the results are then narrowed down to potentially interesting documents, the GranularityTable with its browsing comfort can be used. Now content is the primary goal; modalities can be changed frequently. In this manner, our initially-developed scenarios were partly validated by empirical results, though our scenario characters begin the planning process with only analytical and very formal queries formulated in a sophisticated manner, while during the planning (and possibly the iterative retrieval) process they become more informal and data driven.

Although this evidence should not be given too much weight, we took it as a signal to handle both design variants equally and to further integrate them into the overall SuperTable/Scatterplot framework. A first step was to implement both versions in Java. Thus, the VisMeB system was introduced. All visualizations and proposed interaction techniques were implemented to provide the possibility of working with them in a running system. The next step should be to integrate both table ideas into a single one. The aim was to present an even more convenient way of handling the problems we were confronted with, such as e.g. the change of detail levels, or the complete integration of all
information-seeking process steps into a single system in a meaningful manner. Thus, the MediaGrid was developed. It combines the characteristics of LevelTable and GranularityTable as well as newly-introduced features to build a self-contained system. Again, different variants named MediaGrid and MovieVis were developed. This provided the opportunity to test diverse approaches.

6.2.3 Java Prototypes

The first Java prototype including the test results was implemented within the VisMeB project, i.e. using the INVISIP geographical meta-data domain. All versions of the SuperTable so far were integrated - the LevelTable and the GranularityTable. Tests were conducted and will be presented in the following section.

When the INVISIP project came to an end, a new application domain was sought - and found in the Mediothek of the University of Konstanz. A new prototype was therefore built, adapted to the situation and given the name MedioVis. Again, two versions of the SuperTable were implemented. The first one, the GridTable, was to provide an intuitive and easy-to-learn alternative to MOSAIC, the current search catalogue of the university’s library. Highly sophisticated visualizations and interaction possibilities were avoided to enable non-expert users to handle the system without long training periods. The second one, the MediaGrid, provided more features that in part need some period of vocational adjustment. The target group were administrative employees who work with the system nearly every day and have much more experience with sophisticated systems than normal users. An evaluation of the MedioVis system is still ongoing and more details can be found in the Outlook, Chapter 8.

6.2.3.1 The VisMeB Prototype

The results of the test described above were validated and consolidated in a first prototype Java implementation for the VisMeB project. The Java test sessions involved more and better interaction possibilities, which were to be used during a typical search process with VisMeB. While the HTML mock-up still had restricted interaction, the users now had all the facilities they were used to having when working with a completely functional system.

The test scenario was developed using a search with three query terms on a database containing crawled WebPages about geographical information systems (GIS). The participants (n=8) worked with the SuperTable in both design variants. They were project members of the INVISIP project and therefore from the user domain of spatial planners. It is important to note that all had considerable computer literacy, but none of them had detailed experience with visual information retrieval. In further test sessions (see Chapter 7), we evaluated the efficiency of VisMeB and the SuperTable design idea. This was not the focus of the tests described in the following; instead, we wanted to concentrate on qualitative usability aspects. Two hypotheses, mainly derived from the results of the
preceding usability tests, were initially defined:

- $T_1$: users who prefer a more browsing-oriented style of searching will prefer the GranularityTable.

- $T_2$: for creating the detailed relevance curve, TileBars are more intuitive than stacked columns due to their vertical orientation, which behaves like an enhanced scrollbar, and to the tight coupling of tile and text.

There was no evidence that supported hypothesis one. Users who described themselves as browsing oriented and were observed behaving accordingly during the tests did not show any significant preference for the GranularityTable, nor did they differ in any respect from the other users.

The second hypothesis has also been invalidated. All participating users could work fairly easily with the detailed relevance curve version in the LevelTable, but sometimes had severe problems with the TileBars visualization in the GranularityTable. They did not understand the conceptual model behind the TileBars. Six out of the eight participants preferred to work with the stacked columns. Asked why they preferred this variant, four participants had no answer, but two mentioned that it seemed natural to read a text from left to right. Accordingly they used this left-to-right visualization, horizontal, in contrast to the top-down approach of the TileBars. This conceptual mapping of left-right movement in the visualization and its translation into the corresponding highlighted segment of the document text was observed as being quite marked. Therefore a possible design solution for the TileBars might be a change from a vertical to a horizontal orientation.

The overall results of the LevelTable were more encouraging than those of the GranularityTable. Not only did the users give it better ratings and express more positive opinions, but they also worked with fewer errors and misunderstandings. User comments additionally suggest that, in general, this table-based visualization is easier to use than the less common GranularityTable. Nevertheless, the GranularityTable has benefits. The freedom to explore different levels of detail side by side within one result set was mentioned by three users. They could also see how to integrate the GranularityTable into their respective model of work as spatial planners (e.g. to search and compare information in short texts, technical definitions etc.) All users accepted the GranularityTable itself, but the training effort that is needed to use it effectively is definitely higher than with the LevelTable.

Asked which level of detail the users would prefer as the initial view of the result set visualizations (i.e. when the search results are presented for the first time), level 2 of the LevelTable was named by six out of eight users. As one participant stated: "It gives the feeling of Google but I can check many more results at once". This strong preference of level 2 may also have been the result of a usability problem in both of the level 1 views. All participants found these overview levels tricky to operate. The fisheye technique at the
first level in both variants of the SuperTable (see Chapter 3) enlarges the row height from 3 pixels to 25 pixels (default values). Users had no problems in recognizing the benefits and working conceptually with it, but the interaction itself was difficult. Although these pixel values of the fisheye effect can be adjusted individually, the default size should be bigger and the fisheye view should include rows above and below the mouse pointer to ease the transition between the highlighted focus and its context.

If the users had to compare the importance of several documents, 66% would solve this task with the first two levels of the LevelTable. To find the most important document regarding each single keyword, 80% would choose the LevelTable as well. As seen in Figure 6.6, the single keyword relevancies are visualized from the start and can be compared in columns next to each other up to level three. We conclude that, for assessing the distribution of keyword relevancy, level 1 or level 2 of the LevelTable are most suitable. For further exploration of document details, both the Level- and GranularityTable are suitable, depending on individual user preferences. For displaying the whole document details, level 4 of the LevelTable with the detailed relevance curve using stacked columns in combination with (multiple) browser view(s) is the best visualization.

Figure 6.6: VisMeB level 2 supporting the recognition effect for keywords by color in column three (geo in green), four (information in red), and five (system in purple)

Although the GranularityTable performed less well than the LevelTable, we are inclined to pursue its development. As mentioned before, the user group tested was not composed of specialists in visual information retrieval. Further tests can determine if a user group like e.g. information brokers with an extended training period can use both versions of the SuperTable efficiently, or if the GranularityTable has to be scaled down to a mere add-on to the LevelTable, or possibly even discarded.

In a further test session, the efficiency of VisMeB was evaluated by comparing the SuperTable design idea to a traditional result list. This is the subject of Chapter 7. An evaluation was also made to compare our idea of a visual query preview versus a form-based interface. This is the focus of the work of [Kle05]. All these tests lead us to the currently existing MedioVis framework. The advantages of both tables were included in the MediaGrid and additional features round off the system.
6.2.3.2 The MedioVis Prototype

As described above, the MedioVis prototype was created with the intention of satisfying two different user groups - “normal” library users and administrative staff. To meet the first users group’s need, the GridTable is implemented (Figure 6.7). The layout resembles the LevelTable introduced in the VisMeB framework, but fewer features and visualizations are available. It is restricted to an interface that does not overstrain the users. All in all, ten different levels in the two SuperTable variants would certainly produce an overhead that would deter the users from further work with the system. We still have to keep in mind that the idea of the global project was to provide a system created for experts, not for occasional users. Thus, our objectives had to be adapted to the situation and the user’s abilities. Figure 6.7 displays the GridTable in its original version.

![Figure 6.7: GridTable implemented in the MedioVis project to present information to a library user](image)

The strong similarity to the LevelTable can not be ignored. Differences can be found in e.g. the way of changing the level of detail. A specific number seemed to be less meaningful than simple “+” and “-” buttons, supported by a graphical presentation as a kind of stairway. At a later stage, the names changed from ”LevelTable“ to ”Table View“ and from ”ScatterPlot“ to ”Graphical View“, to describe the corresponding visualization in words the user is familiar with. In the original LevelTable the different levels provide different information about the data sets. Nevertheless, additional visualizations like the SegmentView or BarCharts were omitted to minimize the number of new techniques to be learned.

In contrast to the GridTable, the MediaGrid can be seen as a more research oriented development. New ideas and concepts are given a chance and are developed concurrently with the GridTable implementation. The possibility of zooming into single cells instead of moving the whole data set from one detail level to another is the main advance in this
SuperTable version. Additionally, further information resources (like the IMBD\footnote{International Movie DataBase, http://www.imdb.com}) are used to include other material like movie posters or images of actors. The new concept even allows the playing of a trailer in a cell. Visualizations like the SegmentView, Bars, or even the LocationMap can be integrated again; no specific level has to be determined to display them. The necessary space is defined by the size of the visualization or the length of the presented text, not the level that provides a fixed height or width for rows and columns. Figure 6.8 shows an example displaying a poster in one column, a trailer in a second, and the LocationMap in a third one.

![Figure 6.8: The MediaGrid of the MedioVis project. A zoom to single cells is possible, resulting in e.g. the display of a movie poster (left column), a trailer (center), or the location map (right column).](image)

To find out if users are comfortable with the MedioVis system further evaluations have to be undertaken. The DROID project has therefore been established. More details about its current status can be found in the Outlook, Chapter 8.

### 6.3 Summary

This chapter gives a short introduction to the field of user-centered design. As part of interaction design, several lifecycle models are introduced that describe the complete development process. A very important aspect is the iterative approach that leads to a well-designed product. It is impossible to define requirements that remain completely unchanged during the whole lifecycle. Feedback has to be obtained by different techniques, as described above. Examples of interaction and design paradigms as well as data-gathering techniques are cited. Furthermore, the diverse development stages that the VisMeB project went through are considered, starting with a paper-based mockup, to a working system implemented in Java, i.e. the final implementation within the MedioVis.
project. The importance and advantages of a user-centered design process are explained, emphasized by several examples. This knowledge forms a basis for understanding Chapter 7, where a simple list and the LevelTable are compared.
The complete implementation of the VisMeB framework followed the user-centered design process (see Chapter 6). This included tests using paper mockups, HTML mockups and a working system implemented in Java. But all tests focused on the question, of whether users could handle the system, whether they liked the idea or not, what was good and what was bad, and so on. So far, the question if the visualizations used lead to an improvement in efficiency, was not considered. Therefore, there was a need for a user study to verify and quantify the benefits of the SuperTable and measure the subjective user preferences. As a result, in summer 2003 an evaluation was performed as part of a Bachelor thesis [Ger03]. The test’s objective was to compare the SuperTable with a typical list-based visualization such as that e.g. Google. We decided to test these visualizations, because at the present time only a few users are familiar with visualizations similar to the SuperTable. A list or simple table presentation for results of a database query is the usual way to present the information retrieved from a database. The LevelTable was chosen in accordance to a heuristic evaluation made initially by the developers of the system. Information concerning the statistical background can be found in [AA03], or [KBW00]. A detailed report of the evaluation will now be given.

7.1 Introduction

To guarantee the validity of usability tests, an appropriate analysis is indispensable. Depending on the kind of evaluation, statistical methods, applied in a correct manner, can fulfill this condition. In the present case, a performance test to compare two visualizations - list and LevelTable - complies with the conditions. As we have seen in the preceding chapter, the first step in starting an evaluation is to define the goals that are to be attained. Depending on these goals, the questions can be formulated to ensure proper results. This can be done by formulating hypotheses. One possible hypothesis is the null hypothesis. It assumes that there is no effect between the dependent and independent variable. Nowadays the use of multiple dependent and independent variables is usual, although the analysis becomes more complex. In our case, the dependent variable is given by time (performance) whereas the different visualizations used (list or LevelTable) can be de-
scribed as the independent variable. Therefore, the null hypothesis $H_0$ and the research hypothesis $H_1$ for the present case can be written as

- $H_0$: "Concerning effectiveness, the LevelTable does not have any advantages over a list-based visualization."

- $H_1$: "One result presentation is more efficient than the other."

If the test participants solve the task faster with one visualization than the other, the null-hypothesis $H_0$ can be rejected.

Typical analysis methods for usability tests like the present one are ANOVA (ANalysis Of VAriance) and Student’s T-Test. The dependent variable is given by the performance, i.e. the time users need to fulfill a task. The independent variable is available in two variants, visualization A (list) and visualization B (LevelTable). For that reason, a Student’s T-Test or an ANOVA should be used to get a correct interpretation. This is conditioned by the respective situations, which influence the choice of the corresponding method. In this case, both methods are used.

To compute the statistical analysis, the widespread and reliable SPSS system was chosen. Among other possibilities it provides output in the form of tables that display the values that are most important in coming to a decision concerning a rejection or retention of the null-hypothesis, as can be seen in e.g. Table 7.1.

Besides the correct evaluation method, there is yet another decision to make. What kind of test design is chosen? Possible test designs are the between-subjects design and the within-subjects design. Both versions are widespread and commonly applied. For the present case a between-subjects design was chosen. The visualizations to be compared were limited to the SuperTable, or rather the LevelTable as one possible version, and an additionally created list visualization, based on the typical layout of a search result presentation such as that known from e.g. Google. All other visualizations were excluded. One reason was the additional training time that would have been necessary to work in an efficient manner with additional visualizations like the CircleSegmentView, or the ScatterPlot in its two- and three-dimensional versions. The functionality provided by the system could overwhelm the user and thus confront him with problems that would lead to less meaningful results. The purpose of the VisMeB framework was to support professional users at their work, so a longer initial training is reasonable and acceptable. But this extra time could not be spent on the usability test, which is very restricted to a specific time frame of at most two hours - including introduction, pretest, main test, and posttest.
7.2 Test Setting

The test setting has to be described in different sections, because different tests were carried out. First of all, a pretest had to be undertaken to check the test setting for inconsistency, imprecise questions and similar, possibly disturbing factors. Afterwards the test persons were asked to fill out an entry questionnaire to ascertain some demographic data. This part was followed by the main test, again split into diverse tests - the baseline test (Test 1) and the actual main test (Test 2). Finally, a posttest provided the possibility of giving feedback to the system, partly answering direct questions, partly entering free text.

7.3 Pretest

A pretest with members of the Department of Computer and Information Science of the University of Konstanz lead to some small refinements, but the test setting itself did not need to be changed. The test phase could thus start without any time delay.

7.4 Test Persons

The group of test persons consisted of 32 participants, among them 8 women and 24 men. The results from only 30 out of the 32 users could be used for evaluation. One reason was that, for one user, the video recording failed and the data recorded by the minute-taker were too imprecise without the help of the tape recording. A second data set was removed from the group of results to be evaluated because the user displayed extreme "thinking aloud", which was not suppressed during the test session and lead to long explanation breaks. All other test results passed into the evaluation. A detailed description of the participants and their characteristics is given in Section 7.5, Entry Questionnaire.

7.5 Entry Questionnaire

A questionnaire to gather the test persons’ demographic data like age, sex, or education, as well as their daily use of computers and experience with search engines and the internet in general, was presented at the beginning of the test. The detailed questions and results can be found in appendix A; only a summary will be given in this section. The average age was 26, and most of them were students at the University of Konstanz. On average, every test user spends 5 hours a day working on a computer, so most of the participants considered themselves as experienced users. 28 had a university-entrance diploma (German: "Abitur"), 15 persons were members of the Department of Computer and Information Science. All participants were familiar with internet search engines, especially with Google which was stated to be the favoured search engine for 100% of the
users. On a seven-point Likert scale the mean experience with search engines was described with a value of 4.8 where 1 conforms to "no experience" and 7 to "very good experience". On balance, two different user groups could be identified. The first group consisted of students of computer science and persons working in this field, the second of people working with computers, but not exclusively in this domain. To differentiate these user classes, group one will be called experts, whereas group two will be called non-experts. This classification will not influence the evaluation but explains the sometimes diverse estimations carried out by particular test persons. Collectively, all users fulfill the pre-condition of being familiar with computers and search engines.

The entry questionnaire had no influence on the subsequent maintest, but helped us to understand the technical and educational backgrounds, to ensure the right selection of test participants. As stated above, the typical users for the available VisMeB version are experts, not beginners. This can be proved by the results of the entry questionnaire.

7.6 Maintest

To compare and validate the results of the between-subjects design evaluation, two groups were formed - the so called Control Group (CG) and the Experimental Group (EG). Members of the groups were selected randomly to reduce the effects caused by a specific and targeted choice. Because of an organizational problem, 18 persons were assigned to the experimental group and 14 to the control group. This distribution has no effect on the results because a oneway ANOVA was used and thus the values can be computed by a harmonic mean from both groups. In the first test phase (Test1 - the Baseline Test), both groups had to work with the list-based visualization. The objective was to reveal differences between the two groups. In the following test phase (Test2), the tasks were split up: whereas the control group stayed with the list-based visualization, the experimental group had to use the VisMeB LevelTable visualization. To prepare the experimental group for the unfamiliar visualization, each group member had to watch a short tutorial video about VisMeB and the LevelTable. After finishing the second test phase, the users answered a posttest questionnaire to describe their subjective impressions. This only concerned the experimental group because the topic of the questionnaire was the LevelTable which was not worked on by the control group. Additional information was gained by asking questions involving a comparison between the list visualization (which all users worked with in the baseline test) and the LevelTable, and thus the advantages and disadvantages could be determined.

To get a structured overview of the detailed test setting, the 5-T environment (see Chapter 3) is used. Therefore, the tasks, the type of data, the type of user, the technical environment, and the training period will now be described.
7.6 MAINTEST

7.6.1 Tasks
The tasks were the same for both groups. For instance, they were asked to find a specific document, compare several documents, or even extract specific information from an unknown document. These tasks are based on the task taxonomy proposed by [Shn98] and can be split into specific fact finding and extended fact finding. Three types of tasks were used, where the first two can be assigned to specific fact finding and the last one to extended fact finding.

- **Type 1 - Search documents**: a specific document had to be found, which was defined by concrete attributes like relevance or language. All questions had to be answered by a base query; no additional query was to be used. Five of the twelve questions fell into this category.

- **Type 2 - Compare documents**: in this case the documents had to be compared by means of specific meta-data, like size or language. Again, the base query had to be used. Four questions were assigned to this class.

- **Type 3 - Compare documents’ content**: in contrast to type 2 the questions focussed on finding a specific meta-data attribute and specific information contained in the documents. A new query was allowed, and the aim was to find information without scanning all result documents’ content. Three questions fell into this category.

All test participants had to answer twelve questions in all. Eleven of them were to be solved within the initial query; for the last question, additional queries could be made. Query terms were established for these first ones to avoid mistakes and to lay equal foundations for all test persons.

7.6.2 Type of Data
A database (the GISWeb database, created from a webcrawl and stored in a PostgreSQL database including a variety of additional metadata) containing about 2000 websites concerned with the field of geographical information systems (GIS) served as a base for the test questions. To ensure a manageable size (as regards performance and the documents’ relevance) a subset of 300 documents was chosen and stored in a separate file. This was used to supply the result documents.

7.6.3 Type of User
All test persons could be classified as experts in the field of computer and search engine use, although their knowledge of the specific domain of geographical information systems was very limited. No site planner or other GIS expert was involved.

1http://www.postgresql.org/
7.6.4 Technical Environment

An office of the Department of Computer and Information Science served as the usability test laboratory. It was fitted out with the test computer, a 19" TFT screen, a video cam, and a mirror. The computer was equipped with a 1.6 GHz processor and 512 MB RAM, which are the most relevant computer characteristics important for the test. A screen resolution of 1280x1024 dots ensured an excellent image. The video cam was located to record the user’s actions and the screen at the same time. Because of insufficient computing performance, an additional screen recording via a screen cam was ruled out. The environmental conditions were nearly the same for all participants. Figure 7.1 shows the structure of the usability lab that we used.

![Diagram of Usability Lab](image)

Figure 7.1: The Usability Laboratory at the Department of Computer and Information Science, used in the usability tests described.

To provide a consistent test environment, a list-based view was included in the existing VisMeB framework. It was implemented as an additional visualization that could be started after entering the query terms. Consequently it was possible to decide if the list or the LevelTable visualization would be used to display the search results. The list-based view was implemented very similarly to the well known Google result-set presentation. Every page displayed ten result hits, ordered by relevance. Additional attributes like size, language and server type were shown, based on the meta-data presented in the LevelTable, to provide equal conditions. Opening a document could be achieved by a simple click that opened a new window to display the content. In the case of the LevelTable this feature was implemented by the additional BrowserView for viewing the content of the document.
7.7 POSTTEST

7.6.5 Training

A short training period was used to explain the LevelTable idea to the experimental group. Therefore a small video was presented that introduced the available features. No hints concerning the tasks to be solved were given in order to avoid a distortion of the evaluation results.

7.7 Posttest

After the maintest, the users were asked to fill out a posttest questionnaire to report their subjective opinion of the LevelTable. As a consequence this questionnaire was only handed out to the experimental group, because the control group was not confronted with the LevelTable visualization. Test 1, which was intended to investigate differences between the control and the experimental group, offered the chance for the experimental group to draw a comparison of the two visualizations used. This way, the subjective impressions of the participants, as well as the advantages or disadvantages of the single visualizations, became visible.

7.8 Test Results

To compare the test results SPSS 11.5 was used. A Student’s T-Test for the baseline test as well as a ONEWAY ANOVA for the second, main test found their way into the evaluation. Results are displayed in the form of barcharts (to demonstrate the mean value of the total time for fulfilling the tasks), boxplots (to consider the distribution of single test users and detect outliers), and tables. In combination, they deliver a good overall impression of the complete test results.

7.8.1 Test1 - Baseline Test

As a first step, the differences between the control and the experimental group should be analyzed. A Student’s T-Test was used for the results of this test. This is meant to compare the average time the two groups took to complete all tasks. The mean time for the control group to solve the ten baseline test questions was 318 seconds. In contrast, the experimental group needed 382 seconds for the same problem formulation (see Figure 7.2).

When executing an ANOVA using a given confidence interval of 95%, no significant difference could be discovered. The exact results can be seen in Table 7.1.

To investigate the results for outliers a boxplot was used for visualizing. As can be seen in Figure 7.3, two extreme outliers could be detected.
Furthermore, five outliers in total were detected: test persons 9 and 12 of the experimental group and test persons 4, 14, and 17 of the control group. By removing the corresponding values from the result computation, the mean values of total time are brought almost level. For the experimental group, it was a reduction of 40 seconds in the mean time, which leads to a value of 342 seconds. The value for the control group did not differ so markedly; only a reduction of 11 seconds could be recognized, leading to a mean value of 305 seconds. Just as before no significant difference between the two groups could be identified. Statistically, both groups required the same amount of time (see Figure 7.4 and Table 7.2).

Table 7.2: ANOVA for total time within the baseline test, excluding the outliers

<table>
<thead>
<tr>
<th></th>
<th>SUM OF SQUARES</th>
<th>DF</th>
<th>MEAN SQUARE</th>
<th>F</th>
<th>SIGNIFICANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETWEEN GROUPS</td>
<td>8138.359</td>
<td>1</td>
<td>8138.359</td>
<td>2.468</td>
<td>0.13</td>
</tr>
<tr>
<td>WITHIN GROUPS</td>
<td>75844.755</td>
<td>23</td>
<td>3297.598</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>83983.114</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.8 Test Results

The results of the baseline test confirm a non-significant difference between the control and experimental group. The assignments of test persons to the two groups could therefore stay the same for Test 2, the real main test.
7.8.2 Test2 - Main Test

In the next step, a ONEWAY ANOVA was employed to compare the group results of Test 2. A comparison of both, i.e. the full amount of time it took each group to solve all tasks and the average amount of time spent on each task, was executed. It should be mentioned that those tasks that were completely correctly fulfilled were included in the evaluation. In this case the missing values were replaced by the mean value of remaining group members. As a result, small distortions can arise but to compute a total time this approach was unavoidable.

7.8.2.1 Results concerning the total time of task completion

Figure 7.5: Comparison of the mean values for the total time for the main test

Concerning the time the groups required to finish all tasks, the two are almost the same. For the control group, a value of 591 seconds was computed, whereas the experimental group needed 598 seconds as displayed in Figure 7.5. Obviously the ANOVA (see Table 7.5) shows no significant difference. This would mean that the null hypothesis can be confirmed.

Table 7.3: ANOVA for total time within the main test

<table>
<thead>
<tr>
<th></th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Groups</td>
<td>368.757</td>
<td>1</td>
<td>368.757</td>
<td>0.008</td>
<td>0.931</td>
</tr>
<tr>
<td>Within Groups</td>
<td>1344691.152</td>
<td>28</td>
<td>48024.684</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>1345059.909</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Nevertheless, the standard deviation in the experimental group is very high (272 seconds compared to 115 seconds in the control group), which is an indicator for outliers. By means of a boxplot (see Figure 7.6) it is possible to make these variances visible and eliminate them. All in all, six outliers could be detected, four within the experimental group (one positive - test person 5, and 3 negative - test persons 9, 12, and 13) and two within the control group (one positive - test person 17, and one negative - test person 29) that almost balance. This can be seen in the very small difference in total time before (591 seconds) and after (573 seconds) removing the outliers.

![Boxplot showing the total time for the main test](image)

**Figure 7.6: Boxplot showing the total time for the main test**

Without the outliers the results change: now, the experimental group is clearly faster than the control group. The values are 455 seconds vs. 573 seconds, respectively, as can be seen in Figure 7.7.

In addition, the ANOVA (Table 7.4) reveals a significant difference between the groups: the LevelTable visualization is considerably faster than the list-based one. Taking a closer look at the three most obvious extremes (test persons 9, 12, 13), it was discovered that two of them (test persons 9 and 12) had already appeared as exceptions in Test 1, which supported the decision to exclude them. The extremely small value of significance of 0.002 lies far below the level of significance of 0.05, which demonstrates a really significant difference between the experimental and control group.

The use of a completely unknown visualization like the LevelTable almost certainly explains the appearance of more outliers within the experimental group than were found before. A detailed look at the different types of tasks allows us to find the specific types for which the LevelTable has an advantage over the list visualization.
Table 7.4: ANOVA for the total time within the main test, excluding outliers

<table>
<thead>
<tr>
<th></th>
<th>SUM OF SQUARES</th>
<th>DF</th>
<th>MEAN SQUARE</th>
<th>F</th>
<th>SIGNIFICANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETWEEN GROUPS</td>
<td>83025.830</td>
<td>1</td>
<td>83025.830</td>
<td>12.998</td>
<td>0.002</td>
</tr>
<tr>
<td>WITHIN GROUPS</td>
<td>140530.064</td>
<td>22</td>
<td>6387.730</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>223555.894</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7.8.2.2 Results concerning the time for completion of single tasks

Very often systems can prove their strengths and weaknesses in specific task areas. Some are better suited than others. To test the possible advantages with regard to special application scenarios, the test questions were divided into three categories containing twelve questions in all. These different classes are search documents, compare documents, and compare documents’ content, as mentioned above. On the following pages, an overview of the results for each category is given to detect any advantages of one or other search-result presentation.

Question Type 1: Search Documents This category contained a total of five questions. As can be seen in Figure 7.8 the mean time for solving the questions with the LevelTable is definitively lower than the time for working with the list. The task completion times of 87 seconds and 125 seconds imply a significant difference between the two presentations, which can be confirmed by the ANOVA (Table 7.5).

A value of 0.008 implying a significance level of 0.05 proves the significance clearly. The search for outliers with the aid of a boxplot (Figure 7.9) proceeded without a result

Figure 7.7: Comparison of the mean values for the total time for the main test, excluding the outliers
7.8 Test Results

Figure 7.8: Comparison of the mean values for task type 1

Figure 7.9: Boxplot showing time values for task type 1

for either the first or the second group.

The table structure, and in particular the sort function, had a strong influence on this very positive result for the LevelTable. In this way, the time to search for specific metadata could be minimized. In spite of the positive effect of the sort function, it included a drawback. For a correct sorting order, the documents’ names have to be spelled correctly. A leading space can put a document right at the top for no apparent reason. This caused some irritation for the test users. They did not notice the leading space and thus were astonished when the document being searched for was not found at its normal (and, by lexicographic sorting, expected) position. Another drawback was in a missing history function. Renewed access to previously opened documents in the list presentation could definitely be carried out faster. All in all, this type of task caused the least problems.
**Table 7.5: ANOVA for task type 1**

<table>
<thead>
<tr>
<th></th>
<th>SUM OF SQUARES</th>
<th>DF</th>
<th>MEAN SQUARE</th>
<th>F</th>
<th>SIGNIFICANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETWEEN GROUPS</td>
<td>10482.824</td>
<td>1</td>
<td>10482.824</td>
<td>8.262</td>
<td>0.008</td>
</tr>
<tr>
<td>WITHIN GROUPS</td>
<td>35525.262</td>
<td>28</td>
<td>1268.759</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>46008.086</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Question Type 2: Compare Documents**  In contrast to the first category, the type 2 question provided a different result. The four questions were answered in a mean time of 134 seconds within the experimental group whereas the control group fulfilled the tasks a little faster in 122 seconds (see Figure 7.10).

![Figure 7.10: Comparison of the mean values for task type 2](image)

Nevertheless, no significant difference could be detected, which is visible in the ANOVA in Table 7.6. A value of 0.437 is obviously higher than the given significance level of 0.05 for a confidence interval of 95%.

**Table 7.6: ANOVA for task type 2**

<table>
<thead>
<tr>
<th></th>
<th>SUM OF SQUARES</th>
<th>DF</th>
<th>MEAN SQUARE</th>
<th>F</th>
<th>SIGNIFICANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETWEEN GROUPS</td>
<td>1214.333</td>
<td>1</td>
<td>1214.333</td>
<td>0.621</td>
<td>0.437</td>
</tr>
<tr>
<td>WITHIN GROUPS</td>
<td>54769.391</td>
<td>28</td>
<td>1956.050</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>55983.724</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Again, no outliers could be detected from the respective boxplot Figure 7.11. When a given document set had to be compared it had first to be counted, which had to be done repeatedly in the majority of cases. Although a sort function was implemented in the LevelTable, no filter function was available. This could be one reason for the comparatively poor results for the LevelTable visualization. In total, no group had problems in
solving the tasks, but sometimes wrong answers arose. An absence of numbering lead to complications when a specific number of documents had to be counted. In contrast, the list always presented ten documents per page, which made the counting of documents very easy. Another drawback was the lack of a provision for multiple sorting. When users expected this feature they chose an incorrect document. If no counting was required, the LevelTable enabled a significantly faster solution of the task than the list did.

**Question Type 3: Compare Documents’ Content** So far, no outliers were detected although the presentation of the mean value for total time of task completion included more than one. This indicates a cumulative appearance of outliers in this kind of question category. If we consider the mean time to complete question type 3, no distinctive feature becomes visible. The experimental group needed 377 seconds to solve the tasks in this third category, the control group only 345 seconds (see Figure 7.12).
Even though the control group could answer the questions a bit faster, the ANOVA (Table 7.7) provides no significant difference. The value of 0.62 compared to the significance level of 0.05 allows this conclusion to be drawn.

**Table 7.7: ANOVA for task type 3**

<table>
<thead>
<tr>
<th></th>
<th>SUM OF SQUARES</th>
<th>DF</th>
<th>MEAN SQUARE</th>
<th>F</th>
<th>SIGNIFICANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BETWEEN GROUPS</td>
<td>7524.077</td>
<td>1</td>
<td>7524.077</td>
<td>0.251</td>
<td>0.62</td>
</tr>
<tr>
<td>WITHIN GROUPS</td>
<td>838064.773</td>
<td>28</td>
<td>29930.885</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>845588.85</td>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To investigate if outliers are present, a standard deviation has to be done. It provides a result of a 212 seconds deviation for the experimental group whereas only 98 seconds for the control group could be measured. In the present case, the boxplot (Figure 7.13) presents obvious deviations shown by the cumulative occurrence of outliers.

![Boxplot showing the time values for task type 3](image)

**Figure 7.13: Boxplot showing the time values for task type 3**

If all these outliers are removed from the evaluation, a completely different picture appears. In this case the experimental group is able to answer the questions in less time than the control group. Figure 7.14 displays the corresponding values of 261 and 321 seconds, respectively.

The ANOVA (Table 7.8) confirms the significant difference recognizable in the value of 0.012 for a significance level of 0.05.

All in all, most errors occurred in this type of task. The lack of expert knowledge of the task domain was probably one cause of the bad result. Even if users found the correct answer, they were not sure if this answer was the correct one. Reading a complete document was often done only reluctantly. Noticeable is the fact that more participants from the control group aborted reading the complete document than participants from the experimental group, who always tried to find a solution. A higher motivation to find
7.8 Test Results

Figure 7.14: Comparison of the mean values for task type 3, excluding outliers

Table 7.8: ANOVA for task type 3, excluding outliers

<table>
<thead>
<tr>
<th></th>
<th>Sum of Squares</th>
<th>DF</th>
<th>Mean Square</th>
<th>F</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Groups</td>
<td>21918.844</td>
<td>1</td>
<td>21918.844</td>
<td>7.476</td>
<td>0.012</td>
</tr>
<tr>
<td>Within Groups</td>
<td>64503.239</td>
<td>22</td>
<td>2931.965</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>86422.083</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The solution was noticeable, but a connection between the kind of visualization and the respective motivation can only be investigated by another test focusing on this topic.

7.8.3 Posttest Results

The purpose of the posttest questionnaire was to investigate the subjective impressions of the experimental group members. The design as well as the usability was rated with the aid of a seven-point Likert scale. The layout of the graphical user interface was rated as understandable (mean value of 5.61 where 1 = layout is very incomprehensible and 7 = layout is clear and understandable). The colors used were also rated positively with a mean value of 6.11, where 1 = color usage is very unpleasant and 7 = color usage is very pleasant. The navigation was also rated quite highly and reached a mean value of 5.11, where 1 = navigation is definitely not intuitive and 7 = navigation is definitely intuitive. If the participants were asked if they felt lost, the answers varied. The mean value was 4.0 (where 1 = I never felt lost and 7 = I always felt lost), the standard deviation being 1.9. Further questions asked if the participants would like to work with the LevelTable on a daily basis. The advantages of the LevelTable in contrast to e.g. Google were asked about and nearly all test users stated that they could imagine specific situations in which a list presentation like the one known from Google was inferior to the LevelTable. Fourteen out of eighteen people who tested the LevelTable could imagine using it on a daily basis.
Detailed information on the posttest results can be read in [Ger03].

7.9 Summary

This chapter described the comparison of the LevelTable, which is integrated in the VisMeB framework and a simple list such as is familiar from established websites like e.g. Google. In summary, the LevelTable does not have to be hidden away! The results are very positive and encourage the further development of the ideas presented.

As regards the single tasks, the results are as follows: the experimental group completed six of the twelve assignments significantly faster, two of them (including extremes: three) significantly slower, and concerning four tasks (including extremes: three) the two groups were level. Three categories of problems caused by the LevelTable visualization were discovered. One difficulty resulted from the subject matter, a geographical database, which was almost unknown to the participants; hence reading a document and extracting information was complicated. Another key problem was caused by VisMeB, which serves to show the documents. In the beginning, not all users were able to work out how the LevelTable and the browser were connected. The third, and bigger problem was that the documents in VisMeB are not numbered. This was especially problematic when users were asked to analyze a specific number of documents, e.g. the file size of the first ten documents. Another problem occurred in coordination with the BrowserView used. The mouse position determines the particular document that will be shown in the BrowserView. If the mouse moves over another document, the display changes immediately. To fix a document in the BrowserView, the context menu and the menu item "Add To BrowserView" had to be used. Without this technique, errors were made because users thought that they were viewing a different document than was, in fact, the case.

Altogether, the evaluation of user satisfaction was a surprise. All participants could envisage a working situation in which VisMeB would be superior to a list presentation. The majority of the eighteen experimental group members could envisage using it on a daily basis. Since none of them had had any prior knowledge of VisMeB, this result is quite astonishing.

This leads directly to the next chapter, which describes further ideas that could be incorporated and tests that should be done. Test already conducted, as well as various discussions, lead to the points outlined in Chapter 8.
Although the models performed well during evaluation, there is still some work to do. Further evaluations have to be done, in particular to confirm that the combined use of multiple coordinated visualizations and the granularity concept really improves the information-seeking process. A study to investigate user behavior is currently being done within the DROID (Dynamic Remote Operation Incident Detection) project \(^1\). It is part of the VisMeB system and logs user actions in the MedioVis version. In fact, MedioVis is installed on several computers in the University library at Konstanz for test purposes. Thus, we are able to observe typical users during their daily work i.e. when they are not under pressure of participating in a lab based experiment. Some design aspects that are mentioned are implemented in a way that leads to discussions. One example refers to the zoom function available in the Scatterplot. In this view it acts like a filter, because all data points that are not positioned in the zoom area disappear. In contrast, the SuperTable stays unchanged. This behavior was implemented as a design decision (to keep the overview in the SuperTable) but nevertheless it contravenes the Rule of Consistency, because it does not “make the states of multiple views consistent”. Another consideration concerns the best location for the Scatterplot. \textit{Should it be placed on top of the SuperTable because of its overview function, or is it better to leave it below because the detailed information is shown in the table and is probably used more often?} All these questions can only be answered by further user tests.

Two further ideas that could improve the concepts presented here are proposed and will be investigated more thoroughly in the following two sections.

### 8.1 Similarity

A broadened extension of the semantic similarity implemented within the DocumentUniverse could also be implemented within SuperTable or ScatterPlot. As an illustration, take the following situation: Tim Herzog (the architect from Scenario 1, Section 1.1) has

\(^1\text{http://hci.uni-konstanz.de/index.php?a=research\&b=projects\&c=1810659\&lang=en}\)
found an interesting data set and wants to know if there are similar data sets that conform to his requirements. Opening the context menu could provide the function "Find Similar" which would initiate the following process:

1. In the SuperTable the selected data set moves to the top of the table and all similar data sets follow, ordered by the degree of similarity. These values are already computed by the LSI algorithm used in the DocumentUniverse and stored in a single matrix row with respect to the selected data set. Another possible arrangement could center the start document and arrange all similar ones above and below (e.g. the most similar one above, the second most similar one a single row below, the third one two rows above and so on). This idea was already discussed but discarded. The inconvenient arrangement could confuse the user and the mode of reading (one row up, two rows down, three rows up, four rows down, and so on) would be made unnecessarily difficult.

2. In the Scatterplot, different variants are conceivable. The initial situation is displayed in Figure 8.1. A set of 17 items is visible; one data point in the upper right corner is selected, marked by the blue fill color. Possible effects of the similarity function are:

   (a) A similarity slider appears to define a threshold value (see Figure 8.2(a)). The data points that are not defined as similar (by falling below the threshold value, which should stay in the interval [0,1) for normalized values) can be dimmed or completely faded out to emphasize the similar, and hence currently important, data points. Moving the mouse over a specific point causes the similarity value to appear, or labels can be set if the display is still in a readable mode. In this way the context remains constant. Direct manipulation by the threshold slider allows the user to find a meaningful boundary very quickly, depending on the corresponding application scenario.

   (b) In a similar way to the SuperTable scenario, the data points can arrange themselves around the selected point in concentric circles (see Figure 8.2(a)). The radius depends on the value of similarity, the exact position is defined by the number of dots on the same circle and the neighborhood. Dissimilar points have to be dimmed out to emphasize the focussed ones and avoid overlapping. To gain more space for the concentric circles, the selected data point should move to the center unless enough space is available around the current position. If necessary, a logarithmic placement can be used. The advantage of this version is the directly visible similarity value, while a disadvantage is the lost context, i.e. the axis labeling is invalid.

Both approaches offer advantages and disadvantages. The first one retains the context, but similarity is only visible by added labels or tooltips.
8.1 Similarity

The second approach is attractive because of the directly visible presentation of similarity. Nevertheless, the context is lost, which in turn increases the cognitive load for the user.

![Figure 8.1: The initial situation for the semantic similarity scenario with the selected data point at the upper right corner](image)

![Figure 8.2: Semantic similarity visualized by dimming out non-similar data points (left side) or concentric circles (right side)](image)

However, if one of these semantic similarity implementations is used, different kinds of similarity should be defined:

1. Similarity concerning the complete data set
2. Similarity concerning a single metadata, i.e. only a part of the data set

Version 1. obtains its values as mentioned above - from the previously computed LSI matrix. For the second version a new similarity function that calculates similarity with respect to a specific metadata has to be defined. These produce additional costs but can
definitely offer strong support to the user. E.g. if a site planner is only interested in a specific area, he does not want to get data sets in which all the metadata (e.g. resolution, price, format, a.s.o.) except for location is the same. Therefore, it is meaningful to provide this additional possibility.

8.2 Dimension of Interest

Further ideas to improve the VisMeB approach concern the introduction of a *DimOI*- the Dimension of Interest. The introduction of the DimOI can be seen as a three-dimensional extension of the representation matrix introduced above (see Section 4.4). Working with a database enables the user to create different views on the data. An example will clarify the approach. Recall Scenario 3, which introduced the media science student Daniel Beck, who is looking for information concerning the topic of his homework, an essay about Charlie Chaplin and his most famous movies. Let us assume he has already found the movies he was searching for. While he scans the content belonging to ”The Great Dictator” the name Jack Oakie catches his eye. He has already heard this name before, but he does not remember in which context. To get more information about Mr. Oakie he clicks on the name in the table. This describes the moment when the dimension of interest is changed. So far Daniel was interested in movies, especially in those concerning Charlie Chaplin. Now he switches his interest from this topic to a single person, in this case Jack Oakie. The dimension ”movie” has switched to the dimension ”person”. Hence, it is no longer the movies and information about them like language, type of media, or signature, that are in the foreground, but the person Jack Oakie and information about him like age, nationality, curriculum vitae and movies he was involved in (which seems to be similar to the first query concerning Charlie Chaplin but is not). The focus is set on the person himself, not on the movies he took part in, although this information is provided. In this case, Daniel finds out that Jack Oakie was responsible for the special effects in the movie *Devotion* from 1946.

How the information will be displayed is a further topic that has to be considered. One design variant proposes a new window that appears with all the information concerning the respective person. It can be implemented as a window that can change its size, be moved around, and closed if necessary. Although this provides a lot of freedom because other information is still visible in the background, it contains a drawback - the structure of the table is abandoned.

A second variant could include the new data in the same table or, more precisely, in the same cell. The structure is unchanged; no additional windows appear. To emphasize the fact of leaving the dimension, the cell could be displayed as a kind of Post-it note, clarified by a 3D effect. In this way, the impression could be conveyed of pulling something out of the cell. To provide enough space for the information, it is necessary to enable a magnification (and a reduction, respectively) of the cell’s size.
Both variants have their advantages and disadvantages, so a user test should be planned to shed light on this topic.
Figure 8.3 illustrates the idea acting on the previously introduced representation matrix.
The problem of finding important information in an effective, efficient and user-satisfying manner is still a significant one. Visual information-seeking systems try to provide appropriate support by reducing the cognitive load for the user when confronted by thousands of information chunks. The use of visualizations is more and more on the increase because of their advantages over pure text presentations. Although typical information-seeking system users are familiar with textual results, tests proved the advanced efficiency and effectiveness of applications using visualizations.

The great variety of techniques and systems from the field of information visualization lead to a very impressive demonstration of the developments in this domain. In Chapter 2, techniques, used in the VisMeB system, are presented as well as visualizations that had a profound influence on the development. The information-seeking mantra "Overview first, zoom and filter, then details on demand" has a marked effect on the choice of visualizations, e.g. by choosing a scatterplot to provide a complete overview of the entire data set. Multiple coordinated views, their use and their effects, played a very important role in this work. The allocation of different views to explore the data from different perspectives, as well as the possibility of choosing the specific visualization that is a better fit for the given situation or is preferred by one user group, provide one of the main advantage for multiple coordinated views. Various views are used in the VisMeB approach to support the typical user while working in the field of information seeking.

The free adaption of application domains supported by the Visual Configurator provides another advantage of the system. Although the visualizations are tailored to specific scenarios, the independent implementation makes it easy to adapt the application to any arbitrary domain. Meta-data can be assigned to visualizations in a way that allows a user to determine the layout that fits his own needs. Pre-settings always have to be performed (e.g. by an administrator), nevertheless adjustments are easy to execute.

Meta-data, as one main basis for a concrete and efficient search of important information, is gaining more and more influence. The preparation of the available data reaches a significance almost equal to that of the data basis itself. Reducing the mass of information to manageable chunks was another aim this thesis followed, and was brought about the granularity concept. A semantic zoom into the data of interest provides an efficient way to separate useful from useless content. The more a user is interested in data that seems
to be important, the more details he can achieve. A combination of both concepts, *multiple coordinated views* and the *granularity concept*, leads to an enormous advantage over simple information-seeking systems like search engines available in the Web. Research results in the field of user behavior demonstrated that, on average, only the first 20 results (i.e. the first two result pages if a page displays ten results, which is common for current systems) are explored by a user. The problem of context switching (from one page to another or from the link list to the respective page) is reduced by the table presentation that was chosen for VisMeB.

A large variety of interaction techniques were implemented to make the framework easier to use. All of them are common and proven and therefore provide good preconditions for a smooth method of operation. Although there is always an effort in learning to work with new applications that differ from the ones that users are used to, the investment time is worth it, as the evaluations showed. This procedure made it possible to remove or minimize difficulties in the kind of interaction, and use of the application during the design process. To decrease the learning effect and further usability problems, many user tests were conducted during the development of VisMeB, from the beginning up to the final release, directly following the "user-centered design process". Prototypes were built in the form of paper-based or html mockups as well as fully developed Java prototypes. Each development stage was accompanied by a user test that helped to find failures, undetected by expert users or developers because of their strong relation to, and involvement in, the design process.

The last user test that compared a simple list presentation (as known from Google) and one version of the SuperTable (one main visualization of VisMeB) justified the hopes that existed when starting the development:

*The VisMeB framework supports the user in an effective, efficient, and user-satisfying manner during the information-seeking process.*
List of Figures

1.1 Main structure of the thesis .................................................. 6

2.1 Visualization Reference Model [CM99] ..................................... 11

2.2 A cutout of the ISO 19115 standard for geoinformation. A full view is attached on the corresponding CD. ........................................ 15

2.3 Dynamic Homefinder showing houses in the area of Washington D.C. [WS92] ................................................................. 16

2.4 Brushing and Linking in a scatterplot matrix [Lub62] ..................... 17

2.5 Movable Filter by Fishkin and Stone [FS95] .................................. 18

2.6 Overview plus detail view for the North Bend Rail Trail State Park, WV (see: http://www.wvstateparks.com/northbendrailtrail/index.html) .... 19

2.7 The principle of the Bifocal Display [SA82] ................................. 19

2.8 The Perspective Wall as an example of a bifocal display; developed by Inxight Software Inc. [MRC91] .............................................. 20

2.9 Panning describes the smooth movement of a viewing frame over an image (left), whereas zooming is a magnification of a decreasing fraction of an image or vice versa (adapted from [Spe01a]) ........................... 21

2.10 Panning and Zooming in SYNTH - A Gamma-Ray Spectrum Synthesizer. Moving around the blue rectangle in the upper right corner (panning) can be used to explore small areas of the overview, shown in the gray colored overview rectangle. A larger or smaller area can be investigated by changing the blue rectangle’s size (zooming) [Lab02] ..................... 21

2.11 Visualization of telephone traffic in the USA using SeeNet [BEW95] ... 24

2.12 Spotfire showing the deposits of heavy metals in Sweden [Ahl96] ...... 25

2.13 OpenDX displaying the initial purchase intent in the US within a map [Res04] ................................................................. 26

2.14 FilmFinder showing movies with Sean Connery [AS94b] ................. 26

2.15 The Interactive Timeline Viewer presenting an overview of events in Miguel de Cervantes’ life (background) and a pop-up window with detailed information about a specific event (foreground, yellow color) [MFM03] .... 27

2.16 Envision system for visualizing the content of a digital library [FHN+93] 28
2.17 Search Result Explorer from the XFind system plotting search results along two axes [ASL+01]. ................................................................. 29
2.18 The Polaris user interface [CSH02]. .................................................. 30
2.19 The ClusCorr98 system displaying an extract of a scatterplot matrix of cluster memberships [HJMB02]. ....................................................... 31
2.20 Scatterplot Matrix developed at the VRVis research center in Vienna, Austria [Voi02]. ................................................................. 32
2.21 IBM Open Visualization Data Explorer presenting data from a credit card application [Res04] ................................................................. 34
2.22 Three-keyword axes display of the NIRVE system mapping the keyword strength to the axes [CLS00] ....................................................... 35
2.23 Webwinds XYZPlot, developed at the NASA’s Jet Propulsion Laboratory [Lab05] ................................................................. 36
2.24 Voxelplot developed at VRVis research center, combining scientific and information visualization [Sah02] ....................................................... 37
2.25 A Galaxy Of News displaying keywords, article headlines and partially readable articles in a three-dimensional space [Ren94] ...................... 42
2.26 IN-SPIRE system (left) and SPIRE (right) displaying a galaxy [Lab04] ... 43
2.27 Visualization Islands included in the xFIND system [ASL+01] ................................................................. 43
2.28 InfoSky system displaying documents and using polygons to emphasize clusters [AKSG04] ................................................................. 44
2.29 Themescape displayed within the SPIRE system [Lab04] ............ 45
2.30 Thematic Landscape within the Bead system, constructed from articles of an HCI conference, CHI’91 [Cha93] ....................................................... 46
2.31 Cartia Themescape [Car04] ................................................................. 47
2.32 Kohonen map displaying 32,627 articles from a newsgroup collection [Koh99] ................................................................. 48
2.33 Infozoom system from humanIT Software GmbH [SBB96] ............ 49
2.34 TableLens with data from the Superbowl 2001 [RC94] ............ 50
2.35 DateLens using the semantic zoom to magnify a single day in a monthly view [BCCR04] ................................................................. 51
2.36 Silver2 displaying two lenses at the beginning and the end of a shot [LMC+04] ................................................................. 52
2.37 Sockeye system showing all genes on a 151MB chromosome (a) and the individual genes (b) in a 200Kb region around one of the four histogram peaks in the image to the left. As the new region is shorter than the 300Kb threshold for genes, Sockeye automatically queries in and shows individual genes [MAB+04] ................................................................. 53
2.38 TrendDisplay presenting a timeline with different levels of detail [BG03] ................................................................. 54
2.39 The DENIM system supporting web site designers by providing different levels of refinement as can be seen on the left border [LNHL00] ............ 54
2.40 CareVis application window showing the execution of a plan [AM04] ................................................................. 56
2.41 The combined 2D and 3D scatterplot with user interface, implemented in the VRVis project [HPR04] ................................. 57
2.42 Interface of the microarray time-series visualization tool [CK03] ........ 58
2.43 TimeSearcher visualizing time-series for genetic research [SBZ'03] ... 59
2.44 The Hierarchical Cluster Explorer [SS02] .......................... 59
2.45 The Logos Bible Software providing a hierarchical structure for navigating on the left, a bible translation on the top right, and the corresponding commentary on the bottom right [Kos05] ...................... 60
2.46 Corel’s WordPerfect displaying a formatted text (top) and the corresponding formatting code (bottom) [Cor05] ....................... 61
2.47 DEVise system displaying two scatterplots that can be navigated synchronously [WL00] .................................................. 62
2.48 CVS Activity Viewer of the Augur system, visualizing differences by the use of light and dark blue color [FD04] ....................... 62
2.49 The Visible Human Explorer using a combined textual and visual approach [NSP97] .................................................. 63
2.50 Generic Genome Browser displaying the detailed view after zooming out to 200 kb, showing semantic zooming [SMS'02] ............... 64
2.51 Simultaneous Menus to improve the task-performance speed [HS00] ... 65
3.1 Snap components: First, the relations are loaded into visualizations, then snapped together (adapted from [Nor00]) .......................... 81
3.2 The View Coordination Model Architecture (adapted from [PP01]) .... 82
3.3 Abstract model for coordination in exploratory visualization (adapted from [BR03]) .................................................. 83
3.4 Coordination model for coordination of multiple coordinated views in the VisMeB framework ........................................ 86
3.5 VisMeB Architecture .................................................. 87
3.6 INSYDER visualizations integrated in VisMeB: The BarCharts (left) and the RelevanceCurve (right) ................................. 88
3.7 INSYDER visualizations integrated in VisMeB: The SegmentView, implemented with StackedColumns (left) and TileBars (right) .......... 89
3.8 INSYDER visualizations integrated in VisMeB: The ScatterPlot (left) and the BrowserView (right) ........................................ 90
3.9 Overview of the development history of the MediaGrid .................. 92
3.10 The four levels of the LevelTable ................................... 93
3.11 The six levels of the GranularityTable .................................. 95
3.12 GridTable version of the SuperTable, implemented within the MedioVis project .......................................................... 96
3.13 MovieVis first view after entering the query term “james” ................ 98
3.14 The level slider is set to the extreme right and thus all rows are increased to the maximum. By moving a mouse over a term (e.g. “Sean Connery” in the third column) and clicking it, a new query can be started. .... 99
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.15</td>
<td>As a result of the new query, only movies with Sean Connery are displayed. Although there are still James Bond movies, additional ones are added.</td>
</tr>
<tr>
<td>3.16</td>
<td>The result set of MovieVis can be restricted by using a table filter. In the current situation the title is filtered by the term &quot;Bond&quot;.</td>
</tr>
<tr>
<td>3.17</td>
<td>The slider located above the table enables a stepless magnification of all rows simultaneously. Thus, the rows’ height is completely dependent on the user and his slider settings.</td>
</tr>
<tr>
<td>3.18</td>
<td>A single row is enlarged to the maximum, while all other rows stay in the lowest level, displaying a single line of text.</td>
</tr>
<tr>
<td>3.19</td>
<td>The evolution of the MediaGrid: the origin lay in the INSYDER system; a first advance was the LevelTable, followed by the GranularityTable, and resulting in the MediaGrid.</td>
</tr>
<tr>
<td>4.1</td>
<td>Windows Explorer displaying a Detail-only view.</td>
</tr>
<tr>
<td>4.2</td>
<td>Variations of zoom and replace: a) zoom only, b) zoom and the possibility of scrolling, c) zoom with additional levels of magnification (adapted from [PCS95]).</td>
</tr>
<tr>
<td>4.3</td>
<td>Overview and detail implemented in the SVG Browser Squiggle.</td>
</tr>
<tr>
<td>4.4</td>
<td>Overview and detail realized within a tiled multilevel view.</td>
</tr>
<tr>
<td>4.5</td>
<td>London Underground Map using a Bifocal Display.</td>
</tr>
<tr>
<td>4.6</td>
<td>Map of Washington, D.C., using a Fisheye View.</td>
</tr>
<tr>
<td>4.7</td>
<td>Visualization of a file system, using translucent layers in the Macroscope system.</td>
</tr>
<tr>
<td>4.8</td>
<td>Semantic Zooming for the MediaGrid Scenario.</td>
</tr>
<tr>
<td>4.9</td>
<td>Schematic presentation of semantic zooming. The bottom slices ((1)-(3) and (a)-(d)) show views at different points [BHP+96].</td>
</tr>
<tr>
<td>4.10</td>
<td>LevelTable displaying all possible levels. In level 1 a mouse-over effect causes a kind of preview implemented by a row enlargement.</td>
</tr>
<tr>
<td>4.11</td>
<td>GranularityTable displaying all possible levels. Again, a mouse-over effect causes an enlargement of a row to display more information.</td>
</tr>
<tr>
<td>4.12</td>
<td>The GranularityTable is able to display single rows in different levels of detail. This enables the user to compare several data sets using the same level without moving the entire data set to this level and wasting space (data sets in level 1 need less space than in level 4, for instance).</td>
</tr>
<tr>
<td>4.13</td>
<td>The four levels of detail realized within an early MediaGrid version: (1) Only titles are visible, (2) additional information such as tagline or rating are displayed, (3) the movie poster appears, (4) the trailer can be played.</td>
</tr>
<tr>
<td>4.14</td>
<td>Any object in the Visualization &amp; Meta-data Pool can be assigned to an arbitrary cell in the table. The sizes of the object itself and of the surrounding objects in the same row are not restricted.</td>
</tr>
<tr>
<td>5.1</td>
<td>Taxonomy of multiple view coordination [NS97].</td>
</tr>
</tbody>
</table>
5.2 LevelTable Level 1 with ScatterPlot ........................................ 129
5.3 LevelTable Level 3 with BrowserView. One segment with a maximum spike is highlighted. .................................................. 130
5.4 LevelTable Level 4 with BrowserView ........................................ 131
5.5 Interaction via a pop-up menu in the ScatterPlot. The item of interest can be found in the ScatterPlot and explored for further details by using the slider to change the level ........................................ 133
5.6 GranularityTable levels 1 to 6 ..................................................... 134
5.7 Document Universe displaying semantic similarity. The selected points are colored blue whereas the focussed one uses a red highlighting. .......... 135
5.8 MediaGrid and CellZoom .......................................................... 136
5.9 The MediaGrid of MedioVis in combination with the ScatterPlot. Original locations are changed; the ScatterPlot is normally positioned in the upper part of the window. A flexible assignment technique allows the movement of single visualization windows. ........................................ 137
5.10 MedioVis with BrowserView and LocationMap .............................. 138
5.11 MedioVis displaying the TitleView (on the left) and the list of selected items (on the right). ....................................................... 139
5.12 The information-seeking steps implemented within the MedioVis framework. Before entering any stage, the preceding ones must have been executed at least once. ....................................................... 140
5.13 Filter Dialog Box ................................................................. 141
5.14 Circle Segment View used as Filter ............................................. 142
5.15 Two MovableFilters connected by a boolean ’AND’. Blue-colored rows in the table arise from the meta-data ”country” and the characteristic ”Italy”, yellow ones from ”theme_code” ”Infrastructure”, and grey ones from a combination of both. Only points underneath both filters get this combined color ......................................................... 143
5.16 LevelTable and ScatterPlot displaying the results of a query in the movie database, containing the terms ”fantasy”, ”king”, and ”ring”. Three dots are selected in the upper right corner, marked by a dark filling color .... 146
5.17 Level 2 of the LevelTable, sorted by selection to see the three selected movies on top. Only two of them include the query term ”king”, clarified by the red bar in the third column. ........................................ 147
5.18 Level 4 of the LevelTable, including the SegmentView in its stacked-column version for finding query terms in the description that is displayed in the BrowserView in the lower part of the window ............ 148
5.19 Level 4 of the LevelTable highlighting a segment in the BrowserView (lower part) containing the term ”king” ......................................... 149

6.1 Relationships between Interaction Design and academic disciplines, design practices, and interdisciplinary fields .................................................. 155
6.2 Timeline of development stages during the user-centered design process .... 165
6.3 First paper mock-up of VisMeB to test the initial ideas within a user test. 165
6.4 HTML Mock-Up: The Scatterplot allows the level of document detail to be changed (left), a scrollable abstract in the GranularityTable provides further information (right) 167
6.5 SegmentView in StackedColumn form at the top (each bar is a text segment, relevance is shown by size and colour) and in TileBar form at the bottom (each line is a text segment, relevance is shown by saturation. Navigation in text is by the upper left arrow) 168
6.6 VisMeB level 2 supporting the recognition effect for keywords by color in column three (geo in green), four (information in red), and five (system in purple) 172
6.7 GridTable implemented in the MedioVis project to present information to a library user 173
6.8 The MediaGrid of the MedioVis project. A zoom to single cells is possible, resulting in e.g. the display of a movie poster (left column), a trailer (center), or the location map (right column) 174

7.1 The Usability Laboratory at the Department of Computer and Information Science, used in the usability tests described 182
7.2 Comparison of the mean values for total time at the baseline test 184
7.3 Boxplot showing the total time for the baseline test 185
7.4 Comparison of the mean values for total time for the baseline test, excluding outliers 185
7.5 Comparison of the mean values for the total time for the main test 186
7.6 Boxplot showing the total time for the main test 187
7.7 Comparison of the mean values for the total time for the main test, excluding the outliers 188
7.8 Comparison of the mean values for task type 1 189
7.9 Boxplot showing time values for task type 1 189
7.10 Comparison of the mean values for task type 2 190
7.11 Boxplot showing time values for task type 2 191
7.12 Comparison of the mean values for task type 3 191
7.13 Boxplot showing the time values for task type 3 192
7.14 Comparison of the mean values for task type 3, excluding outliers 193

8.1 The initial situation for the semantic similarity scenario with the selected data point at the upper right corner 197
8.2 Semantic similarity visualized by dimming out non-similar data points (left side) or concentric circles (right side) 197
8.3 Adding a third dimension to the representation matrix 199

A.1 The 2D Scatterplot using a movable filter supporting the search for Danish documents 219
A.2  Three-dimensional version of the Scatterplot 220
A.3  MultiDataPoint-View to explore overlapping data points 221
A.4  BrowserView to compare documents using a horizontal (a) and a vertical (b) alignment 221
A.5  DocumentUniverse displaying the semantic similarity between all data sets 222
A.6  LocationMap displaying the location of the focussed media 223
A.7  Visual Configurator showing the meta-data to be visualized on the left side and the assignment to the diverse visualizations on the right. The meta-data ”domain_category” is being dragged to the visualization window on the right. 224
List of Tables

2.1 Data Type by Task Taxonomy (TTT) to identify visualization data types [Shn98] ................................................................. 8
2.2 Tasks concerning the TTT [Shn98] ......................................................... 9
2.3 Another Way to Differentiate Between Visual Structures [Shn98] ....... 10
2.4 Explanation of terms used in the Visualization Reference Model [CM99] 11
2.5 Relative effectiveness of retinal properties. Q = Quantitative data, O = Ordinal data, N = Nominal data. A + indicates the property is good for that type of data, a 0 indicates a marginal effectiveness, and a - only a poor one [Mac95] . ........................................................... 12
2.6 The 15 elements of the DublinCore core meta-dataset, divided into categories ................................................................. 14
2.7 Interaction techniques used in the VisMeB framework ....................... 66
2.8 State-of-the-Art Scatterplot Applications ........................................ 67
2.9 State-of-the-Art Applications for Semantic Similarity Maps, Table-base Visualizations, and Semantic Zoom .......................... 68
2.10 State-of-the-Art applications using Multiple Coordinated Views ......... 69
3.1 Three different scenarios (5T1-5T3) and the corresponding requirements . 75
3.2 Advantages and disadvantages of SuperTable and Scatterplot .......... 78
3.3 Granularity Levels for the MedioVis Scenario .................................. 97
3.4 Advantages and disadvantages within the MediaGrid history ............. 104
4.1 Representation Matrix for Library Scenario ....................................... 116
5.1 Relationship between corresponding views realizing the select - select coordination .......................................................... 131
5.2 Relationship between corresponding views realizing the select - select coordination .......................................................... 134
5.3 Advantages and disadvantages of filters used ..................................... 144
7.1 ANOVA for total time within the baseline test ................................. 184
7.2 ANOVA for total time within the baseline test, excluding the outliers ... 184
7.3 ANOVA for total time within the main test ............................................. 186
7.4 ANOVA for the total time within the main test, excluding outliers .......... 188
7.5 ANOVA for task type 1 ................................................................. 190
7.6 ANOVA for task type 2 ................................................................. 190
7.7 ANOVA for task type 3 ................................................................. 192
7.8 ANOVA for task type 3, excluding outliers ......................................... 193

B.1 Mittelwerte der Likert-Skalen ......................................................... 230
B.2 Mean values for the Pre-Test questionnaire .......................................... 231
B.3 Gender .......................................................... 231
B.4 Education .......................................................... 231
B.5 Study / Education / Profession ..................................................... 231
B.6 PC Experience .......................................................... 231
B.7 Operating System used .......................................................... 232
B.8 Personal connection to Computer Experts ......................................... 232
B.9 Use of internet search engines ...................................................... 232
B.10 Internet search engine used most often ........................................... 232
B.11 Mean values for the post-test questionnaire ....................................... 236
This chapter includes a detailed description of a) the mathematical background for computing semantic similarity maps introduced in Chapter 2 and b) the visualizations of the VisMeB framework (Chapter 3), that are not presented in detail so far, but nevertheless relevant.

A.1 Semantic Similarity Maps

The first step will be to present the exact procedure for applying the LSI algorithm:

Let $t$ be the number of index terms and $d$ the number of documents. Thus, the term-document matrix can be described as a $t \times d$ matrix $M$ and can be decomposed into the product

$$M = TSD'$$

where $T$ and $D$ have orthonormal columns and $S$ is diagonal. $T$ is the matrix of the left singular vectors, i.e. the matrix of eigenvectors of the square symmetric matrix $MM'$ received from the term-term correlation matrix. $D$ is the matrix of right singular vectors, i.e. the matrix of eigenvectors of the transpose of the document-document matrix $M'M$, and $S$ is the diagonal matrix of singular values. The SVD could thus be described by decomposing the $t \times d$ matrix $M$ into the $t \times m$ matrix $T$, the $m \times m$ matrix $S$, and the $m \times d$ matrix $D$, where $t$ describes the number of rows (i.e. number of terms) of $M$, $d$ the number of columns (i.e. number of documents) of $M$, and $m$ the rank of $M$, where

$$m \leq \min(t, d)$$

To obtain a smaller matrix $M_s$ which is closest in the least square sense to the initial matrix $M$, the $s$ largest singular values of $S$ have to be kept, all others (i.e. the smaller ones) are set to zero, the corresponding columns of $T$ and $D$ are deleted. Thus, the reduced model can be presented as

$$M_s = T_sS_sD'_s$$

where $s, s < m$ describes the reduced space dimensionality, $T_s$ is a $t \times s$ matrix, $S_s$ an $s \times s$ matrix, and $D'_s$ an $s \times d$ matrix. The correct choice of $s$ is a critical point. On the
one hand \( s \) has to be large enough to represent the original data’s structure, on the other, it should be small enough to filter out unimportant details. In practice, the parameter \( s \) is chosen such that it leads to good retrieval performance.

The coordinates of points that represent the documents and terms in the \( s \) dimensional space are given by the rows of the reduced matrices of singular vectors. To compare corresponding objects, dot products can be used. The most interesting case in the current context is the comparison of two documents. This relationship can be computed by the equation

\[
M'_s M = (T_s S_s D'_s)' (T_s S_s D'_s) = D_s S_s T_s' T_s S_s D'_s = D_s S_s S_s D'_s = D_s S_s^2 D'_s \tag{A.4}
\]

This implies the computation of element \((i, j)\) by taking the dot product between rows \(i\) and \(j\) of matrix \(D_s S_s\), describing the relationship between document \(i\) and document \(j\). In this case the rows of a \(D_s S_s\) matrix can be seen as coordinates for the documents. Because \(S_s\) is diagonal the positions of points are the same if we take \(D_s\) as coordinates or, with a small exception, \(D_s S_s\): the corresponding diagonal element of \(S_s\) shrinks or stretches the axes in proportion. Thus, a layout can be computed for the whole document collection.

The Self-Organizing Maps, introduced by Teuvo Kohonen (see [Koh97]), are the second focus of investigation in this context. Although the layout provides a similar effect on the user (i.e. semantic similarity is strongly related to closeness), the algorithm works in another way.

Let \(m_i\) be an ordered set of vector models, \(x\) the input vector, \(c = c(x)\) the unit that is most similar to \(x\), i.e. the BMU. To calculate \(c\) the Euclidian metric is generally used, which leads to the following formula:

\[
c = \min_i \{ \|x - m_i\| \} \tag{A.5}
\]

If \(t\) describes the time, i.e. the sample index of the regression step, this formula can also be written as:

\[
\|x(t) - m_c\| \leq \|x(t) - m_i(t)\| \forall i \tag{A.6}
\]

When the BMU is determined the learning process is started, in which the BMU and its neighbors learn from the input. This is defined by the \textit{neighborhood function} \(h\). \(h\) will become smaller with increasing distance from the BMU \(c\). Additionally, \(h\) decreases in dependency with the time \(t\). This leads to a well defined structure of the map at the beginning of the process and only fine-tuning at the end. The adaption of a unit during the process at the time stamp \(t\) thus can be described as:

\[
m_i(t + 1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \tag{A.7}
\]
A typical neighborhood function is the Gaussian function:

\[ h_{ci}(t) = \alpha(t) \exp \left( -\frac{\|r_i - r_c\|}{2\sigma^2(t)} \right), \]  

(A.8)

where \(0 < \alpha(t) < 1\) describes the learning-rate factor, \(r_i \in \mathbb{R}^2\) and \(r_c \in \mathbb{R}^2\) are two-dimensional vectors defining the position in the display grid, and \(\sigma(t)\) corresponds to the neighborhood distance. If \(t\) increases, \(\alpha(t)\) and \(\sigma(t)\) both decrease.

A more simple version of the neighborhood function can be written as:

\[ h_{ci}(t) = \begin{cases} \alpha(t) & \text{if } \|r_i - r_c\| < r(t), \\ 0 & \text{else} \end{cases} \]  

(A.9)

where \(r(t)\) defines the monotonically decreasing radius around \(c\).

After a couple of iterations, regions of similar inputs are formed as connected areas within the map. This leads to a layout like that shown in Figure 2.32.

The last algorithms presented in this context work with Force-Directed and Energy-Based Placement. A precondition for these algorithms is a given graph \(G = (V, E)\) of vertices \(V\) and edges \(E\), connected and undirected.

1. Force-Directed Placement:

Let \(p = (p_v)_{v \in V}\) be the vector that describes the vertex position \(p_v = (x_v, y_v)\). The Euclidean distance between positions \(p_u\) and \(p_v\) is described by the length of the difference vector \(p_v - p_u\), denoted as \(\|p_v - p_u\|\). In addition, the unit length vector \(\frac{p_v - p_u}{\|p_v - p_u\|}\) which is directed from \(p_u\) to \(p_v\) is referred to as \(p_u \overline{p_v}\). In his spring embedder model, Eades [Ead84] describes the operating forces by the following equations:

\[ f_{\text{rep}}(p_u, p_v) = c_{\rho} \cdot \frac{\|p_v - p_u\|^2}{\|p_v - p_u\|} \cdot p_u \overline{p_v} \]  

(A.10)

where \(u, v\) are non-adjacent vertices that repel each other and \(c_{\rho}\) is a repulsion constant. In contrast to these repelling forces the spring forces can be described as:

\[ f_{\text{spring}}(p_u, p_v) = c_{\sigma} \cdot \log \left( \frac{\|p_v - p_u\|}{l} \right) \cdot p_u \overline{p_v} \]  

(A.11)

where \(l\) is the natural length of the spring and \(c_{\sigma}\) controls the spring’s strength.

We now have to turn to the algorithm to arrive at an adequate layout. The aim is to reach a system at equilibrium. Therefore, the vertex positions are decisive. If the system is not in equilibrium, internal stress is implied. This can be changed by moving the vertices in accordance with a net force vector \(F_v(t)\) at a time \(t\). \(F_v(t)\) is computed by the sum of all repulsive and spring forces that influence vertex \(v\).
To position all vertices in a correct manner, first $F_v(t)$ has to be calculated for all $v \in V$, then the vertices are moved $\delta$ times this vector to prevent an excessive movement. After a set of iterations, the system reaches a stable state where no local improvement is possible. The pseudo code of Algorithm Spring Embedder describes the procedure. Many faster variations of this algorithm are known. The interested reader is therefore referred to [KW01].

**Algorithm: Spring Embedder**

**Input:**
- connected undirected graph $G = (V, E)$
- initial placement $p = (p_v)_{v \in V}$

**Output:**
- placement $p$ with low internal stress

for $t = 1$ to NUMBER OF ITERATIONS do 
  for $v \in V$ do 
    $F_v(t) = \sum_{u: \{u,v\} \in E} f_{rep}(p_u, p_v) + \sum_{u: \{u,v\} \in E} f_{spring}(p_u, p_v)$ 
    for $v \in V$ do 
      $p_v = p_v + \delta \cdot F_v(t)$

2. **Energy-Based Placement:**

As mentioned above, the second approach is an energy-based placement. In contrast to a movement of vertices as in the previous case, it is possible to minimize the system’s global energy directly. The potential energy of a spring of natural length $l$, strength $c_{\sigma}$, and actual length $d$ can be defined as

$$U_{spring}(d) = c_{\sigma} \cdot (d - l)^2$$  \hspace{1cm} (A.12)

To obtain the resulting objective function of the global energy, various computations are known. Examples are given by e.g. [KK89], or [DH96]. [KK89] define the objective function by

$$U(p) = \sum_{u,v \in V} \frac{c}{d_G(p_u, p_v)^2} \cdot (\|p_v - p_u\| - l \cdot d_G(u, v))^2,$$  \hspace{1cm} (A.13)

where $c$ is a scaling constant, $d_G(u, v)$ the length of a shortest path between $u$ and $v$, and $l$ the ideal length of a single edge.

In [DH96] the potentials for repulsion and attraction are calculated by

$$U_{attr}(p_u, p_v) = c_{attr} \cdot \|p_v - p_u\|^2$$  \hspace{1cm} (A.14)
and

$$U_{\text{rep}}(p_u, p_v) = \frac{c_{\text{rep}}}{\|p_v - p_u\|^2}$$  \hspace{1cm} (A.15)

which leads to a spring potential of

$$U_{\text{spring}}(p_u, p_v) = U_{\text{attr}}(p_u, p_v) + U_{\text{rep}}(p_u, p_v)$$  \hspace{1cm} (A.16)

for pairs of adjacent vertices.

To compute a local minimum of the objective function, simulated annealing is used. This model provides an analogy between the physical process of annealing, i.e. the way in which a metal cools and freezes into a minimum energy crystalline structure and the search for a minimum in a more general system. The advantage of simulated annealing is the fact that a captivity in a local minimum is avoided. Two main steps characterize the algorithm: change of temperature and computation of the energy level at this temperature. If a candidate solution is given, a new one is obtained by modifying the current one. For the case in which the new solution results in a lower value of the objective function, it becomes the new candidate solution, otherwise it becomes the new candidate solution with only a specific probability $e^{-\Delta U/T}$, where $\Delta U$ is the amount the objective function increased, and $T > 0$ is a temperature parameter. $T$ is used to enable the algorithm to transcend energy boundaries to find a solution that is likely to be better “behind” this boundary. The temperature is under the control of a temperature scheme or temperature scheduling. Thus, the optimization starts at a high temperature $T_0$ which means that a large percentage of random steps resulting in an energy increase is accepted. After a sufficient number of steps has been completed, the temperature is decreased until the final temperature $T_{\text{final}}$ is reached. Usually, one of the following two cooling schedules is used: a linear cooling schedule

$$T_{\text{new}} = T_{\text{old}} - dT,$$  \hspace{1cm} (A.17)

where $dT$ is a fixed value to decrease the temperature, or a proportional cooling schedule

$$T_{\text{new}} = c \times T_{\text{old}},$$  \hspace{1cm} (A.18)

where $0 < c < 1$.

To describe the algorithm more precisely, the pseudo-code ”Algorithm: Simulated Annealing” is presented below.
graph \( G = (V, E) \)
initial placement \( p = (p_v)_{v \in V} \)

**Output:**
placement \( p \) with locally optimal value \( U(p) \)

while \( T > \text{THRESHOLD} \) do
  for \( v \in V \) do
    \( p^{old} = p \)
    \( p_v = p_v + \Delta_{\text{random}} \)
    if \( U(p^{old}) < U(p) \) then
      with probability \( 1 - e^{\frac{U(p^{old})-U(p)}{T}} \)
      reset \( p = p^{old} \)
    reduce \( T \)
A.2 VisMeB Visualizations

In Chapter 3 various visualizations included in the VisMeB framework are mentioned, but not described in detail. This will be done in the following sections.

A.2.1 ScatterPlot-2D

The ScatterPlot (see Figure A.1) is a two-dimensional coordinate system enhanced by the possibility of allocating the x- and y-axis every kind of metadata used in the current context. It eases a comparison of data set properties like e.g. date, size or relevance. Single data points are displayed as small color-filled circles. MultiDataPoints, i.e. items sharing exactly the same position (see Section A.2.2.1) are represented by squares.

Using different colors for the data points adds another dimension to enable a faster perception of important facts. Unmodified points are separated from focussed (light shaded) and selected (dark shaded) points, so three different states are created. On MultiDataPoints, the fill factor adds another dimension, displaying whether no (unfilled), some (half filled) or all (full filled) underlying data points are selected. Additional features like visual filters or distortion techniques are presented in Chapter 5.

![Figure A.1: The 2D Scatterplot using a movable filter supporting the search for Danish documents](image)

A.2.2 ScatterPlot-3D

As well as the 2D-ScatterPlot, a 3D-ScatterPlot was created (see Figure A.2). Data points are therefore 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 grid’s edge to achieve better clarity. Free rotation providing an illumination from all directions, a zoom function as well as different selection mechanisms together complete the equipment of the 3D-ScatterPlot.
A.2.2.1 MultiDataPointView

Special attention was directed to the problem of data point overlapping in both ScatterPlot occurrences. Frequently, several objects share the same metadata for specific characteristics, leading to the same position in the drawing area, e.g. same date and relevance. For this reason, a new glyph has been introduced, the so-called "MultiDataPoint" (MDP), to point this fact out to the user. In the three-dimensional space, we visualized MDPs as two interlocked cubes. In the 2D version the circles representing single data sets are replaced by squares that are partially or fully color-filled to indicate if some or all of the underlying data points are selected. If the user moves the mouse over these MDPs, the current glyph is broadened by a hint showing the number of underlying data points. A single-click or the use of the context menu can change the view and the MDP-View appears (see Figure A.3).

All points underlying this MDP are positioned on a radial arrangement around the view’s center. Moving the mouse over sensitive areas (labeled "Accelerate") starts an animation. The index cards that represent single items "fly" in an ellipsoid orbit around the center. The direction and speed of the radial animation can be varied. The details of the item currently in the foreground, which are normally shown by tooltips, can be seen in the caption to the left. This technique makes it possible to explore a large set of data in an extremely short time. The idea is based on the "rapid serial visual presentation" [BS00], [Spe01b]. Originally, the presentation showed the content of a directory in a file system. We used this visualization to show all data points unified at this concrete location. A metaphor the user is familiar with is used to describe the task of ransacking a stack - index cards. It implies a short description for a data set to give an overview. More information can be explored by analyzing the data set itself, e.g. using the SuperTable. In an earlier, static version of a circular arrangement in VisMeB, a problem arose because...
A.2 VisMeB Visualizations

Figure A.3: MultiDataPoint-View to explore overlapping data points of the large amount of points building such a MDP. Too many data points made it impossible to differentiate between single data points. The use of animation helped to avoid this problem.

A.2.3 BrowserView

Giving the possibility of seeing more text than is feasible in a comparatively small table cell was the first idea behind the BrowserView. Rapidly, this was enhanced by other features like keyword highlighting, the opportunity of changing the font size or providing the possibility of comparing more than one data set exploiting the available space, and providing different alignments. Figure A.4 to the left shows three documents to be compared using a horizontal alignment; to the right the alignment is changed from horizontal to vertical. Both figures share the font size of 140%.

Figure A.4: BrowserView to compare documents using a horizontal (a) and a vertical (b) alignment
A.2.4 DocumentUniverse

The semantic similarity between data sets within the complete data set collection is another important piece of information that should not be suppressed. This information can be found by comparing single data sets to one another. But to compare all data sets (e.g. resulting in a similarity matrix), the effort would be too time consuming. The DocumentUniverse is therefore introduced. Via an LSA (Latent Semantic Analysis) algorithm, a layout is computed to visualize the semantic similarity between all data sets (see Figure A.5). Similar data sets are positioned close to one another, non-similar ones wide apart. Because of the restriction on a two-dimensional layout, the positioning will always be only as good as the algorithm allows. Hence, you will get hints about which data have to be explored in more detail by hand.

Figure A.5: DocumentUniverse displaying the semantic similarity between all data sets

A.2.5 LocationMap

The LocationMap is only used in the library scenario. It displays a map of the media center, the "Mediothek", a part of the library of the University of Konstanz where media like DVD, CD-Rom, VHS are located. As soon as the user focusses a single data set in e.g. the table or the scatterplot, the corresponding site in the library shelves is highlighted on the map. This enables the user to find the requested medium more easily and faster.

A.2.6 Visual Configurator

The Visual Configurator is an application that works completely separated from the query and the result-set visualizations. If an administrator includes and adapts the correct assignments, there is no necessity for a normal user to load the configurator. The idea is to keep the application independent from the domain when visualizing any data. The first
A.2 VisMeB Visualizations

Figure A.6: LocationMap displaying the location of the focussed media

step is to load a database into the program. Additionally, further ”internal” data like ”relevance”, ”internal ID”, or ”selection”, which are created when the program is running, are included. The second step is to assign the data to the desired visualizations. Completely different assignments can be made and stored to adapt the visualization to the current task. This allocation can be done by simple drag & drop actions, i.e. dragging a meta-data from the database window to the visualizations window (see Figure A.7). Subsequent changes can be made by loading the assignment, changing the respective entries and then saving. This configurator supports the adaption of VisMeB to a completely new domain in a very simple, but efficient way.
Figure A.7: Visual Configurator showing the meta-data to be visualized on the left side and the assignment to the diverse visualizations on the right side. The meta-data "domain_category" is being dragged to the visualization window on the right.
The original terms of the evaluation tasks were written in German and therefore will be presented first. To ease the work for the English-speaking reader, a translation is given afterwards in section B.4. The complete content of the following section is taken from [Ger03]. For further information please have a look at his thesis.

B.1 Pre-Test Fragebogen

1. Wie viele Stunden am Tag benutzen Sie einen Computer?

2. Fällt es Ihnen leicht sich mit neuer Software vertraut zu machen?
   (1 bedeutet "nein, fällt mir eher schwer", 7 bedeutet "ja, bereitet mir keine Probleme")

3. Wie schätzen Sie Ihre Erfahrung mit Computern allgemein auf einer Skale von 1-7 ein?
   (1 bedeutet "keine Erfahrung", 7 bedeutet "sehr viel Erfahrung")

4. Wie würden Sie ihre Erfahrung mit Internet Suchmaschinen auf einer Skala von 1-7 einschätzen?
   (1 bedeutet "keine Erfahrung" und 7 bedeutet "sehr viel Erfahrung")

5. Wie würden Sie Ihre generelle Einstellung zu Computern auf einer Skala von 1-7 einschätzen?
   (1 bedeutet "ärbeite sehr ungern mit Computern" und 7 bedeutet "ärbeite sehr gerne mit Computern")

B.2 Performance Test Ergebnisse

1. Da diese Aufgabe lediglich das Starten der Suche beinhaltete, wurde sie für die Auswertung nicht berücksichtigt.
2. "Suchen Sie das Dokument mit der höchsten Relevanz und nennen Sie den Titel”

Fehleranalyse:
Kontrollgruppe: 1 Fehler
Versuchsgruppe: 0 Fehler

Kontrollgruppe:

3. "Vergleichen Sie die Dateigröße der ersten 10 Dokumente (geordnet nach Relevanz - dieser Zusatz existierte nur bei Verwendung der Visualisierung) - welches ist das grösste?”

Fehleranalyse:
Kontrollgruppe: 0 Fehler
Versuchsgruppe: 4 Fehler

Versuchsgruppe:
Zwei VP wählten einfach das falsche Dokument aus, zwei weitere sortierten die Tabelle nach Grösse und wählten das insgesamt grösste Dokument aus.

4. "In welcher Sprache ist dieses Dokument vorhanden und wie lautet der server_type?”

Fehleranalyse:
Keine Fehler

5. "Sie suchen ein deutschsprachiges Dokument. Überprüfen Sie ob in der Ergebnismenge eines vorhanden ist” (bei der Listendarstellung wurde der Ergebnisraum auf 100 Dokumente eingeschränkt, da hier ansonsten ein zu grosser Nachteil entstanden wäre)

Fehleranalyse:
Keine Fehler

B.2 PERFORMANCE TEST ERGEBNISSE

Fehleranalyse:
Kontrollgruppe: 0 Fehler
Versuchsgruppe: 3 Fehler

Versuchsgruppe:
Wie bei der ähnlichen Aufgabe 3 hatte die Kontrollgruppe keinerlei Probleme. Bei der Versuchsgruppe hatten drei VP Probleme die Aufgabe richtig zu beantworten. Dabei ist auch eine VP, die bereits Aufgabe 3 nicht korrekt beantworten konnte. Die Versuchspersonen entschieden sich alle drei für den falschen server_type, obwohl .com relativ offensichtlich ist.

7. "Suchen Sie das Dokument - GISLinx - What is a GIS (Zusatz Kontrollgruppe: Es ist innerhalb der ersten 100 Dokumente enthalten)"

Fehleranalyse:
Kontrollgruppe: 1 Fehler
Versuchsgruppe: 2 Fehler

Kontrollgruppe:
Eine VP fand das Dokument nicht, erst nach Tipp auf der ersten Seite noch mal nachzusehen.

Versuchsgruppe:
2 VP fanden das Dokument nicht, da bei Sortierung nach Titel zu oberst die Dokumente stehen, welche mit einem Leerzeichen beginnen. Erst nach mehrmaligen Hinweisen auf diese Tatsache fanden beide VP das Dokument.

8. "Bietet das Dokument GISLinx - What is a GIS Informationen bezüglich des Suchterms "Design" (ist das Wort darin enthalten)?"

Fehleranalyse:
Kontrollgruppe: 5 Fehler
Versuchsgruppe 2 Fehler

Kontrollgruppe:
Die Kontrollgruppe hatte deutliche Probleme die Aufgabe zu lösen. Die Dokumentenansicht der Listendarstellung bot keine Suchfunktion, weswegen das Dokument durchgelesen werden musste. Einige Teilnehmer hatten dazu keine Lust und brachen einfach sofort ab (VP6,VP14,VP18,VP29). Eine Versuchsperson (VP27) durchsuchte das Dokument zwar sehr lange (über zweieinhalb Minuten), konnte sich am Ende aber nicht entscheiden.
Versuchsgruppe:
Bei der Versuchsgruppe ergibt sich ein anderes Bild. Eine Versuchsperson (VP15) brach die Aufgabe nach 2 Minuten 40 Sekunden ab, ohne die Lösung zu nennen. Eine weitere (VP24) fand sich in der Browserview nicht zurecht - nach Tipp vom VL doch mal die Farbkodierung zu beachten, konnte sie die Aufgabe schliesslich aber doch korrekt lösen.

9. "Vergleichen Sie alle Dokumente - welches ist das kleinste Dokument (ausgenommen Dokumente mit einem Wert von "-1")?" Anmerkung: Die Kontrollgruppe durfte sich auf die ersten 20 Dokumente beschränken, da der Aufwand hier ansonsten unverhältnismässig hoch gewesen wäre

Fehleranalyse:
Kontrollgruppe: 2 Fehler
Versuchsgruppe: 2 Fehler

Kontrollgruppe:
2 VP (VP31 & VP6) entschieden sich für das falsche Dokument.

Versuchsgruppe:
Bei einer VP (VP1) war diese Aufgabe noch nicht in dieser Form im Test enthalten, die andere VP (VP9) suchte anstelle des kleinsten Dokument das grösste und fand hier auch das richtige, insofern auch nur ein Missverständnis der Frage.

10. "Suchen Sie erneut das Dokument - GISLinx - What is a GIS"

Fehleranalyse:
Kontrollgruppe: 1 Fehler
Versuchsgruppe: 0 Fehler

Kontrollgruppe:
VP6 überspringt Aufgabe aus Versehen

11. "Finden Sie mindestens 5 Dokumente mit dem Server_type .edu"

Fehleranalyse:
Keine Fehler

12. "Versuchen Sie aus diesen das grösste Dokument auszuwählen"

Fehleranalyse:
Kontrollgruppe: 3 Fehler
Versuchsgruppe: 1 Fehler

Kontrollgruppe:
3 Versuchspersonen (VP6, VP8 & VP20) entschieden sich für das falsche Dokument.

Versuchsgruppe:
1 Versuchsperson verstand die Aufgabe nicht und brach sie deswegen ab.


Fehleranalyse:
Kontrollgruppe: 5 Fehler
Versuchsgruppe: 6 Fehler

Kontrollgruppe:
Zwei Versuchspersonen brechen die Aufgabe nach einiger Zeit ab ohne eine Lösung zu nennen. Zwei weitere sind zunächst der Meinung, mehrere Dokumente finden zu müssen - nach Hilfe finden sie das richtige Dokument, können aber Aufgrund der Thematik nur raten. Wie die 5. VP nennen sie letztendlich zwei richtige Lösungen, brechen die Aufgabe aber dennoch ab, da sie nicht sicher sind.

Versuchsgruppe:
Zwei Versuchspersonen haben bereits Probleme das richtige Dokument zu finden und erhalten hierbei kleine Hilfestellungen (Dokument suchen, welches vom Titel bereits ähnlich der Suchanfrage ist). Eine der beiden VP kann daraufhin auch die Aufgabe noch korrekt lösen. Eine weitere VP findet das richtige Dokument, ist sich dessen aber nicht bewusst und weiß nicht richtig, was sie machen soll - kleiner tipp dass VP sich im richtigen Dokument befindet und nur die Systeme nennen soll hilft ihr soweit, dass sie die Aufgabe noch korrekt lösen kann. Drei weitere VP finden zwar das richtige Dokument, brechen dann aber ab ohne Lösung zu nennen.

B.3 Post-Test Fragebogen

Lediglich die Teilnehmer der Versuchsgruppe (18) mussten den Post-Test Fragebogen ausfüllen. Für alle Fragen galt die Vorgabe: 1 bedeutet "nicht mit einverstanden" und 7 bedeutet "Ja, das stimmt!"
(a) War das generelle Layout der Oberfläche klar?
(b) War der Gebrauch von Farben angenehm?
(c) War die Navigation intuitiv?
(d) Haben Sie sich manchmal "verloren" gefühlt?
(e) War auf einzelnen Seiten die Informationsflut zu hoch?
(f) Boten einige Seiten zu wenig Informationen?
(g) War die verwendete Terminologie verständlich?
(h) Könnten Sie sich vorstellen, mit dieser Visualisierung (Leveltable) täglich zu arbeiten?
   Ergebnis: 14 x Ja, 4 x Nein
(i) Könnten Sie sich Situationen vorstellen, in denen eine derartige Visualisierung (Leveltable) einer herkömmlichen, listenbasierten Darstellung (wie Google) überlegen ist?
   Ergebnis: 18 x Ja

Table B.1: Mittelwerte der Likert-Skalen

<table>
<thead>
<tr>
<th>GRUPPE</th>
<th>F. 1</th>
<th>F. 2</th>
<th>F. 3</th>
<th>F. 4</th>
<th>F. 5</th>
<th>F. 6</th>
<th>F. 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>VG</td>
<td>5.61</td>
<td>6.11</td>
<td>5.11</td>
<td>4.00</td>
<td>4.17</td>
<td>3.00</td>
<td>5.53</td>
</tr>
</tbody>
</table>

B.4 Pre-Test Questionnaire

i. How many hours do you use a PC every day?

ii. Is it easy for you to get used to a new piece of software?
(1 means "No, it is very hard for me", 7 means "Yes, it is very easy for me")

iii. How do you rank your computer experience in general?
(1 means "No experience", 7 means "Very experienced")

iv. How do you rank your experience with internet search engines?
(1 means "No experience", 7 means "Very experienced")

v. How do you rank your general attitude to PCs?
(1 means "I work with PCs very unwillingly", 7 means "I really like working with PCs")

Further test persons’ characteristics:
### Table B.2: Mean values for the Pre-Test questionnaire

<table>
<thead>
<tr>
<th>GROUP</th>
<th>AGE</th>
<th>QUEST. 1</th>
<th>QUEST. 2</th>
<th>QUEST. 3</th>
<th>QUEST. 4</th>
<th>QUEST. 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental Group</td>
<td>25.56</td>
<td>5.06</td>
<td>5.33</td>
<td>5.50</td>
<td>5.33</td>
<td>5.83</td>
</tr>
<tr>
<td>Control Group</td>
<td>26.29</td>
<td>4.77</td>
<td>4.21</td>
<td>4.57</td>
<td>4.14</td>
<td>5.21</td>
</tr>
<tr>
<td>Total</td>
<td>25.88</td>
<td>4.93</td>
<td>4.84</td>
<td>5.09</td>
<td>4.81</td>
<td>5.56</td>
</tr>
</tbody>
</table>

### Table B.3: Gender

<table>
<thead>
<tr>
<th></th>
<th>OCCURENCES</th>
<th>PERCENTAGE</th>
<th>ACCUMULATED PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>24</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>Female</td>
<td>8</td>
<td>25</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>32</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

### Table B.4: Education

<table>
<thead>
<tr>
<th></th>
<th>OCCURENCES</th>
<th>PERCENTAGE</th>
<th>ACC. PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grammar School</td>
<td>28</td>
<td>87.5</td>
<td>87.5</td>
</tr>
<tr>
<td>Secondary Modern</td>
<td>1</td>
<td>3.1</td>
<td>90.6</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>3</td>
<td>9.4</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>32</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

### Table B.5: Study / Education / Profession

<table>
<thead>
<tr>
<th></th>
<th>OCCURENCES</th>
<th>PERCENTAGE</th>
<th>ACC. PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSc IE</td>
<td>7</td>
<td>21.9</td>
<td>21.9</td>
</tr>
<tr>
<td>Information Science</td>
<td>1</td>
<td>3.1</td>
<td>25.0</td>
</tr>
<tr>
<td>Computer Science</td>
<td>1</td>
<td>3.1</td>
<td>28.1</td>
</tr>
<tr>
<td>Information Technology Officer</td>
<td>1</td>
<td>3.1</td>
<td>31.3</td>
</tr>
<tr>
<td>Law</td>
<td>3</td>
<td>9.4</td>
<td>40.6</td>
</tr>
<tr>
<td>Literature</td>
<td>1</td>
<td>3.1</td>
<td>43.8</td>
</tr>
<tr>
<td>Engineering</td>
<td>4</td>
<td>12.5</td>
<td>56.3</td>
</tr>
<tr>
<td>MSc IE</td>
<td>5</td>
<td>15.6</td>
<td>71.9</td>
</tr>
<tr>
<td>Pharmacy</td>
<td>1</td>
<td>3.1</td>
<td>75.0</td>
</tr>
<tr>
<td>Physics / English</td>
<td>1</td>
<td>3.1</td>
<td>78.1</td>
</tr>
<tr>
<td>Self-employed</td>
<td>2</td>
<td>6.3</td>
<td>84.4</td>
</tr>
<tr>
<td>Mathematics</td>
<td>1</td>
<td>3.1</td>
<td>87.5</td>
</tr>
<tr>
<td>Management Science</td>
<td>3</td>
<td>9.4</td>
<td>96.9</td>
</tr>
<tr>
<td>Research Assistant</td>
<td>1</td>
<td>3.1</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>32</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

### Table B.6: PC Experience

<table>
<thead>
<tr>
<th></th>
<th>OCCURENCES</th>
<th>PERCENTAGE</th>
<th>ACCUMULATED PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>More than a year</td>
<td>32</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
### Table B.7: Operating System used

<table>
<thead>
<tr>
<th>OCCURRENCES</th>
<th>PERCENTAGE</th>
<th>ACCUMULATED PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win9x</td>
<td>6</td>
<td>18.8</td>
</tr>
<tr>
<td>Win2K/XP</td>
<td>24</td>
<td>75.0</td>
</tr>
<tr>
<td>Linux</td>
<td>2</td>
<td>6.3</td>
</tr>
<tr>
<td>Total</td>
<td>32</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table B.8: Personal connection to Computer Experts

<table>
<thead>
<tr>
<th>OCCURRENCES</th>
<th>PERCENTAGE</th>
<th>ACCUMULATED PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>31</td>
<td>96.9</td>
</tr>
<tr>
<td>No</td>
<td>1</td>
<td>3.1</td>
</tr>
<tr>
<td>Total</td>
<td>32</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table B.9: Use of internet search engines

<table>
<thead>
<tr>
<th>OCCURRENCES</th>
<th>PERCENTAGE</th>
<th>ACCUMULATED PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>32</td>
<td>100</td>
</tr>
<tr>
<td>No</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>32</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table B.10: Internet search engine used most often

<table>
<thead>
<tr>
<th>OCCURRENCES</th>
<th>PERCENTAGE</th>
<th>ACCUMULATED PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>32</td>
<td>100</td>
</tr>
</tbody>
</table>
B.5 Main Test Performance Results

To present the results of the main test, the questions are first noted, followed by an error analysis for the control group (CG) and the experimental group (EG), and a detailed description of the kind of errors.

i. This task concerned the start of a new query and therefore was not included for the analysis!

ii. Search for the document with the highest relevance and name its title!
   CG: 1 error
   EG: 0 errors

   CG: One test person chose the second document on the list because in her eyes it represented a better match of title to query.

iii. Compare the size of the first ten files, ordered by relevance. Which one is the largest?
    CG: 0 errors
    EG: 4 errors

    EG: Two test persons simply chose the wrong document, two sorted the table by size and chose the overall largest document.

iv. Name the language of this document and the server-type!
    No errors.

v. You are looking for a German document. Please check if the result set contains a German document!
    No errors.

vi. Compare the server-type of the first 20 documents (ordered by relevance). Which server-type seems to appear the most?
    CG: 0 errors
    EG: 3 errors

    EG: All three test persons chose the wrong server-type.

vii. Search for the document GISLinx - What is a GIS?!
     CG: 1 error
     EG: 2 errors

     CG: One test person could not find the document without a hint from the evaluator to look on the first page.

     EG: Two test participants could not find the document because of the
lexicographic order and its consequence viz. that titles containing leading spaces were displayed first.

viii. *Does the document GISLinx - What is a GIS? contain information about the query term Design?*
CG: 5 errors  
EG: 2 errors

CG: Because no search feature was implemented for the list presentation, the document had to be examined visually. Four test participants gave up almost immediately, one person searched for a long time (more than 2-1/2 minutes) but could not come to a decision.

EG: One test participant gave up the search after two minutes and 40 seconds without being able to name a solution. One person could not manage the BrowserView without the help of the evaluator. After a hint to consider the color coding, she finally solved the task correctly.

ix. *Compare all documents. Which one is the smallest (except the ones with a value of ”-1”)?*
CG: 2 errors  
EG: 2 errors

CG: Two test participants chose the wrong document.

EG: This question in this form was not included in the test for test participant V1; a further person searched for the largest instead of the smallest document and found the right one. Thus, the question was misunderstood, leading to the wrong answer.

x. *Search for the document GISLinx - What is a GIS? again!*
CG: 1 error  
EG: 0 errors

CG: One test participant skipped past this question unintentionally.

xi. *Find at least five documents with server-type .edu!*
No errors.

xii. *Choose the document with the largest size from these five documents!*
CG: 3 errors  
EG: 1 error

CG: Three test participants chose the wrong document.
EG: One test participant did not understand the question and hence aborted the task.

xiii. Find out which Online Mapping Systems are provided by Geoscience Australia. Name at least three. To solve this task you are allowed to start a new query.
CG: 5 errors
EG: 6 errors

CG: Two test participants aborted the task after a while without giving any answer. Two others initially thought that they had to search for several documents. After a hint from the evaluator they found the correct document could be found. Nevertheless, only two correct answers were given. The same result applied to the fifth test participant. All three aborted the task because they were not sure about the correctness of their answers.

EG: Two test participants had problems in finding the correct document. After a hint from the evaluator, one person was able to solve the task in a correct manner. A further participant found the correct document but was not aware of it. Again, a hint helped to get the right answers. Three other persons found the correct document but aborted the task without naming the right answer.

B.6 Post-Test Questionnaire

Only the members of the experimental group had to fill out the post-test questionnaire, because it contained questions about the LevelTable visualization that was presented. A seven-point likert scale was used for all questions, where 1 meant "I strongly disagree" and 7 meant "I strongly agree".

i. Did the general layout of the interface appear clear to you?
ii. Was the use of color pleasing?
iii. Was the navigation intuitive?
iv. Did you sometimes feel "lost"?
v. Was the amount of information on single pages too high?
vi. Did some pages provide too little information?
vii. Was the terminology used easy to understand?
viii. Can you imagine working with this visualization (LevelTable) on a daily basis?

Result: Fourteen $x$ YES, four $x$ NO
ix. Can you imagine a situation where such a visualization is superior to a conventional list-based presentation (like Google)?
Result: Eighteen $\times$ YES

\textit{Table B.11: Mean values for the post-test questionnaire}

<table>
<thead>
<tr>
<th>GROUP</th>
<th>Q. 1</th>
<th>Q. 2</th>
<th>Q. 3</th>
<th>Q. 4</th>
<th>Q. 5</th>
<th>Q. 6</th>
<th>Q. 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>EG</td>
<td>5.61</td>
<td>6.11</td>
<td>5.11</td>
<td>4.00</td>
<td>4.17</td>
<td>3.00</td>
<td>5.53</td>
</tr>
</tbody>
</table>


[AM04] AIGNER, Wolfgang ; MIKSCHE, Silvia: Supporting Protocol-Based Care in Medicine via Multiple Coordinated Views. In: Second International Conference on Coordinated Multiple Views in Exploratory Visualization (CMV’04), 2004, S. 118–129


[BR03] BOUKHELIFA, Nadia ; RODGERS, Peter J.: A model and software system for coordinated and multiple views in exploratory visualization. In: Information Visualization 2 (2003), Nr. 4, S. 258–269. – ISSN 1473–8716


[Car00] CARROLL, J. M.: Introduction to this special issue on ”scenario-based system development”. In: Interacting with Computers 13 (2000), Nr. 1, S. 41–42


[Ead84] EADES, P. A heuristic for graph drawing. 1984


[HS00] H.HOCHHEISER ; SHNEIDERMAN, B. Performance benefits of simultaneous over sequential menus as task complexity increases. 2000


**BIBLIOGRAPHY**


[Kle05] KLEIN, P.: *Circle Segment View*, University of Konstanz, Diss., 2005. – PhD Thesis


[NCIS02] NORTH, Chris ; CONKLIN, Nathan ; INDUKURI, Kiran ; SAINI, Varun: Visualization schemas and a web-based architecture for custom multiple-view visualization of multiple-table databases.


Ein visueller Metadaten Browser für die explorative Erkundung großer Datenmengen
Reiterer, H., Limbach, T., Klein, P., Müller, F., Jetter, H.-C.

Scenario based Design of a Visual Meta-data Browser
Limbach, T., Reiterer, H., Klein, P., Müller, F.

Visualization and Interaction Techniques of the Visual Metadata Browser VisMeB
Müller, F., Klein, P., Limbach, T., Reiterer, H.

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 International 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,

Visualization of Metadata using the SuperTable+Scatterplot
Limbach, T., Müller, F., Klein, P., Reiterer, H., Eibl, M.

INVISIP: Usage of Information Visualization Techniques to Access Geospatial Data Archives
Göbel S. and Haist J. and Müller F. and Reiterer H.

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

INVISIP: Metadata-based Information Visualization Techniques to Access Geodata Archives and to Support the Site Planning Process
Göbel S. and Haist J. and Müller F. and Reiterer H.
5-T Environment, 55
Activity-Based Models, 159
   Instructing, 160
   Manipulating And Navigating, 160
Activity-Based Models:Conversing, 160
Activity-Based Models:Exploring And Browsing, 160
AdministrationObject, 85
AOI, 97
ArcMap, 32
Aspect Of Interest, 97
Attentive Environments, 160
Attribute Explorer, 58
Augmented Reality, 160
Augur, 61
Bead, 44
Best Matching Unit, 39
Bifocal Display, 13
BMU, 39
Brushing, 9, 15
Camera Movement, 12, 20
CareVis, 56
Cartia, 44
ClusCorr98, 30
Conceptual Coordination Models, 80
   Exploratory Visualization Model, 82
   Snap, 80
   View Coordination Architecture, 81
   VisMeB Conceptual Model, 84
Conceptual Design, 156, 158
   Design Aspects, 159
   Principles, 159

Conceptual Model, 158, 159
   Activity-Based, 159
   Object-Based, 159
Context Of Use, 157
Contextual Task Analysis, 158
   Hierarchical Task Analysis, 158
   HTA, 158
Data Gathering, 156, 157
   Documentation, 157
   Focus Groups, 157
   Naturalistic Observation, 157
   Questionnaires, 157
   Workshop, 157
Data Requirements, 157
Data Type by Task Taxonomy, 8
DataGathering
   Interviews, 157
DateLens, 50
DECIDE Framework, 161
DENIM, 52
Design Phase, 156, 158
   Conceptual Design, 158
   Conceptual Model, 158
   Physical Design, 158
DestinationObject, 85
DEVise, 61
Dimension of Interest, 198
DimOI, 198
Distortion, 12
DROID, 195
Dublin Core, 1, 12, 13
Dynamic HomeFinder, 14, 23
Dynamic Queries, 9
Dynamic Query, 14

Effectiveness, 151
Efficiency, 151
Energy-Based Placement, 216
Environmental Requirements, 157
Envision, 27
EOSDIS, 154
Essential Use Cases, 158
Evaluation Paradigms, 156
   Field Studies, 160
   Predictive Evaluation, 160
   Quick And Dirty, 160
   Usability Testing, 160
Evaluation Phase, 156, 158
   Prototype, 158
      Evolutionary, 159
      High-Fidelity, 158
      Horizontal, 159
      Low-Fidelity, 158
      Throwaway, 159
      Vertical, 159
Evaluation Techniques, 156
   Asking Experts, 161
   Asking Users, 161
   Modeling Users’ Task Performance, 161
   Observing Users, 160
   User Testing, 161
Exploratory Data Visualizer, 58
Extraction and Comparison, 9

Field Studies, 160
FilmFinder, 25, 63
Filtering, 9
Focus & Context, 18
Focus Groups, 157
Force-Directed Algorithm, 39
Force-Directed Placement, 215
Functional Requirements, 156

Galaxy, 41
Galaxy Of News, 41
Gaussian Function, 215

Generic Genome Browser, 63
GranularityTable, 166

Hierarchical Cluster Explorer, 58
Hierarchical Task Analysis, 158
High-Fidelity Prototype, 158
Highlighting, 9
Horizontal Prototype, 159
Hubble Space Telescope, 154
HutchWorld, 153
HyperSlice, 32

IN-SPIRE, 41
Index Cards, 158
Influence Explorer, 32
Information Mural, 64
Information Visualization, 7
InfoSky, 41
Infozoom, 49
INSYDER, 163

Interaction Design, 152, 154
   Basic Activities, 154
   Lifecycle Models, 155
      RAD, 155
      Rapid Applications Development, 155
      Spiral Lifecycle, 155
      Star Lifecycle, 155
      Waterfall Lifecycle, 155
Interaction Paradigm, 160
   Attentive Environments, 160
   Augmented Reality, 160
   Pervasive Computing, 160
   Tangible Bits, 160
   Ubiquitous Computing, 160
   Wearable Computing, 160
Interactive Timeline Viewer, 25
Interface Metaphor, 160
Interviews, 157
INVISIP, 4
ISO 19115, 1
ISO 92115, 13

Jazz, 51
Kohonen Layer, 39
Kohonen Map, 39, 48

Latent Semantic Indexing, 38, 213
Law Of Creativity, 159
LevelTable, 166
Lifecycle Models, 155
Linking, 9, 15
LinkKit, 58
Location Probes, 12
Logos Bible Software, 60
Low-Fidelity Prototype, 158
LSI, 38, 213
  Singular-Value Decomposition, 38
  SVD, 38
  Term-Document Matrix, 38

Magic Lens, 12, 16
Magic Lenses, 9
MCV, 55
Meta-Data, 11, 13
Mirage, 58
Model-View-Controller Pattern, 81, 84, 102
Movable Filter, 12, 16
Multiple Coordinated View, 55
Multiple Coordinated Views
  Conceptual Coordination Models, 80

Naturalistic Observation, 157
Navigation, 55
Navigational View Builder, 58
Neighborhood Function, 215
Net Force Vector, 215
Neural-Network Algorithm, 38
Neuron, 39
NIRVE, 33
Node, 39
Non-Functional Requirements, 156

Object-Based Models, 159
Olympic Messaging System, 153
OMS, 153
OpenDX, 24, 33
Overview + Detail, 17
Overview+Detail, 17

Pad, 51
Pad++, 51
Panning & Zooming, 19
Perspective Distortion, 9
Perspective Wall, 13, 18
Pervasive Computing, 160
Physical Design, 158
Piccolo, 51
Pilot Study, 162
Polaris, 29
Potential Energy, 216
Predictive Evaluation, 160
Prototype, 158

Questionnaires, 157
Quick And Dirty, 160

RAD, 155
Rapid Applications Development, 155
Raw Data, 11
Relational Topic Mapping, 44
Repulsion, 39, 215
Requirement Analysis, 156
Requirements, 156
  Context Of Use, 157
  Data Requirements, 157
  Environmental Requirements, 157
    Organizational, 157
    Physical, 157
    Social, 157
    Technical, 157
  Functional Requirements, 156
  Non-Functional Requirements, 156
  Usability Requirements, 157
  User Requirements, 157
Requirements Analysis, 156
Retinal Properties, 9

Satisfaction, 151
Scatterplot, 22
  2D-Scatterplot, 23
  3D-Scatterplot, 33
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographical Scatterplot</td>
<td>23</td>
</tr>
<tr>
<td>Scatterplot Matrix</td>
<td>29</td>
</tr>
<tr>
<td>Scenario</td>
<td>1</td>
</tr>
<tr>
<td>Scenarios</td>
<td>157</td>
</tr>
<tr>
<td>Scientific Visualization</td>
<td>7</td>
</tr>
<tr>
<td>Search Result Explorer</td>
<td>28</td>
</tr>
<tr>
<td>SeeNet</td>
<td>23</td>
</tr>
<tr>
<td>Seesoft</td>
<td>61</td>
</tr>
<tr>
<td>Selection</td>
<td>55</td>
</tr>
<tr>
<td>Selective Aggregation</td>
<td>9</td>
</tr>
<tr>
<td>Self-Organizing Map</td>
<td>38, 48</td>
</tr>
<tr>
<td>Self-Organizing Maps</td>
<td>214</td>
</tr>
<tr>
<td>Semantic Similarity Maps</td>
<td>37</td>
</tr>
<tr>
<td>Semantic Zoom</td>
<td>50</td>
</tr>
<tr>
<td>SERS</td>
<td>154</td>
</tr>
<tr>
<td>Silver2</td>
<td>51</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>39, 217</td>
</tr>
<tr>
<td>Simultaneous Menus</td>
<td>64</td>
</tr>
<tr>
<td>Singular-Value Decomposition</td>
<td>38</td>
</tr>
<tr>
<td>Sketching</td>
<td>158</td>
</tr>
<tr>
<td>Sockeye</td>
<td>52</td>
</tr>
<tr>
<td>SOM</td>
<td>38, 48, 214</td>
</tr>
<tr>
<td>Best Matching Unit</td>
<td>39</td>
</tr>
<tr>
<td>BMU</td>
<td>39</td>
</tr>
<tr>
<td>Gaussian Function</td>
<td>215</td>
</tr>
<tr>
<td>Kohonen Function</td>
<td>39</td>
</tr>
<tr>
<td>Kohonen Map</td>
<td>39</td>
</tr>
<tr>
<td>Neighborhood Function</td>
<td>215</td>
</tr>
<tr>
<td>Neuron</td>
<td>39</td>
</tr>
<tr>
<td>Node</td>
<td>39</td>
</tr>
<tr>
<td>Unit</td>
<td>39</td>
</tr>
<tr>
<td>Winner-Neuron</td>
<td>39</td>
</tr>
<tr>
<td>Spiral Lifecycle</td>
<td>155</td>
</tr>
<tr>
<td>SPIRE</td>
<td>41, 44</td>
</tr>
<tr>
<td>Spotfire</td>
<td>24</td>
</tr>
<tr>
<td>Spreadsheet Framework</td>
<td>32</td>
</tr>
<tr>
<td>Spring Embedder</td>
<td>39, 216</td>
</tr>
<tr>
<td>Springs</td>
<td>39</td>
</tr>
<tr>
<td>Stakeholders</td>
<td>153</td>
</tr>
<tr>
<td>Facilitating Stakeholders</td>
<td>153</td>
</tr>
<tr>
<td>Primary Stakeholders</td>
<td>153</td>
</tr>
<tr>
<td>Secondary Stakeholders</td>
<td>153</td>
</tr>
<tr>
<td>Tertiary Stakeholders</td>
<td>153</td>
</tr>
<tr>
<td>StartObject</td>
<td>85</td>
</tr>
<tr>
<td>State of the Art</td>
<td>7</td>
</tr>
<tr>
<td>Storyboarding</td>
<td>156</td>
</tr>
<tr>
<td>SVD</td>
<td>38</td>
</tr>
<tr>
<td>SYNTH</td>
<td>20</td>
</tr>
<tr>
<td>TableLens</td>
<td>50</td>
</tr>
<tr>
<td>Tangible Bits</td>
<td>160</td>
</tr>
<tr>
<td>Task Analysis</td>
<td>156</td>
</tr>
<tr>
<td>Task Description</td>
<td>157</td>
</tr>
<tr>
<td>Essential Use Cases</td>
<td>158</td>
</tr>
<tr>
<td>Scenarios</td>
<td>157</td>
</tr>
<tr>
<td>Use Cases</td>
<td>157</td>
</tr>
<tr>
<td>Temperature</td>
<td>40, 217</td>
</tr>
<tr>
<td>Temperature Scheduling</td>
<td>40, 217</td>
</tr>
<tr>
<td>Linear Cooling Schedule</td>
<td>217</td>
</tr>
<tr>
<td>Proportional Cooling Schedule</td>
<td>217</td>
</tr>
<tr>
<td>Temperature Scheme</td>
<td>40, 217</td>
</tr>
<tr>
<td>Term-Document Matrix</td>
<td>38</td>
</tr>
<tr>
<td>Themescape</td>
<td>42</td>
</tr>
<tr>
<td>TimeSearcher</td>
<td>57</td>
</tr>
<tr>
<td>TrendDisplay</td>
<td>52</td>
</tr>
<tr>
<td>Ubiquitous Computing</td>
<td>160</td>
</tr>
<tr>
<td>UCD</td>
<td>151</td>
</tr>
<tr>
<td>Unit</td>
<td>39</td>
</tr>
<tr>
<td>Usability</td>
<td>3, 152</td>
</tr>
<tr>
<td>Usability Engineering Lifecycle</td>
<td>156</td>
</tr>
<tr>
<td>Usability Requirements</td>
<td>157</td>
</tr>
<tr>
<td>Usability Testing</td>
<td>160</td>
</tr>
<tr>
<td>Use Cases</td>
<td>157</td>
</tr>
<tr>
<td>User Centered Design</td>
<td>151, 152</td>
</tr>
<tr>
<td>Principles</td>
<td>152</td>
</tr>
<tr>
<td>Priniciples</td>
<td>152</td>
</tr>
<tr>
<td>User Requirements</td>
<td>157</td>
</tr>
<tr>
<td>Vertical Prototype</td>
<td>159</td>
</tr>
<tr>
<td>ViewNotifier</td>
<td>85</td>
</tr>
<tr>
<td>Viewpoint Controls</td>
<td>12</td>
</tr>
<tr>
<td>Views</td>
<td>11</td>
</tr>
<tr>
<td>Visage VQE</td>
<td>58</td>
</tr>
<tr>
<td>Visible Human Explorer</td>
<td>63</td>
</tr>
<tr>
<td>VisMeB</td>
<td>4</td>
</tr>
</tbody>
</table>
Development Stages, 163
HTML Mockup, 166
Java Prototypes, 170
Paper-Based Mockup, 164
Web-Based Evaluation, 169
Visual Meta-data Browser, 4
Visual Structures, 8, 9
Visualization Islands, 41
Visualization Reference Model, 11
Voronoi, 40
  Voronoi Cell, 40
  Voronoi Diagram, 40
Voxelplot, 35
VRVis, 31
Waterfall Lifecycle, 155
Wearable Computing, 160
Webwinds, 35
WIMP, 160
Winner-Neuron, 39
Wizard Of Oz, 158
WordPerfect, 60
Workshop, 157
xfIND, 28, 41
XGobi, 58
XmdvTool, 58
XYZPlot, 35