

Visual Analysis of Relatedness and Dynamics in Complex, Enterprise-Scale Repositories

Vedran Sabol

Dissertation submitted to the Graz University of Technology,
Faculty of Computer Science,
for the attainment of the degree of
Doctor of Engineering Sciences (Dr. techn.)

Visual Analysis of Relatedness and Dynamics in Complex, Enterprise-Scale Repositories

submitted by

Vedran Sabol

March 2012



© Copyright 2012 by Vedran Sabol

First Reader: Univ.-Prof. Dr. Stephanie Lindstaedt
Second Reader: Prof. Dr. Michael Granitzer
Advisor: Prof. Dr. Klaus Tochtermann

Abstract

Large data repositories, such as those used in enterprises, typically store many millions of unstructured, human-readable documents, which contain complex, high-dimensional, multi-faceted information. Such repositories exhibit a highly dynamic behavior characterized by continuous adding, removing and modification of the data set elements. This work deals with the development and application of interactive visual techniques supporting exploratory analytical processes in large, complex, dynamically changing data sets. Developed visual techniques and algorithms empower the user to obtain an overview and gain insight into the data set at different levels of detail through organizing and structuring the data according to relatedness. Through unveiling underlying, implicitly present structures, the user can understand the degree of relatedness between these structures, identify features revealing their essence, perceive their size and cohesion, discover anomalies and outliers, and correlate orthogonal facets of the data such as rich metadata. Particular focus is laid on capturing dynamics by revealing changes and trends, and unveiling causal relationships. Developed visual techniques were integrated and combined with automatic processing methods into interactive user interfaces for discovery and analysis of topical-temporal patterns in textual data. Also, to demonstrate the applicability of the developed technologies on other domains and data types, selected techniques were applied on semantic data. Performed research and development was driven by real-world needs from application domains such as business and governmental intelligence, and media analytics. Feedback from early adopters helped improve applicability of the resulting visual techniques, leading to their productive deployment in selected knowledge discovery workflows and use cases. Data collected from the performed usability experiments was analyzed to identify and fix usability issues, and allow statements on performance of the developed technologies.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wortlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Date:

Signature:

Acknowledgements

I want to thank my advisors Stephanie Lindstaedt and especially Michael Granitzer for advice and feedback, continuous support, attention to my questions, and for helping me improve the draft versions of this document. I also wish to thank Klaus Tohtermann for his guidance during early stages of this work. Additional thanks goes to all members of the Know-Center's Knowledge Relationship Discovery area and the KnowMiner team for their assistance in technical aspects, in particular to Wolfgang Kienreich for his mighty rendering engines and a fruitful, almost a decade long collaboration in visualization, Werner Klieber for ensuring the KnowMiner framework is up and running rock-solidly, and Markus Muhr for his fine-tuned, blazingly fast clustering algorithm implementations. I would like to thank everyone at the Know-Center Graz, and the Knowledge Management Institute of the Technical University of Graz, for their friendly cooperation and feedback, and many other colleagues and friends for providing constructive criticism and invaluable help. Last but not the least, many thanks to my family and especially to my Ana, for all the patience and your unconditional, enduring support.

Vedran Sabol
Graz, Austria, March 2012

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Focus	2
1.2.1	Scope	2
1.2.2	Goals and Research Questions	4
1.2.3	Methodology	6
1.3	Contributions	7
1.4	Structure of the Document	9
2	Foundations and Related Work	13
2.1	Principles of Visually Supported Analysis	13
2.1.1	Information Visualization	15
2.1.2	HCI and Usability Evaluation	18
2.1.3	Visual Analytics and Knowledge Discovery	20
2.2	Clustering Techniques	22
2.2.1	Hierarchical Clustering	23
2.2.2	Partitional Methods	24
2.2.2.1	K-Means	24
2.2.3	Scalable Clustering Methods	25
2.2.4	Other Methods	26
2.2.5	Clustering for Browsing Document Collections	26
2.2.6	Faceted Categorization	27
2.3	Ordination Methods	29
2.3.1	Principal Component Analysis	30
2.3.2	Multidimensional Scaling	30
2.3.3	Self Organizing Maps	31

2.3.4	Force-Directed Placement	31
2.3.5	Scalable Ordination Techniques	33
2.4	Visualizing Large, High-dimensional Data Sets	33
2.4.1	Scatterplot	34
2.4.2	Parallel Coordinates	35
2.4.3	Treemaps	36
2.4.4	Cluster Map	38
2.4.5	Information Landscapes	38
2.4.5.1	Bead	40
2.4.5.2	SPIRE	41
2.4.5.3	VxInsight	42
2.4.5.4	Other Approaches	42
2.5	Visualization of Temporal Data	43
2.5.1	Perspective Wall	43
2.5.2	LifeLines	44
2.5.3	ThemeRiver	45
2.5.4	More Temporal Data Visualizations	45
2.6	Other Visualizations	46
2.6.1	Tag Clouds	46
2.6.2	Geovisualization	47
2.6.3	Visualizing Relationships	48
2.7	Coordinated Multiple Views	49
3	Early Contributions	51
3.1	Starting Point	51
3.2	Incremental Visualization of Search Results	53
3.2.1	Goals	53
3.2.2	Concept	54
3.2.3	Incremental Visualization	56
3.2.4	Architecture	57
3.2.5	Integrated Ordination and Clustering Algorithm	60
3.2.6	Evaluation	64
3.2.7	Other Applications	65
3.3	Visualizing Hierarchical Document Collections	68
3.3.1	Requirements	69

3.3.2	Visual Interface	70
3.3.3	Searching and Highlighting	71
3.3.4	Navigation	72
3.3.5	Algorithms	74
3.3.6	Evaluation	75
3.3.6.1	Preliminary Evaluation	76
3.3.6.2	Main Evaluation	78
3.4	Temporal Visualization	81
3.4.1	Temporal and Topical Analysis of Search Results	81
3.4.2	Visualization of Communication Patterns in Meetings	84
3.4.2.1	MISTRAL Overview	85
3.4.2.2	Visual Conversation Analysis Tool	86
4	Algorithms and Visual Techniques	91
4.1	Approach and Overview	91
4.2	Scalable, Incremental Ordination Algorithm	94
4.2.1	Incremental Computation	99
4.2.2	Scalability	101
4.2.3	Visual Evaluation	103
4.3	Landscape3D Visual Component	106
4.3.1	Interactivity and Navigation	107
4.3.2	Visual Property Coding	108
4.3.3	Visualizing Change through Dynamic Topography	109
4.4	StreamView Visual Component	110
4.5	Coordinated Multiple Views Framework	112
4.5.1	Additional Coordinated Components	115
4.6	Knowledge Discovery Visual Environment	116
4.7	Semantic Mediation Tool	117
4.7.1	Ontology Alignment	118
4.7.2	Visual, Semi-Automatic Ontology Alignment with SMT	120
4.7.3	Alignment Algorithm	122

5	Architecture and Implementation	125
5.1	KnowMiner	126
5.1.1	KnowMiner Modular Architecture	127
5.1.2	Knowledge Discovery API	135
5.2	VisTools	135
5.2.1	VisTools Architecture	135
5.2.2	Analytical Components	137
5.2.3	Coordinated Multiple View Framework	139
5.2.4	Algorithms	140
5.3	Integration of VisTools and KnowMiner	141
5.3.1	Prototype Applications	142
6	Case Study	145
6.1	Fused Topical-Temporal Analysis	145
6.1.1	Step 1: Getting an Overview	146
6.1.2	Step 2: Topical Relatedness of Temporal Peaks	148
6.1.3	Step 3: Finding Possible Causes of Temporal Peaks	150
6.1.4	Step 4: Validating the Hypothesis using Faceted Metadata	151
6.1.5	Step 5: Validating the Hypothesis using Retrieval	155
6.2	Semantic Mediation	158
6.2.1	Starting a Mediation Process	158
6.2.2	Collaborative Reviewing	159
6.2.3	Drill Down to Area of Interest	163
6.3	Production Scenarios	167
6.3.1	Business Intelligence	167
7	Evaluation	171
7.1	Testing Methodology and Environment	172
7.2	Topical-Temporal Analysis Experiment	175
7.2.1	Tasks	178
7.2.2	Task Execution Times	180
7.2.3	User Feedback	180
7.2.3.1	Feedback on StreamView	180
7.2.3.2	Feedback on the User Interfaces	181
7.2.3.3	General Impressions	181
7.2.4	Results Discussion	182

7.3	Explorative Analysis with the Information Landscape	184
7.3.1	Navigation Experiment	184
7.3.1.1	Tasks	185
7.3.1.2	Task Execution Times	186
7.3.1.3	User Feedback	188
7.3.1.4	Results Discussion	188
7.3.2	Visual Properties Coding Experiment	189
7.3.2.1	Tasks	190
7.3.2.2	Task Execution Times	192
7.3.2.3	User Feedback	192
7.3.2.4	Results Discussion	193
7.3.3	General Impressions	194
7.4	Summary	194
8	Conclusion	195
8.1	Result Summary	195
8.2	Future Work	198
A	Authored and Co-authored Publications	201
A.1	About the Author	201
A.2	Publication List	202

List of Figures

2.1	Visual representations where preattentive processing is possible (left) and not possible (right) [Healey 2009].	14
2.2	Data vs. information vs. knowledge.	15
2.3	The knowledge discovery chain [Fayyad et al. 1996].	21
2.4	Categories of faceted metadata (left) extracted from text documents and a topical cluster hierarchy (right) for 296 news documents returned for query "IBM PC".	28
2.5	A scatterplot visualizing book metadata: publication year (x-axis), page count (y-axis), file size (icon size), author (icon type).	34
2.6	Parallel coordinates showing nine metadata types on parallel axes for eBooks (note that synthetic data is used).	35
2.7	Treemap showing the file hierarchy on a hard drive.	37
2.8	An information landscape built from several thousand documents on climate change [Sabol et al. 2008c] (see Chapter 4 for details).	39
2.9	An information landscape showing a mix of images and text documents [Lux 2004].	40
2.10	A mock-up illustrating the principle of Perspective Wall (arrows indicates the direction of time).	43
2.11	An example illustrating the idea behind ThemeRiver.	44
2.12	A visualization showing a global tag cloud (central area) and tag clouds for six different categories (surrounding areas).	46
2.13	Geo-visualization of Austria showing locations referenced in news articles. Location references are shown as cones, with the size corresponding to the number of articles referencing that location.	47
2.14	A graph visualization of relationships between concepts extracted from a text data set. Edge bundling improves clarity and reduces clutter in the edge layout. (Data courtesy of German National Library of Economics, 2011.)	48

3.1	Visualisation Islands showing clustering and an information landscape for search results returned for query "visualisation".	52
3.2	Visual query refinement with WebRat: 1) overview for query "Knowledge Management", 2) zoom in on "certification" cluster, 3) Refining the query with the term "Information".	55
3.3	A series of nine snapshots of WebRat incrementally processing about 300 scientific paper abstracts.	56
3.4	The system architecture of WebRat.	59
3.5	Comparing two environmental information sources (in German), UDK Germany and UDK Austria, for the search query "Atom*".	66
3.6	InfoSky user interface consisting of a visualization component, a tree and a table, all showing the top level collections.	70
3.7	Search results for "linux" (magenta), "windows" (green), and "virus" (red).	72
3.8	Navigation the hierarchy collections "Software", "Operating Systems", "Linux" (highlighted with gray overlay).	73
3.9	InfoSky evaluation setup.	76
3.10	InfoSky test conditions: V on the left and T on the right.	77
3.11	OnAir user interface.	82
3.12	Visualization of cluster temporal development (left) and topical relationships (right).	82
3.13	MISTRAL modules.	85
3.14	Visual Communication Analyzer (VCA).	87
3.15	VCA architecture.	89
4.1	Ordination algorithm UML diagram.	100
4.2	Execution times of the ordination and clustering algorithm, in seconds, for data set sizes from 10000 to 100000 documents.	102
4.3	An example with 529 documents returned for query China: A standard landscape is shown on upper left, with data element stress color coding on upper right. On the bottom left is a StressMap for the same data set, with a magnification of local stress phenomenon at the border between two clusters shown on bottom right.	105
4.4	Landscape3D component showing 400000 documents.	106
4.5	Zooming in the hierarchy of clusters. The visualization shows approximately 4000 search results on "terrorism" from the Reuters data set.	107

4.6	Mapping of features and metadata onto visual properties: in this case locations mentioned in the documents are mapped onto color. The query was "computer industry".	108
4.7	A sequence of growing, incrementally computed information landscapes. Documents added in each step are shown as red dots.	110
4.8	StreamView component showing temporal developments of topical clusters for news on "terrorism".	111
4.9	Temporal developments of topical clusters for news on "oil spill", extended by a document group containing the geographic feature "russia" (yellow stream on the bottom).	112
4.10	Coordination of colors and selection using a Landscape3D (top) and StreamView (bottom) components. Shown are 6900 documents for query "space".	114
4.11	Trees showing a topical hierarchy (up-left) and faceted metadata hierarchy (up-right), and a table (bottom) showing document details.	115
4.12	KDVE visual analysis window showing 6900 news documents on space. Document selection (by time: from June to August), document coloring (each topical cluster in different color) and navigation in the hierarchy (location: Cluster 3 shuttle, mars, columbia) are coordinated.	118
4.13	Ontology alignment.	119
4.14	SMT Visualization Window.	121
5.1	A processing chain of comprising of KnowMiner and VisTools functional blocks.	126
5.2	KnowMiner framework modular architecture. Color schema for modules: yellow - crawling and importing, green - semantic enrichment, red - information retrieval, blue - data mining, gray - data management.	127
5.3	Faceted search example: Searching for "computer" returned over 20000 hits, with filtering possible using faceted categories (available are locations, organizations, persons, tags and sources). Selecting organizations "Microsoft", "Intel" and "IBM" reduces the hit count to merely over 100.	133
5.4	RadialView visualizing associated concept search: concepts returned for query "computer" (center) include companies, persons and locations (placed radially around the query term). . .	133

5.5	A hierarchy of topical clusters (on left) and a similarity layout (on right) for about 20000 documents returned by the query "computer".	134
5.6	Classifying documents on "real-time software" into twelve classes representing major IT companies. Each document is assigned to more than one category, with the estimated confidence determining the winning class.	134
5.7	VisTools modular architecture. Color schema for modules: green - coordination of multiple views, blue - visual analysis components (incl. supporting data structures), red - algorithms, yellow - specialized renderers.	136
5.8	Coordinated components are based on the model-view-controller paradigm, extended with coordination-related data and logic.	137
5.9	Integrating KnowMiner and VisTools to create visual analytics applications.	141
6.1	Search for "oil spill" returns 385 news articles, which is too many for manual analysis of the whole data set.	146
6.2	Topical and temporal visualization of all results returned for the query "oil spill".	147
6.3	Selection of two temporal peaks with the time interval selection bar (selected documents are larger in the Landscape). In the Landscape documents from the first peak are predominantly placed at the top of the cluster, while those from the second peak are at the bottom, indicating that the peaks are topically unrelated.	149
6.4	Choosing and reading one document (shown in white) from the first temporal peak (upper row) and one from the second peak (lower row). The first document mentions an oil spill caused by the Russian ship "Nakhodka", the second one an oil spill caused by the Japanese-operated tanker "Diamond Grace".	150
6.5	Displaying faceted metadata category for location "Russia" (in cyan): In the StreamView it correlates strongly with the first temporal peak, not with the second. In the landscape the same can be seen. However, only a subset of documents from the first peak mention "Russia".	153

6.6	Displaying faceted metadata category for location "Japan" (in cyan): Correlates with both temporal peaks, in the StreamView and in the Landscape, making no conclusions possible. The reason being, that "Japan" is the main feature of the topical cluster, which is mentioned in the majority of the cluster's documents.	154
6.7	Searching for "Nakhodka" and showing hits in red. In the landscape correlates predominantly with the first event (top), and mostly not with the second one (bottom).	156
6.8	Searching for "Diamond Grace" and showing hits in red. In the landscape clearly correlates only with the second event (bottom), definitely not with the first one (top).	157
6.9	After logging in the user can choose a pair of ontologies (left) and start the mediation process (right).	158
6.10	Result of mediating two medical ontologies: table of mapping suggestions is on top-left, an information landscape visualizing all concepts from both ontologies is on bottom-left, and two graph visualizations for browsing the ontologies are on right.	159
6.11	Selecting the area "left/right, heart, failure" with the lasso selection tool.	161
6.12	Assign the 10 mappings between selected concepts (enlarged in the information landscape) to user "jim".	161
6.13	Save the newly created task under the name "heart failure".	162
6.14	After logging in as user "jim", only mappings assigned to his task ("heart failure") can be reviewed. A few mapping suggestions have been accepted (green check mark), several others rejected (red cross) in this example.	162
6.15	The administrator can view the progress of tasks assigned to different users. Here, two tasks out of five show progress.	163
6.16	Mediating the ACM and MRDCS classification systems yields 9128 mapping suggestions. A user with focus on computer networks can immediately identify the area "networks, international, network" in the mid-left part of the concept landscape.	164
6.17	Drill-down to the area of interest - computer networks - by following the labels of the topical cluster hierarchy. Concepts are selected using lasso-selection (bottom).	165
6.18	After the drill-down and selection of topic of interest the amount of suggested mappings was narrowed down from 9128 (top) to merely 6 (bottom).	166

6.19	Classifier refinement workflow using a dynamic topography information landscape.	168
7.1	Usability testing environment.	174
7.2	Child table showing 10 sub-clusters, with sibling-similarities being shown for each sub-cluster sorted in descending order. . . .	177
7.3	User interface configuration used for the two explorative analysis experiments.	185
7.4	Icons used for evaluation: disk icons in different colors (on left), icons with different shapes and different colors (on right). . . .	189
7.5	Examples of using colored disks and overlaid colored symbols to convey properties, with the left image showing a single property and the right showing two properties mapped [Krnjic 2008].	190

List of Tables

4.1	Algorithm execution times, in seconds, for increasing data set sizes.	102
7.1	Each user executes the test tasks twice, with the order of user interfaces (U1, U2) and data subsets (D1, D2) given here. . . .	173
7.2	Task execution times, in seconds, with a user interface employing and information landscape for analysis of topical relatedness.	179
7.3	Task execution times, in seconds, with a user interface employing a child table with sorted sibling lists for analysis of topical relatedness.	179
7.4	Improvements in average task execution times achieved by using an interface with a Landscape and StreamView compared to an interface using a table and a StreamView.	180
7.5	Statements used to collect user feedback on the StreamView temporal visualization.	181
7.6	Results of the user feedback on the StreamView temporal visualization.	181
7.7	Statements used to collect user feedback about the two different user interfaces for topical-temporal analysis.	182
7.8	Results of user feedback collected on the two different user interfaces for topical-temporal analysis.	182
7.9	Statements used to collect general user feedback about the visual application for topical-temporal analysis using both visualizations (Landscape and StreamView).	183
7.10	Results of general user feedback on the visual application for topical-temporal analysis using both visualizations (Landscape and StreamView).	183
7.11	Task execution times, in seconds, with the automatic cluster focusing.	186

7.12	Task execution times, in seconds, using manual zooming and panning.	187
7.13	Improvements and deterioration in average task execution times achieved by automatic focusing compared to manual navigation.	187
7.14	Statements used to collect user feedback after performing the navigation experiment.	188
7.15	Results of user feedback collected after performing the navigation experiment.	188
7.16	Task execution times, in seconds, using colored disk icons. . . .	191
7.17	Task execution times, in seconds, using overlaid colored symbols icons.	191
7.18	Differences in average execution time between colored disks and overlaid colored symbols.	191
7.19	Statements used to collect user feedback after performing the the visual property coding experiment.	192
7.20	Results of user feedback collected after performing the visual property coding experiment.	192
7.21	Statements used to collect general user feedback about the visual application after performing both explorative analysis experiments.	193
7.22	Results of general user feedback on the visual application used for explorative analysis experiments.	193

Chapter 1

Introduction

1.1 Motivation

The already enormous amount of information available electronically is growing at a very high rate. An IDC study [IDC 2007] conducted in 2006 estimates that by 2006 the amount of information available in digital form is approximately 161 Exabyte. An updated forecast [IDC 2008] from 2008 revealed that the information growth outpaces the estimates made a year before, and that by 2012 the amount of information will double every 18 months. While tools for finding and retrieving a single or a few relevant pieces of information have been successfully applied in practice, it is clear that when a holistic view on large amount of complex data is needed, scalable analysis techniques considering the entirety of the data set are required.

Knowledge discovery (KD) is a process of automatically processing very large amounts of raw data in order to identify patterns and extract new knowledge [Fayyad et al. 1996]. Knowledge discovery is wide area of research where variety of approaches can be applied to perform the analysis. Typically statistical and machine learning methods are used, however, approaches such as rule-based systems, artificial neuronal networks, natural language analysis and similar may also be employed. Knowledge discovery process typically includes the following steps: data selection and gathering, data pre-processing and cleansing, data transformation, data mining and analysis, presentation and visualization, as well as user feedback. In this process it is the mining step is where patterns are identified and new knowledge is extracted and aggregated from large amounts of raw data. Traditionally, data mining techniques are applied on structured information saved in databases. However, a significant part of the available information is typically present in unstructured or weakly structured form, such as multimedia files or text documents, which

necessitates adequate mining techniques.

Mining methods rely solely on automated machine computation capabilities. Automated analysis techniques have a tendency of being very hungry for computing power, and although computer hardware has experienced enormous speed ups during the past decades, for certain tasks machines still do not come even close to the capabilities of humans. In contrast to automatic analytic methods, visualization techniques make use of the enormously powerful human visual system, which is capable of recognizing patterns and identifying correlations at once, even in huge amounts of data. Humans can quickly see, explore, and understand complex relationships as long as the data is available in a form which is convenient for the eye to process information: the visual representation. This remains true even when the underlying data is incomplete or contains contradictory information to a certain degree.

Visual analytics is defined as the science of designing and applying interactive graphical user interfaces with the aim of facilitating analytical reasoning [NVAC 2005]. It is an interdisciplinary field based on information visualization, knowledge discovery, as well cognitive and perceptual sciences. Visual analytics combines the advantages of visualization techniques with automatic processing by machines to provide means for effectively revealing patterns and trends, and unveiling hidden knowledge in complex data. Interactive visual techniques are an effective enabler for exploratory analysis [Tukey 1977] empowering users to pose and test a hypothesis, provide assessments and quickly derive conclusions. These characteristics make visual analytics effective for gaining insight into and acquiring knowledge about a data set the user is unfamiliar or only partially acquainted with.

1.2 Focus

1.2.1 Scope

The main topic of this work is the development and application of interactive visual techniques for supporting exploratory analytical processes in large, unstructured, complex, dynamic data sets. To narrow down the scope and define the focus more precisely, the characteristics of the targeted data sets are specified as follows:

- Large data set is, in the context of this work, a repository containing up to several million elements, as used in typical organizations (enterprises), whereby data set elements are usually documents, but can also be any other kind of entity such as for example ontological concepts. Therefore,

large scale is understood to be significantly smaller than huge, Web-scale data sets with sizes of up to billions of elements.

- Unstructured, or weakly structured, data lacks, to a large degree, an explicit structure organizing data set as a whole. An example is information present in human readable text documents stored in a file system, as opposed to data stored in a relational databases or spreadsheets.
- Complex information, in the context of this work, consists of data elements characterized by rich metadata and high-dimensionality, i.e. a very large number (thousands) of describing features (or variables, or dimensions), where at the same time these features can be of different types, for example topical, temporal, or geospatial, each type describing a different aspect of the data elements.
- Dynamically changing data sets are characterized by a significant influx of new, removal of old, as well as modification of present data set elements. However, it should be noted that handling of document versioning is not within the scope of this work.

In the past the visual techniques were applied on large, but predominantly static data sets, with the analyses usually focused on a single aspect of the data, for example on similarity in text or gene expression data, or trends in financial (numeric) information. Visual tools for discovery and analysis of complex relationships and correlations, i.e. those taking into account different aspects of the data, such as topical, temporal, or geospatial, in dynamically changing repositories are still underrepresented. Investigation of possibilities for combining visual and automated techniques for analysis of such repositories, with the aim of applying them in real-world, productive knowledge discovery scenarios is an exciting and promising area of research, and it also represents the novelty introduced by this work.

Research performed within this work was on one hand inspired by the creative impulse wishing to improve on existing visual techniques and methods, and on the other hand driven by the real-world needs from application domains such as business and governmental intelligence or media analytics. The applied character of the performed research should be underlined by close cooperation with technology recipients from the industry, where the involvement of early adopters in the development cycle has the purpose of increasing the practical applicability of the resulting visual analysis methods. Developed visual techniques were productively deployed in specific knowledge discovery workflows and use cases.

1.2.2 Goals and Research Questions

The main goals which should be achieved by the developed visual techniques were derived during a discussion process involving (i) users with particular needs in the application domains such as business, governmental and media intelligence, and (ii) on the other side Know-Center's Knowledge Relationship Discovery [KC-KRD 2011] research team which generated ideas, suggestions and proposals how to address these needs through visual analysis and knowledge discovery methods. Visual analysis techniques and algorithms which resulted from this process are targeted at the domain expert users, and should enable them to accomplish the following:

1. Obtain an overview and gain insight into the data set at different levels of detail.
2. Navigate and explore the data set along the implicitly present structure arising from organizing and aggregating data set elements according to their relatedness. To facilitate explorative navigation on that structure the user must also be able to:
 - (a) understand the degree of relatedness between different data set elements and/or aggregated structures, including discovery of anomalies and outliers,
 - (b) perceive the size and cohesion of the discovered structures,
 - (c) identify important features which reveal and describe the essence of the structures and/or data set elements.
3. Reveal trends and changes over time occurring in the data set, including causal relationships and recurring events,
4. Correlate the discovered structures with other, orthogonal facets of the data, such as various metadata or extracted entities.

To summarize, the main goal of this work is to develop visual techniques and methods for discovery and analysis of topical-temporal patterns in textual data and their correlation with rich metadata.

The concept of "relatedness", being central to this work, deserves particular attention. In the context of this work relatedness should be understood as the degree of connectedness of some kind between a pair of entities or, more specifically, the strength of an implicit or explicit relationship of a possibly complex or abstract type, connecting a pair of entities. A few examples:

- topical similarity is probably the most natural measure of relatedness for text documents
- a relationship between text documents arising due to sharing the majority of the authors
- a relation between a scientific organization and an expert due to activity in the related fields of research
- complex relationships, i.e. those simultaneously taking into account different aspects of the data, such as topical, temporal, spatial, etc.

Allowedly, the concept of relatedness as defined above is wide. It would be out of the scope of this work to provide evidence that technologies and methods described in this work can address relatedness in its general sense. Instead, this work focuses on discovery and analysis of patterns arising from topical relatedness in textual data sets. While the developed techniques should, in principle, be applicable to large, high-dimensional, dynamic data sets in general, in the context of this work they are targeted primarily to relatedness analysis in metadata-rich text repositories, such as enterprise content and knowledge management systems, news and media repositories, scientific publication, patent or legal documentation databases, etc. The reason being, that this work was driven by practical problems and requirements in application domains such as business and governmental intelligence, or media analysis, where the importance of analysis of textual data can not be overestimated [Zanasi 2005].

However, to demonstrate the applicability of the developed techniques on data types other than text, a selection of developed techniques is applied on semantic data (ontologies). For this purpose, a proof-of-concept visual application was developed addressing discovery and analysis of patterns which arise from semantic relatedness between concepts from different knowledge bases.

While large scale and high-dimensionality are two challenges commonly perceived as "hard" in visual analytics and knowledge discovery methods, this work also considers a third one which, although common in real-world data sets and applications, has often been given less attention: the fact that in today's fast paced world data sets grow and change at an astonishing rate. Therefore, addressing dynamics in large, high-dimensional data sets is one of the primary concerns of this work. Examples of temporal analysis include the discovery of a series of recurring events occurring in a fixed chronological order, or gaining insight into causal dependencies between events.

Based on the above discussion, the research questions this work addresses are as follows:

1. How can visual analytics techniques, i.e. visual methods combined with automatic processing, be used to achieve the above listed goals?
2. Does the integration of multiple visualization components into a single interactive user interface provide an effective instrument for simultaneously addressing all goals?
3. Can the developed techniques be extended and applied on more than one data type and more than one type of relatedness?

1.2.3 Methodology

To achieve the goals listed above the following methodology was executed:

- Investigation of the available visual analysis tools and related algorithms which are applicable to the proposed application domain, and an analysis of those in order to identify areas offering largest potential for innovation and where the largest impact can be achieved.
- Definition of the application domain(s) and selection of the data sets. Analytical scenarios to be addressed were defined in cooperation with the early adopters.
- Design and implementation of the visual components along the lines of usability engineering.
- In parallel with the design and implementation heuristic evaluation was applied. Improvements were integrated into developed components according to evaluation results.
- At a later stage of development usability evaluation was performed. By running a series of controlled experiments, the performance of the visual components and tools was measured on specific tasks. The collected data was statistically analyzed to identify and fix usability issues, and allow statements on performance of the developed technologies.
- Depending on usability evaluation results improvements were implemented, and the resulting visual tools were applied in productive environments.

The used methodology was conceived to ensure that the design and implementation of the visual techniques are successful in yielding visual components

and tools which are suitable for addressing the targeted goals and subsequently demonstrate that, for selected use cases and problem classes, developed components and tools can provide an improvement compared to traditional, i.e. non-visual, approaches.

It should be noted, that the contributions of this work build upon well-known visual representation, which are developed further through introduction of new ideas and techniques, and refined in detail depending on the results of the usability evaluation studies and feedback collected from early adopters, to achieve practical applicability in real-world scenarios. Note that planing and conceiving completely new visual representations from scratch was not a goal, as this would imply a strong shift of focus towards cognitive and perceptual sciences, which is clearly out of the scope of this work.

1.3 Contributions

Contributions of this work include development of methods, components and tools for visual analysis of large, unstructured, dynamic data sets. Developed techniques were evaluated and deployed in selected knowledge discovery pilot applications.

Several interactive visualization components, each addressing some of the goals defined in the previous section, were developed. These are in particular:

- An information landscape featuring hierarchical space subdivision and dynamic topography: this advanced visualization component is capable conveying relatedness, quantitative distribution, cohesion, and changes, features hierarchical data structuring and navigation, and offers various visual channels for encoding other orthogonal aspects of the data which are typically expressed as rich faceted metadata.
- A stream-view component for visualization of temporal data, including tracking of change, identification of trends, temporal patterns, cyclic phenomena and causal relationships.
- Extensions of standard GUI widgets such as trees and tables.

A framework for combining and integrating various components into a single unified, coherent user interface was implemented along the line of coordinated multiple views paradigm. Through view coordination tight coupling of several visual components was achieved where interactions performed in one component are immediately reflected in all components within the GUI, extending analytical functionality beyond the capabilities of each single visual components.

Developed visual techniques were designed for supporting knowledge discovery workflows and use cases. As such they integrate or are applied on the results of mining algorithms provided by the KnowMiner framework [KnowMiner 2011], such as for example information extraction and named entity recognition, clustering and classification, faceted search and faceted metadata, and others. In this context a prominent position is occupied by a scalable, incremental, hierarchically aggregating ordination algorithm, which was developed with the purpose of providing structures and geometry to the visual components listed above. This algorithm, and its evaluation using visual stress analysis, represents an important contribution of this work.

The development of the visual techniques and algorithms has been performed in collaboration with early adopters who employed the developed technologies for addressing particular real-world knowledge discovery use cases. Feedback collected from pilot users during early stages of the development helped reveal the strengths of the proposed concept, identify weaknesses, generate ideas for improvements, and define the directions of further development.

Built upon the aforementioned technologies a prototypical demonstrator application, the Knowledge Discovery Visual Environment (KDVE), consisting of multiple coordinated visual components was used as proof of concept and testbed. Usability evaluation was performed on the demonstrator including formal experiments, thinking aloud tests, and heuristic evaluation. Following tasks were performed using Reuters Corpus Volume 1 (RCV1) [RCV1 2000], an English language news corpus, to allow statements on the performance of the visual techniques:

- Explorative analysis of temporal-topical relationships through multiple visual components: a user interface including a temporal and topical visualization was compared to a user interface consisting of a temporal visualization and standard GUI widgets.
- Navigation in topical hierarchies and exploration of topical relationships: a complex user interface including visualization components was compared to a user interface consisting of standard GUI widgets (such as trees and tables) only.
- Usage of visual channels for correlating metadata and topical relationships: evaluation of perceptibility of symbols (color only vs. color and icons) used in an information landscape.

Generally speaking the evaluation results did confirm the usefulness of the visual components and the integrated user interface in executing of particu-

lar analytical tasks, but also revealed some substantial usability issues and identified areas where the results were inconclusive.

A second prototype application, the Semantic Mediation Tool (SMT), demonstrates the applicability of visual techniques for data sets other than text. This tool addresses collaborative, semi-automatic alignment of ontologies through a combination of alignment algorithms and a visual user interface consisting of several coordinated visual components.

1.4 Structure of the Document

After defining the goals and outlining the methodology and contributions of the thesis in this chapter, the rest of the document is organized as follows:

Chapter 2 presents state-of-the-art and discusses related work. It introduces central concepts and principles of information visualization and visual analytics, and outlines their role within knowledge discovery processes. A brief survey is given on visual methods providing on overview of large, unstructured data sets, and for visualization of change and temporal data. The chapter also includes a summary on ordination and clustering algorithms. Publications authored or coauthored by me contributing to this chapter include [Sabol et al. 2008a] (journal publication), [Sabol et al. 2008b] (book chapter), and an upcoming book chapter on "Visual Analysis and Knowledge Discovery for Text" accepted for publication (as of December 2011) in "Large-Scale Data Analytics" (to be published by Springer).

Chapter 3 introduces relevant early research performed by the author before the goals and the focus of this work were defined. Some visual techniques and algorithms described in this chapter introduce new ideas, others represent an intermediate state in the development, while others provide fundamentals which were used and extended to achieve the goals listed in Chapter 1.2.2, and which led to the development of the techniques described in the following chapter. Publications authored or coauthored by me contributing to this chapter include but are not limited to: [Andrews et al. 2002] (journal publication), [Sabol et al. 2007], [Kienreich et al. 2005a], [Kienreich et al. 2005b], [Granitzer et al. 2004], [Andrews et al. 2004], [Kappe et al. 2003], [Kienreich et al. 2003b], [Granitzer et al. 2003], [Sabol et al. 2002a] (conference proceedings), and [Andrews et al. 2003] (poster).

In Chapter 4 describes algorithms and visual techniques, providing an expert user with tools for exploratory visual analysis of large, complex, unstructured, dynamically changing repositories. The chapter begins with a description of scalable, incremental aggregation and ordination algorithms used to

automatically extract structure from the data set and transform the data into a representation suitable for visualization. Subsequently, components providing an interactive visual representation of data set are introduced. Finally, an analytical user interface consisting of several visual components built around a coordinated multiple view framework is described. The chapter also discusses the choices made and compares the developed techniques to state of the art. Publications authored or coauthored by me contributing to this chapter include [Granitzer et al. 2010] (journal publication) [Sabol et al. 2010b], [Muhr et al. 2010], [Seifert et al. 2010a], [Sabol et al. 2009a], [Sabol et al. 2008c], [Sabol et al. 2007] (conference proceedings), and [Onn et al. 2011], [Sabol et al. 2009b] (posters).

Chapter 5 describes selected implementation details and the overall software architecture of the developed components and tools. This includes the VisTools library which implements the visual component and the coordination functionality, and the KnowMiner framework which embeds the developed ordination algorithm. KnowMiner, the Knowledge discovery framework developed at the Know-Center over the past decade, also provides numerous other algorithms which, although not developed within this work, provide results used by visual components and tools, and therefore represents an essential part of the overall software architecture. Publications coauthored by me contributing to this chapter include [Klieber et al. 2009a], [Klieber et al. 2006] (conference proceedings).

Chapter 6, demonstrates how the two prototype visual applications can be used to address specific analytical tasks, whereby Knowledge Discovery Visual Environment (KDVE) is applied on metadata-rich text data, while Semantic Mediation Tool (SMT) is applied on ontological concepts. Several examples and simple use cases are presented to illustrate the advantages of the systems are discussed. Also, use case addressed by productive installations employing the developed visual techniques are briefly outlined and discussed.

Chapter 7 concentrates on the algorithms implemented in Visualisation Islands. Design decisions and a detailed description and analysis of the three processing steps for constructing the visualization is given. The algorithms (or classes of algorithms) described are: clustering of the retrieved documents using single-pass, k-means, or hierarchical agglomerative clustering algorithms; mapping the documents from the high-dimensional term space to the 2-D viewport space using a force-directed placement algorithm; generation of a 2-D or 3-D style map background image. Central to this chapter are two Bachelor Theses which I have supervised: [Krnjic 2008] and [Weitlaner 2009].

Chapter 8 provides a summary of the results achieved in this work and also discusses ideas for improvements in areas which have not been addresses

in a satisfactory manner. The chapter concludes with an outlook for future research.

Appendix A briefly introduces the author, provides a list of his publications and gives a short overview of his past research activities. The document concludes with the Bibliography, containing a comprehensive list of relevant scientific literature.

Chapter 2

Foundations and Related Work

This chapter gives a short introduction to information visualization and visual analysis, presents state-of-the-art relevant to this work, and discusses the advantages and disadvantages of related work. After introducing the reader to general principles of information visualization and visual analytics, and their application in the context of knowledge discovery processes, a summary of algorithms used for aggregating data and computing geometry needed by the visualizations is provided. The chapter concludes with a survey on relevant visual techniques, including space filling visual methods for providing an overview and exploration of large, unstructured data sets, and methods for visualization of change and temporal data.

2.1 Principles of Visually Supported Analysis

Visualization in general includes all techniques dealing with creating images or animations for communication of data, information and knowledge. Visualization relies on the human visual perception capabilities, allowing us to process large amounts of information and recognize patterns at once by simply seeing things. Preattentive processing [Treisman 1985] is an illustrative example, allowing humans to process information automatically and without the need for focused attention, in as little as 200 to 250 milliseconds. It requires adequate use of visual features such as color, contrast, size, curvature, size, etc. An example is given in Figure 2.1, where the red circle can be detected immediately in the left image, while in the right image it is possible only through scanning.

Through the use of computer graphics and pervasiveness of computers visualization has expanded into many fields of human activity. Depending

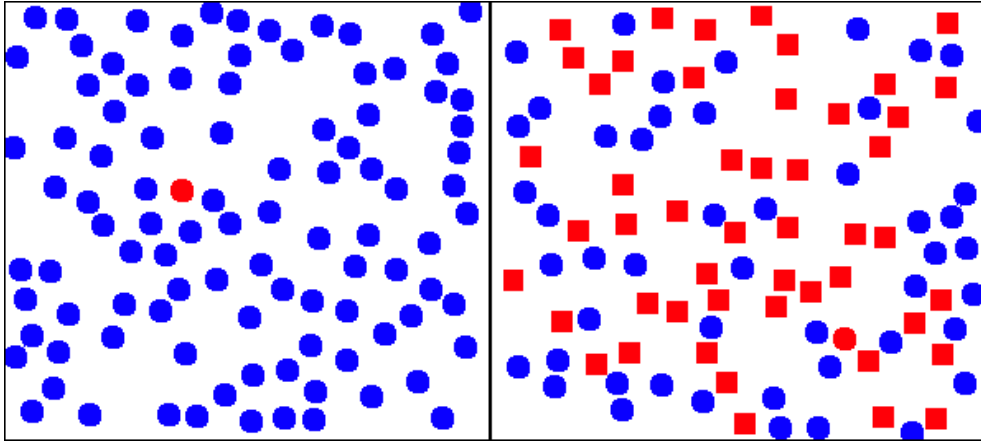


Figure 2.1: Visual representations where preattentive processing is possible (left) and not possible (right) [Healey 2009].

on level of abstraction and complexity (see Figure 2.2) visualization can be subdivided into:

- Data visualization deals with representing data as raw material in particular format. A common example is scientific visualization which graphically represents data which has a natural geometric representation in the real world, such as for example simulation or sensory data, with typical applications being in the fields of physics, medicine, industry etc.
- Informations visualization is about representing abstract information spaces, i.e. information which does not have a natural representation in the real world. Instead, the abstract information, being a result of processing, manipulation and interpretation of data in a given context, is represented in an abstract way suitable for that particular context.
- Knowledge visualization is about communication of knowledge as identified, classified, and as valid recognized information, with visual representations depending on formal, domain-specific models used to represent abstract concepts, facts and conditions.

Note that alternative classifications for visual representations are also available, for example in [Lengler 2007] where authors attempt to classify a large number of different representations into a so-called "periodic table of visualization methods".

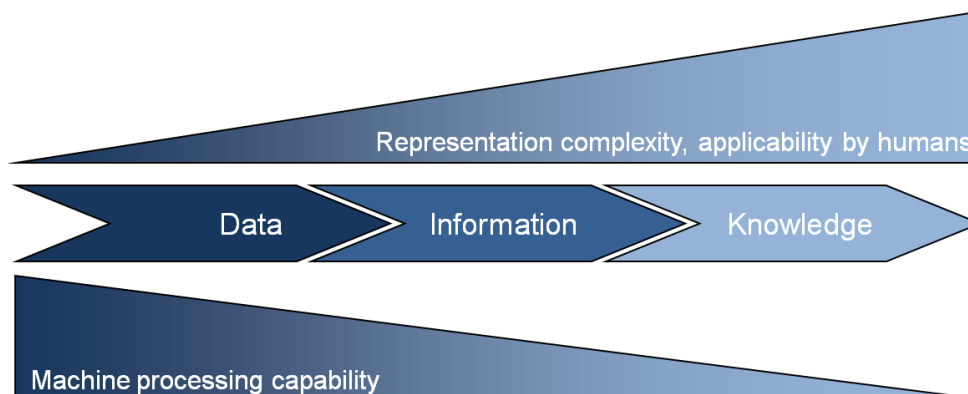


Figure 2.2: Data vs. information vs. knowledge.

2.1.1 Information Visualization

Since information visualization focuses on representing abstract information which has no natural visual representation, the design and geometrical composition of the used visual representation must be adapted to the characteristics of the visualized information and the context in which this information is used and interpreted. Different visual representations can fall into one of the following fundamental categories [Eibl et al. 2001][Nardi & Zamer 1990]:

- Formalisms are abstract schematic representations which typically do not have equivalents in the real-world, requiring the user to learn how to read and use them. The advantage of formalisms is that they have a clearly defined semantics. Simple examples are using arcs to represent percentages or using an x-y plot to visualize dependency between two variables.
- Metaphors visualize abstract information using an well-known equivalent for information which does have a natural representation in the real world. Metaphors are usually more intuitive than formalisms, because the user can infer the meaning through analogy. However, this comes at the price of "fuzzier" semantics, as analogies between two different domains have to be drawn. An example is using a geographic map metaphor for visualizing pools of gene expression data arranged and located depending on relatedness.
- Models are based on internal mental representations of real physical system - the mental models - which can be mentally manipulated to simulate the system and reason about the real world. Therefore, models

present information which has a real-world representation in its natural form, for example using 3D virtual worlds.

Due to the abstract nature of the visualized data, visual representations used in information visualization make use of formalisms and metaphors. Models are used only in special cases, for example geovisualization can be employed when abstract data can be mapped onto geospatial coordinates. Despite a wide variety of available visual representation each with a distinct interaction model, design guidelines for information visualization applications can be summarized in the well-known visual information seeking mantra: "overview first - zoom and filter - details on demand" [Shneiderman 1996]. When designing a visual representation decisions have to be made concerning the mapping of data set (logical) attributes to the visual attributes of the items in the visualization. Some often used visual coding principles are:

- Size coding: Size is used to indicate quantitative, numeric information such as a length of a document or a number of elements within a collection.
- Color coding: Using different colors is good for coding discrete (nominal) values, such as the type of an object. Color is also used to show quantitative information, but faces problems when several colors are mixed. A special case of color coding is transparency coding.
- Brightness coding: Brightness is suitable for numeric information. It is similar to size coding, but has the advantage that it can be applied when the space is limited.
- Shape coding: Shapes are typically used to display non-numerical, discrete (nominal) information. Shapes can be purely geometric structures or metaphors derived from real world objects. As metaphors influence the mental model of the user care must be taken to use match visual metaphors to properties of the data.
- Proximity coding: The distance between objects in the visualization can be used to code numeric relationships in the data, for example in graphs connected elements will be placed closer to each other than the unconnected.
- Position coding: A position of an object relative to an axis is usually dependent on some numerical value. Nominal values can also be coded this way when the order of the attributes which are placed on the axis is defined.

Besides the visual design the interactivity of a visual component is of its central properties. A visualization must offer the possibility to intuitively navigate the information space, manipulate the displayed objects, request additional information, and perform operations specific to that particular component. Typical interactions offered by interactive visual components are:

1. **Zooming:** Zooming in displays a smaller region of the currently displayed for a more detailed display. Zoom out is the opposite operation, changing the view from a smaller space area to a large one decreasing the level of detail.
2. **Panning:** When the visualization space is larger than the available screen area (smooth) motion from one part of the visualized space to another reveals information which was previously outside of the displayed screen area.
3. **Selection:** This operation chooses a set of displayed objects so that some other operation can be applied on them.
4. **Dragging:** Moving a group of (selected) objects so that their position is changed compared to the rest of the visualized objects. Dragging usually triggers another operation specific to the involved visual component.
5. **Filtering:** The operation of removing, temporarily or permanently, uninteresting elements from the display, according to a specified criteria.
6. **Pointing to:** By pointing, for example with the mouse pointer, a visualized object is declared to be in the focus of user's interest, usually to reveal additional information on the object.

Although the field of information visualization is still relatively new, there is already a large number of various visual representation and components each targeting different characteristics of the data or different user needs. Various schemata for classifying different visual representations and visualization systems were proposed. In [Shneiderman 1996] a taxonomy was proposed depending on seven data types (1-, 2-, 3-dimensional data, temporal and multi-dimensional data, and tree and network data) and seven user tasks (overview, zoom, filter, details-on-demand, relate, history, and extract). In another example [Andrews 2010a] visual representations can be classified depending on the characteristics of the data into those visualizing:

- linear structures
- hierarchies

- networks and graphs
- multidimensional metadata
- high-dimensional feature spaces (such as text)
- query spaces

For further reading on information visualization up-to-date information can be found in [Ware 2004], [Chen 2006] and [Spence 2007]. For a wider, more universal approach spanning the fields of psychology, perception, and cognition, over art and design, to engineering disciplines of visualization, computer graphics and user interfaces [Ware 2004] and [Tufte 1990] can be recommended.

2.1.2 HCI and Usability Evaluation

Human-computer interaction (HCI) is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. [Hewett 1992]. It is a field at the intersection of computer science, design, behavioral and perceptual sciences with the goal of providing usable interactive computer interfaces. Usability is a measure for ease of use and effortlessness of learning how to use a system, or more formally 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 [ISO 1998]. These three attributes of usability can be objectively and subjectively measured:

- Effectiveness is the accuracy and completeness with which users can achieve specified goals.
- Efficiency is the amount of resources expended to achieve the specified goals in relation to effectiveness.
- Satisfaction describes freedom from discomfort, and positive attitudes towards the use of a software product.

Usability engineering focuses on practical methods for building usable user interfaces. It is an process in which a user interface is prototyped according to general and project-specific guidelines and principles, and then tested and evaluated to uncover and remove the deficiencies. The process is repeated iteratively to further refine and tune the interface. At the heart of this process are usability evaluation activities aiming to uncover usability issues and measure the performance of the user interface. Depending on the goal which should be achieved usability evaluation methods can be subdivided into [Andrews 2010b]:

- Exploratory evaluation, which is performed before or during early design stages, explores current usage and space for potential new ideas and designs.
- Predictive evaluation, performed in late design stages and before implementation starts, delivers estimates on the quality of the future interface.
- Formative evaluation, which is usually performed during design and early prototyping phases, focuses on collecting qualitative observations on what could/does go wrong and why, with the goal of improving the user interface.
- Summative evaluation, which is performed in the later implementation stages, collects quantitative measurements such as duration of tasks or error count, in order to assess and compare the objective performance of the user interface.

Usability evaluation techniques are also differentiated depending on who is involved in the evaluation:

- Usability inspection methods involve usability experts only, who apply heuristics and judgment in performing the evaluation. One example is the heuristic evaluation, where several evaluators inspect an interface design against general principles and produce a list of potential issues. Another example is the cognitive walkthrough, which focuses on learnability, where a team of experts executes a task in the mind set of a novice user to estimate the chances the user would successfully execute throughout various steps of the tasks. Both examples are formative evaluation techniques.
- Usability testing methods are about empirical testing of user interfaces with real, representative users. An example is the thinking aloud test, a formative method where users comment all their actions in a task with verbal descriptions of what they are seeing, thinking, doing and expecting. Another example are formal experiments, a summative method where the performance of an implemented design is objectively measured and/or compared to alternative designs.

The importance of usability engineering, and in particular usability evaluation, for delivery of interactive visual interfaces can not be overestimated. This is especially true when innovative, advanced techniques, such as visualization, are involved. For further reading on HCI and usability-centered design of computer user interfaces see [Shneiderman et al. 2009].

2.1.3 Visual Analytics and Knowledge Discovery

Visual analytics is defined as the science of designing and applying interactive graphical user interfaces with the aim of facilitating analytical reasoning [NVAC 2005]. It is an interdisciplinary field based on information visualization, knowledge discovery, cognitive and perceptual sciences, which combines the advantages of visualization techniques with automatic processing by the machines. While information visualization techniques were successful at providing visual representation of large collections of abstract information, data mining algorithms approach the analysis of large data sets from the purely automatic point of view. Combination of visual and automated analysis techniques has been discussed by several authors, such as in [Shneiderman 2002] and [Keim et al. 2008]. Visual analytics strives to achieve a tight integration between humans and computers, where the computer performs the automatic analysis and presents the results in visual form, while the humans steer the process by interacting with the visualization and providing feedback. An advantage of the tight integration of visual methods with automatic processing is the opportunity to interchangeably apply visual and automatic techniques, as needed by the user. Also, user feedback can be utilized to adjust and improve the models used by automatic methods. This extended scope of visual analytics over information visualization has been outlined by extending the well-known information visualization mantra which becomes the new visual analytics mantra: "analyze first - show the important - zoom, filter and analyze further (iteratively) - details on demand" [Keim et al. 2008b]. Advances introduced by visual analytics approach have implications on the targeted data set characteristics. While information visualization is typically applied on large, predominantly static, homogeneous data sets, visual analytics methods focus on huge, dynamically changing, heterogeneous repositories containing complex, incomplete, ambiguous and conflicting information.

The motivation behind applying visual analysis techniques is that when massive amounts of complex information are transformed by machines into a form convenient for visual representation, human eye's wide pathway into the brain allows users to quickly understand and efficiently explore complex patterns, and immediately apply their knowledge and creativity. Analytical processes in large, dynamic, complex data sets, which take advantage of the powerful human visual system backed by automated processing, enable users to see and recognize patterns, identify correlations, perform on-demand analysis, and gain insight into complex relationships. Besides applying human perceptual abilities on large data sets, the involvement of humans in the analytical process has several other advantages: humans can deal with noisy, ambiguous, conflicting or incomplete data easily, can apply explorative examination

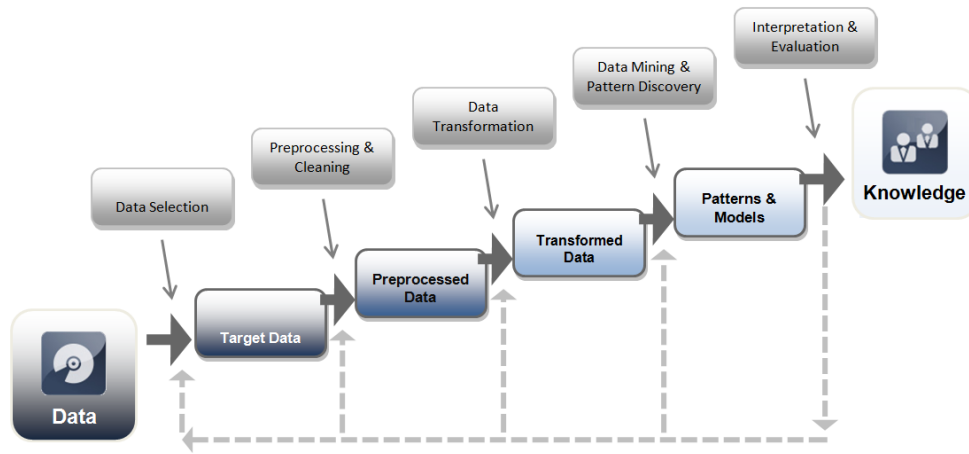


Figure 2.3: The knowledge discovery chain [Fayyad et al. 1996].

even when the data is poorly understood or the goals are vaguely defined, and are capable of adjusting their strategies and goals on-the-fly based on their experience and intuition.

A particularly interesting aspect of visual analysis workflows is the support for generation and validation of hypothesis. Visual representations allow humans to see patterns for which users, equipped with their knowledge and intuition, may come up with possible causes and explanations - which is a potentially valid hypothesis about the data. Visual analysis methods provide means for validation of such hypotheses, typically through a combination of visual and automatic methods. The generation of hypotheses and their validation through on-demand analysis, produce new insights and facts which steer the further direction of the analytical process. Analytical reasoning is supported based on the revealed patterns and confirmed (or rejected) hypotheses, resulting in a process which empowers users to provide assessments, derive conclusions, and communicate the knowledge they have acquired.

Knowledge discovery is defined as the overall, nontrivial process of identifying valid, novel, potentially useful, and understandable patterns in data [Fayyad et al. 1996], where the phrase knowledge discovery emphasizing that higher-level knowledge is the end product discovered from the raw data. The field stands at the crossroads of various research fields, most notably machine learning, pattern recognition, statistics, artificial intelligence, visualization, and high-performance computing. Knowledge discovery is a process consisting of several steps (see Figure 2.3) which are all essential to ensure that useful knowledge is identified in the data. Applying visual techniques in knowledge discovery scenarios implies combining the machine processing power with

the human visual processing capabilities to effectively reveal trends and patterns, and to unveil evidence based on complex data. Therefore, visual analytics can be also be seen as a further development of the classical information visualization put into the context of knowledge discovery. In the context of knowledge discovery process the two steps which are of particular concern for visual analytics are the data mining step, where specific algorithms for identification of patterns and extraction of knowledge are executed, and the interpretation/evaluation step which is a natural fit for application of visual techniques.

2.2 Clustering Techniques

Clustering is a name for unsupervised learning techniques which identify groups, or clusters, of related objects from unorganized, unstructured collections. Objects assigned to the same cluster are similar according to a certain criteria, while objects from different clusters have a smaller degree of relatedness. To apply a clustering algorithm on a data set, there must be a way to compute the relatedness between any pair of objects (including data set elements and clusters). This is accomplished through a definition of a similarity measure, usually the cosine coefficient, or a distance function, typically Euclidean distance (see [Cha 2007] for a survey on similarity and distance measures). Clustering (an unsupervised learning technique) is similar to classification (a supervised learning technique) as both deal with assigning related objects into buckets, the difference being that in classification the categories are defined a priori and must be learned before processing, while in clustering the categories are created dynamically during the computation.

There is a wide variety of clustering algorithms with a different features and performance characteristics targeting various types of data and application domains. Extensive surveys of the field are available in [Berkhin 2002], [Xu et al. 2005] and [Jain et al. 1999]. While different authors propose a variety of (extensive) taxonomies for classifying clustering algorithms, from the perspective of this work an overview of important clustering algorithm characteristics can be summarized as follows:

- **Partitional vs. hierarchical:** Clustering methods are usually divided depending on the the structure which they generate, which may be a single level of clusters or a hierarchy of cluster.
- **Exclusive vs. overlapping:** In exclusive methods each object is assigned to only one cluster, where the overlapping strategy allows multiple assignments.

- Fuzzy vs. hard: Fuzzy methods, being overlapping in their nature, assign objects to clusters with a degree of membership between 0.0 and 1.0. In hard clusters members of a cluster have a degree of membership exactly equal to 1, while non-members exactly equal to 0.
- Deterministic vs. stochastic: Deterministic methods produce exactly the same results if executed with the same starting conditions. Stochastic techniques produce different results each time.
- Order sensitive vs. order insensitive methods: Order sensitive methods generate results which depend on the order of objects in the initial collection. In order insensitive methods the order of items does not play a role.
- High-dimensional vs. low-dimensional: Clustering methods which excel at low dimensional data may produce bad results with high-dimensional data, and vice-versa.
- Incremental vs. non-incremental: Incremental methods allow adding and removing of objects to an existing clustering result, which is modified and adapted accordingly to the change. Non-incremental methods require a re-computation from scratch which may (and likely will) yield a completely different result. .
- Scalable vs. non-scalable: Non-scalable techniques may produce good results with small data sets, but the running time becomes prohibitively expensive for larger data sets. Scalable techniques can handle larger data sets with acceptable running times.

From the point of view of knowledge discovery and visual analytics, and this work in particular, scalability (up to millions of documents), ability to handle very high-dimensional data (thousands of dimensions), and the possibility to deal with dynamically changing data sets are of paramount importance. The rest of this section provides a brief overview of common clustering approaches and their applications relevant to this work.

2.2.1 Hierarchical Clustering

Hierarchical cluster algorithms build a cluster hierarchy, either bottom-up (agglomerative clustering) starting with one object-clusters and recursively merging the most similar pairs, or top-down (divisive clustering) where a single cluster containing all objects is recursively split, until a stop condition, usually the targeted number of clusters, is satisfied. Depending on how the

similarity between clusters is computed in agglomerative clustering, several linkage strategies exist, the most common being:

- Single-link method defines the similarity between two clusters as the similarity between the two most similar members, where each member is from a different cluster.
- Complete-Link method defines the similarity between two clusters is defined as the minimum off all pairwise similarities between members of the two clusters.
- Average-link method computes the similarity between two clusters as an average pairwise similarity between their members.

While a simple implementation of hierarchical agglomerative clustering will have a time complexity of $O(N^3)$, optimized implementations targeting a particular linking strategy reach a time complexity of $O(N^2)$ (single-link [Sibson 1973], complete-link [Defays 1977], average-link [Voorhees 1986]). However, quadratic time complexity makes them unsuitable for clustering of larger data sets. More details on hierarchical clustering algorithms and on their application on high dimensional data, such as text, are available in [Voorhees 1986], [Kaufmann & Rousseeuw 1990], and [Zhao & Karypis 2002].

2.2.2 Partitional Methods

Partitioning clustering methods, given an initial partition, iteratively optimize clusters by relocation data objects from one cluster to another. The iterative process converges to a local or global minimum of an optimization criterion. Depending on the model which is optimized partitional algorithms are be grouped into probabilistic methods, which optimize a probabilistic model (see [Dempster et al. 1977]), and k-means method [Hartigan & Wong 1979], which optimizes a dissimilarity or similarity function, such as the sum of squared distances between each cluster member and the cluster. The algorithm stops the the maximum number of iterations has been reached or when the shift in cluster centroids falls below a specified threshold.

2.2.2.1 K-Means

K-means is one of the (if not the one) most widely used clustering methods. The main advantages of k-means are its time complexity of $O(KN)$ which, provided the number of clusters K is constant, scales linearly with the size of the data set N , and its capability to handle high-dimensional data. This

was demonstrated in an evaluation [Zhao & Karypis 2002] where k-means, and bisecting k-means in particular, outperformed hierarchical clustering on text data. The main disadvantage of k-means is that must be initialized by a set of initial cluster centroids, so called seeds, and the quality of the local minimum the algorithm converges to depends strongly on this initial configuration. There are several ways to address this problem in the praxis:

- Running the algorithm several times with different random initial configurations and choosing the best result. The downside of this solution are much higher running times.
- Positioning the seeds uniformly by finding seed with largest possible distances between them. This is a simple and common way to address the problem.
- Running a hierarchical agglomerative clustering on a sample of the original set and use the results as seeds. Buckshot and fractionation algorithms are examples of this strategy [Cutting et al. 1992].

The ISODATA algorithm [Tou & Gonzales 1974] is a modification of k-means incorporating a number of heuristic procedures for splitting, combining, and discarding clusters to guess the number of clusters and obtain an optimal cluster set. At the beginning of every iteration various statistical measures are evaluated, for example large or non-cohesive clusters will be split into two smaller ones, or if two small clusters are similar they will be merged into a single one. These cluster splitting and merging strategies, combined with the inherent capability of k-means to refine the partition when new objects are added to or old objects are removed from the data set, make k-means a good candidate for an incremental clustering algorithm.

2.2.3 Scalable Clustering Methods

When applied on huge document repositories, containing many millions data items, clustering algorithms face scalability problems both in terms of running time and memory consumption. Given existing methods with linear or slightly super-linear time complexity, the memory problem poses a more acute problem. With huge data sets it is hardly possible to keep the whole data set in the main memory, so clustering algorithms must operate on a subset of the whole data set, which is the data sampling approach, or on a compressed representation of data, which is the data squashing approach. Data sampling approaches, such as CURE [Guha et al. 1998], rely on a relatively small but reliable, representative sample of the data set which is kept in memory for

main clustering, while the rest of the data set is inserted later. Data squashing methods, such as BIRCH [Zhang et al. 1996], perform clustering by sweeping over the data set on disk to create a compressed representation in memory.

Another problem faced by clustering methods is the so-called curse of the dimensionality: in high-dimensional spaces the distance from an object to its nearest neighbor becomes indistinguishably small from the distance to the majority of other data objects. The capability of coping with high dimensionality differs strongly from one clustering algorithm to the other. Dimensionality reduction techniques, such as PCA [Jolliffe 2002], may be applied to alleviate the problem.

2.2.4 Other Methods

In density-based methods, such as DBSCAN [Ester et al. 1996] and DENCLUE [Hinneburg & Leim 1998], clusters are defined as a connected, dense regions which grow in directions where the density leads. Main advantages of density based clusters are handling of noise, and that generated clusters may be arbitrary shaped, whereas clusters found by other methods are mostly of convex, hyperspherical shape. Grid-based methods, such as STING [Wang et al. 1997] and CLIQUE [Agrawal et al. 1998], partition the feature space into segments (cubes or cells), and the clustering is performed by merging adjacent cells which have density above a certain threshold. Grid-based methods are usually fast and scale well both with data set size and dimensionality, but cluster shapes are always limited to union of grid cells which may degrade accuracy. Cluster ensembles, see [Hu & Yoo 2004], are based on the fact that different clustering methods have different advantages and disadvantages. They attempt to combine the results of different methods in such a way to exploit their positive features while suppressing the effects of negative ones.

2.2.5 Clustering for Browsing Document Collections

A hierarchically organized document collection can be explored using a metaphor of a conventional textbook: the table of contents is suitable for getting an overview and for providing information on what is contained in the data set. A cluster hierarchy is a tree whose internal nodes (i.e. sub-clusters) are subtrees of hierarchies and whose leaves are single documents. Similar documents will have a common ancestor lower in the tree, while less similar documents will have a common ancestor further up in the hierarchy (i.e. closer to the root). Clusters (including internal clusters) are represented by textual summaries, for example by keyword or titles of the underlying documents. The

hierarchy of clusters allows the user to navigate in the data set and view the collection at different levels of detail.

Scatter/Gather [Cutting et al. 1993] is a clustering-based document browsing system which computes and presents a cluster-based, dynamic table of contents to the user. Documents are grouped into topically coherent clusters and labeled by descriptive textual summaries consisting of topical keywords and titles which represent the contents of the cluster. The hierarchy is generated by recursively applying a k-means algorithm on the document set. The k-means issue of initial partition sensitivity is addressed by a strategy including sophisticated sampling and hierarchical clustering (fractionation algorithm). Through balancing of the hierarchy and an upper limit of number of clusters on each hierarchy level a near linear time complexity can be reached ($O(N \log N)$, N being the size of the data set). Guided by cluster summaries the user can select several clusters or documents for further study and re-cluster those using a faster but less effective clustering algorithm, the buckshot algorithm, which uses a simpler sampling schema. Buckshot algorithm simply takes a random sample of size \sqrt{N} , clusters the sample with a hierarchical agglomerative clusterer (group average linkage), and uses the clusters as seeds for the subsequent k-means clustering. The procedure of scattering and gathering of clusters and documents can be repeated to refine the groups and to pin down topics and documents of interest.

2.2.6 Faceted Categorization

Clustering has several advantages for exploring a data set, for example it is useful for showing dominant topical groups. In information retrieval it can be applied for disambiguating ambiguous queries and eliminating outliers. However, clustering has several disadvantages such as suffering from the lack of predictability or the difficulty of finding descriptive labels for the clusters. Hierarchical faceted categories introduce an alternative system for grouping documents which introduces meaningful labels with well-defined semantics organized to reflect concepts relevant to a particular application [Hearst 2006]. Assignments to hierarchical faceted categories can be created manually or automatically by classifying documents to categories using classification techniques.

Rich semantic metadata provide another application possibility for hierarchically organized faceted categories. Metadata is assigned manually (for example creation date or author) or extracted automatically using semantic enrichment methods. Automatic enrichment can be achieved by extracting domain-specific semantics from document content and enriching it with knowl-

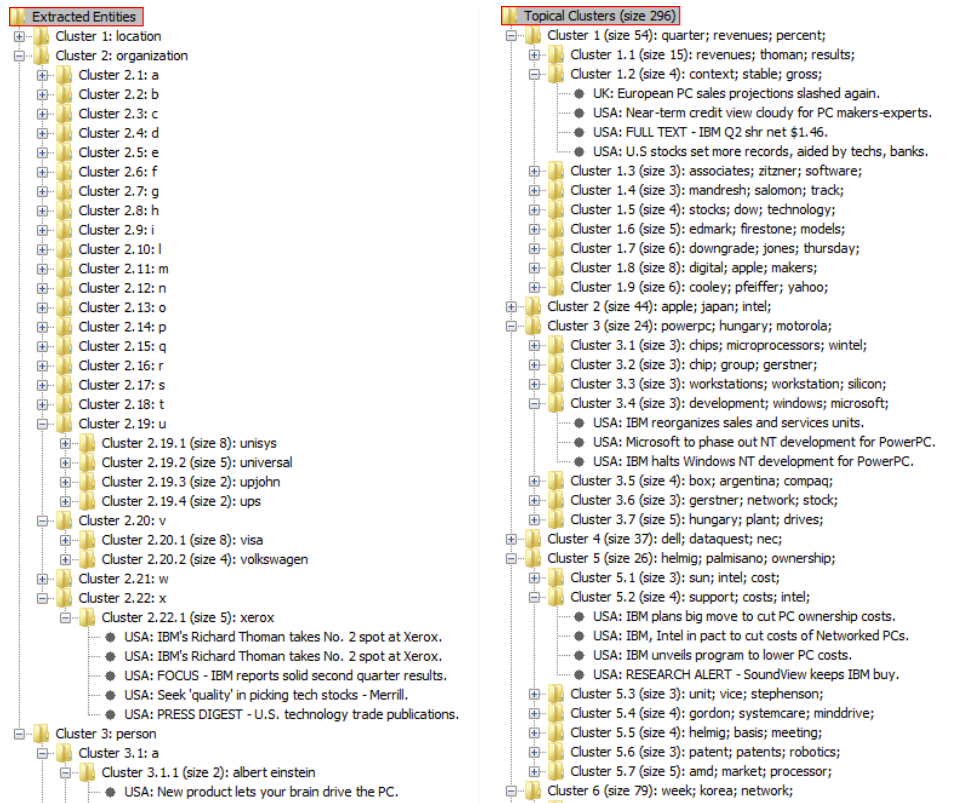


Figure 2.4: Categories of faceted metadata (left) extracted from text documents and a topical cluster hierarchy (right) for 296 news documents returned for query "IBM PC".

edge present in external knowledge bases such as thesauri or ontologies. Information extraction (IE) [Kaiser & Miksch 2005] methods deal with extracting structured information from unstructured or weakly structured text documents using natural language processing methods. IE techniques annotate text by decomposing it into basic building blocks (such as tokens and sentences), performing part-of-speech tagging (i.e. identification of nouns, noun phrases, verbs, etc.), extracting named entities (such as persons or locations) and other metadata, and assigning a well-defined semantics to the extracted information. Selected annotations can be used to extract metadata which are assigned to documents.

In Figure 2.4 a comparison between hierarchically organized faceted metadata categories (on left) and a hierarchy of topical clusters (on right) can be seen for 296 documents on "IBM PC". In the faceted hierarchy the user can see which metadata categories are available for the data set (in this case locations,

organizations and persons), which particular metadata values are mentioned in the documents, and then select documents containing a particular exact value. For example by following "Cluster 2: organization", "Cluster 2.22: x", "Cluster 2.22.1: Xerox" one can see that a person, who is a high-ranked IBM official, moved to another organization, Xerox. The topical cluster hierarchy provides automatically computed labels, which are more numerous and more "fuzzy" than the faceted hierarchy labels, so the user will have to make an additional effort for reading and interpreting labels in the particular context. However, it is possible to discover facts which are not connected to explicitly available metadata. For example, although no technical metadata describing CPUs and operating systems is available, by following "Cluster 3: powerpc, hungary, motorola", "Cluster 3.4: development, windows, microsoft" the user can discover that Microsoft and IBM have stopped the development of Windows NT for PowerPC systems.

2.3 Ordination Methods

Visual systems dealing with complex, high-dimensional data sets need a way to present complex relationships present in high-dimensional space in a 2-D or 3-D visualization space while preserving the original relations as much as possible. Ordination is an umbrella term for methods which position high-dimensional objects in a low dimensional space so that similar objects are placed near each other while dissimilar objects are farther from each other. Ordination methods are targeted at exploratory data analysis and as such are a natural match for visual analysis methods. Ordination methods can be seen as a subset within a broader concept of dimensionality reduction techniques. As its name implies dimensionality reduction also deals with reducing the number of dimensions in a high-dimensional feature space, either through elimination of features (feature selection) or by transforming the original space into a lower dimensional one consisting of new features (feature extraction). While ordination targets exploratory analysis through visualization, dimensionality reduction encompasses methods targeting a broader set of goals, typically addressing the previously mentioned curse of the dimensionality which poses a problem for many algorithms. An overview of dimensionality reduction techniques can be found in [Fodor 2002].

To express the goodness of fit, i.e. the quality of a result produced by an ordination method a the so-called stress value is computed. Stress expresses the degree to which the distances in the low-dimensional space differ from the original distances in the high-dimensional space. High stress values mean that a particular result poorly reproduces the original distances, while lower

stress values imply a better fit. Several different stress definitions have been proposed, the most common one being the sum of squared pairwise distance differences.

This rest of this chapter gives a short overview of ordination techniques, and while they all share the same principle - neighbors in the high-dimensional space must remain neighbors in the low-dimensional, non-neighbors are placed far apart - the focus is on methods addressing large, high-dimensional data sets.

2.3.1 Principal Component Analysis

Principal Component Analysis (PCA) [Jolliffe 2002] transforms the original high-dimensional space into a space spanned by a coordinated system such that for its axes, called principal components, the following holds: the first dimension of the new space has the direction of the highest variance in data, and each consecutive principal component, while being orthogonal to all the previous ones, again has the highest possible variance in the data. PCA is one of the most widely used dimensionality reduction techniques. It has been applied both in addressing the dimensionality curse and in exploratory analysis scenarios involving visualization. For visual applications the first two or first three principal components are taken yielding a 2D or 3D layout. PCA is usually computed by eigenvalue decomposition which is quite compute intensive and does not scale very well to large data sets.

2.3.2 Multidimensional Scaling

Multidimensional scaling (MDS) [Kruskal 1978], [Berry & Groenen 2010] is a name for ordination techniques which takes a distance or similarity matrix as input, instead of considering high-dimensional space coordinates and vectors directly. Pairwise similarity or distance values in the matrix are computed by applying a similarity or distance coefficient, such as cosine similarity or the Euclidean distance. Given the matrix and the dimensionality of the target space, a multidimensional scaling method uses a function minimization algorithm to place data objects in the low-dimensional space in such a way, that the goodness of fit is maximized and the original high-dimensional distances are preserved as far as possible. MDS methods can be realized in different ways for example by using eigenvalue decomposition, or by employing heuristic iterative methods which converge towards a (local) stress minimum, such as force-directed placement (see below: force-directed placement). The former has the advantage that it finds a global stress minimum, but the ordinations tend to be very tightly clustered which is not adequate for interactive visualiza-

tion, while the latter group of methods tends to get stuck in a local minimum, but can be tuned to generate ordinations adequate for appealing, usable visual applications. To circumvent the disadvantages and utilize the advantages of both groups, an eigenvalue decomposition methods can be used to produce an initial layout which is then iteratively refined with a heuristic method to produce a visually acceptable layout [Davidson et al. 1998].

2.3.3 Self Organizing Maps

Artificial neural networks (ANN), similar to their biological counterparts, have the ability to learn by adaptively readjusting their interconnection weights. Data objects represented by high-dimensional vectors are presented at the input nodes which are associated with the output nodes over the weighted network interconnections. Weights between input and output nodes are iteratively adapted in a learning process until some termination criterion is satisfied. Self organizing map (SOM) [Kohonen 1988] is an artificial neuronal network composed of interconnected input and output layers two node layers, where the output nodes are ordered as a 2D matrix (or a hexagonal grid) forming a topological map with bounded regions. Projection of the high-dimensional input vectors is performed by assigning them to the output nodes in such a way that vectors which are neighbors in the high-dimensional space will remain close in the 2D topography. While SOMs can, generally speaking, handle large high-dimensional data sets such as document collection [Kohonen et al. 2000], their learning process is comparatively slow. The computationally intensive procedure requiring many iterations to complete, and is at the same time sensitive to initial weights and several parameters, which can cause a suboptimal projection or compromise convergence. This makes SOMs impractical for agile interactive scenarios where results must be produced quickly or on-the-fly.

2.3.4 Force-Directed Placement

Force-directed placement (FDP) [Fruchterman & Reingold 1991] is a widely used iterative technique based on a spring model, for creating aesthetically pleasing graph layouts. It can be used as a heuristic multidimensional scaling method, where the similarity or distance matrix is interpreted as an adjacency matrix of a full weighted graph. The method employs a physical simulation of a mechanical system, where data set elements are represented by masses connected to each other as springs. Each pair of masses exhibits attractive and repulsive forces on each other, depending on whether their distance in low-dimensional space is larger or smaller, respectively, than their distance

in the high-dimensional space. Masses are iteratively moved depending on the resultant force exhibited by other masses, until these forces fade to zero and the system reaches a minimal energy state. Iteration terminates when the improvement of stress between two iterations falls below a threshold, or even simpler, when the movement of particle falls below a certain rate. Force-directed placement reproduces the high-dimensional relationships well in 2D or 3D spaces, and at the same time is flexible enough to allow for adjustment and fine tuning resulting in aesthetically pleasing, visual appealing, usable layouts. Another advantage of FDP is that it is inherently incremental. When a layout data set is modified, i.e. items are added or removed, changes can be smoothly incorporated into the old layout without excessive disruptions and with a fraction of computation effort necessary for a full recompute. This is not the case for methods based on eigenvalue decomposition, which require a full recomputation from scratch, and which may yield a completely different result even for a moderate modification of the data set.

One drawback of FDP is that the algorithm has a tendency of getting stuck in a local minimum, especially for larger data sets. There are a number of strategies for ameliorating this issue, for example adding a certain amount of jitter to a stabilized configuration may shake it out of a local minimum and let it converge to a state equal, or closer to the global energy minimum. Another approach is the barrier breaking where an object with a low resultant force can tunnel through its low-dimensional neighborhood when its neighboring objects exert a very high repulsive force on it. However, the main problem with FDP is that it does not scale well, because each of N objects must be compared to $N - 1$ objects to compute its next position. This gives a time complexity of $O(N^2)$ per iteration and, if we assume that $O(N)$ iterations are required to reach a stable layout, an $O(N^3)$ time complexity results for the whole algorithm. An efficient optimization of the n-body problem known from physics is the Barnes-Hut method [Barnes & Hut 1986], which recursively subdivides the space into an octree (for 3D space, or a quad-tree for 2D space) and compares each particle only to particles from the same cell and to octree cells which represent all particles contained within them. However, such optimizations are not applicable to high-dimensional data sets, because as particles move around, the updating of cells in the octree hierarchy becomes excessively expensive due to a very large number of features carried by each particle. An alternative approach using stochastic sampling [Chalmers 1996] was found to have lower running times and was applied for visualization of very high-dimensional data sets (text repositories). It preserves the advantages of the force based model, while having a linear execution time per iteration achieved by considering only a constant size sample, instead of the whole

data set, when calculating the force on each object. Every object maintains two sets of constant size: a random set, which is randomly selected from the whole data set in each iteration, and a neighbor set which emphasizes the high-dimensional neighborhood of the object. Neighbor set is refined in every iteration by including random set elements which are closer to the object than the furthest neighbor collected so far. The strategy reduces the overall time complexity of the resulting algorithm to $O(N^2)$.

2.3.5 Scalable Ordination Techniques

In order to scale to large data sets an algorithm with time complexity smaller than quadratic is needed. In [Morrison et al. 2002] a hybrid approach was proposed where a random sample consisting of \sqrt{N} elements is taken and positioned using the force-directed placement algorithm with stochastic sampling in $O(N)$. Each of the remaining $N - \sqrt{N}$ data elements is positioned starting from the 2D position of its nearest neighbor in the sample using a constant time interpolation technique: beginning from the circle with the radius equal to the high-dimensional distance between the data element and its nearest neighbor in the sample, a position is computed using a random, constant-size subset from the sample, such that the stress with regard to this subset is minimized. The layout is refined with a constant number of iterations of the sampling FDP algorithm. Nearest neighbor search is performed by a brute force approach, which takes $O(\sqrt{N})$ time for each element, making it the dominant factor. Therefore, the time complexity of the algorithm is $O(N^{1.5})$. In [Morrison & Chalmers 2003] the brute-force nearest neighbor search is improved with a pivot-based technique reducing the time complexity of the algorithm to $O(N^{1.25})$. Although the pivot-based nearest neighbor search is only approximate, the quality of the layout, expressed as stress was comparable to the brute-force approach. In [Jourdan & Melancon 2004] a further improvement of nearest neighbor finding strategy reduced the running time to $O(N \log(N))$, with an implementation available in [MDS API 2005].

2.4 Visualizing Large, High-dimensional Data Sets

When users have to deal with large, unstructured, complex, high-dimensional data sets they are unfamiliar with, such as enterprise document repositories or patent databases, they need tools allowing them to get an overview of the data, gain insight in complex relationships and structures hidden in the data, and discover central concepts and features. This can be achieved with visual components capable of displaying large data sets and convey relatedness in

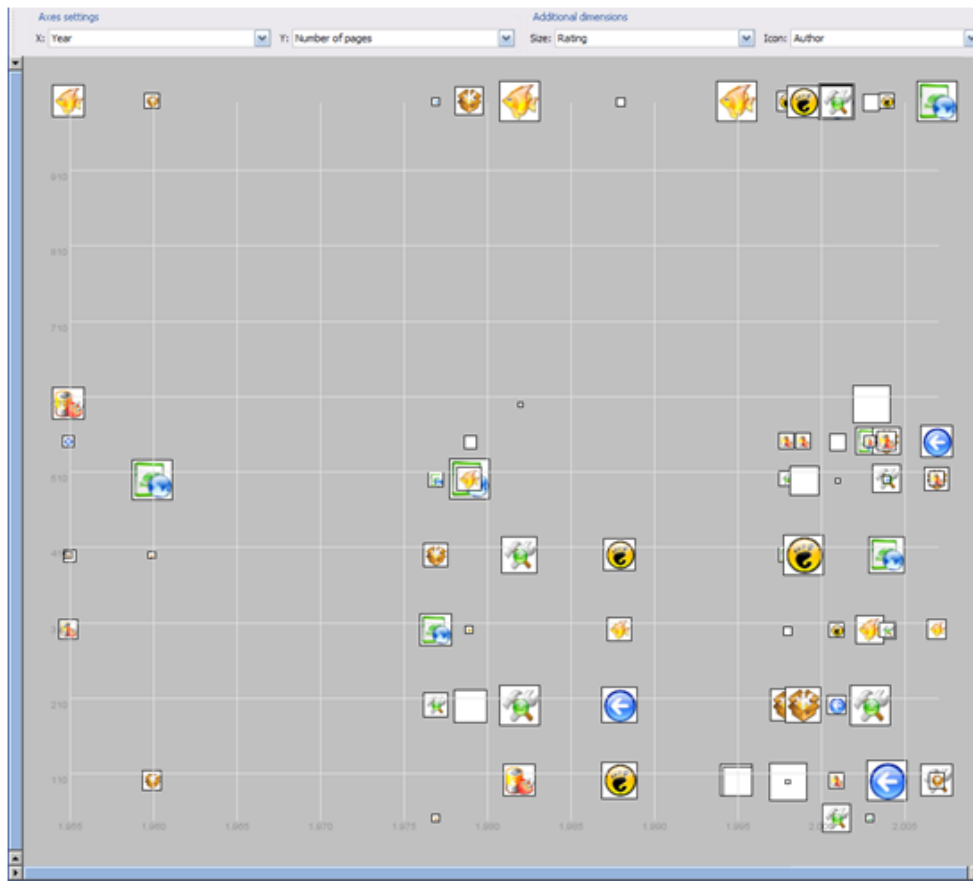


Figure 2.5: A scatterplot visualizing book metadata: publication year (x-axis), page count (y-axis), file size (icon size), author (icon type).

data sets defined by a large amount of features. Large data sets are usually handled by space filling visual methods utilizing the screen real estate provided by high resolution graphics hardware and monitors. Providing overview and conveying relatedness in large data collections necessitate grouping and aggregation of related objects and labeling of the aggregated structures so that the user can quickly find out where to locate information of interest.

2.4.1 Scatterplot

Scatterplot is a visual representation designed for analysis of multidimensional metadata. Typical scatterplot implementations allows mapping of up to five different metadata types (or dimensions) to the axes of a 2D display and visual properties and of displayed items. Besides the x- and y-axis, the available visual channels which can be used to map additional properties are size, color

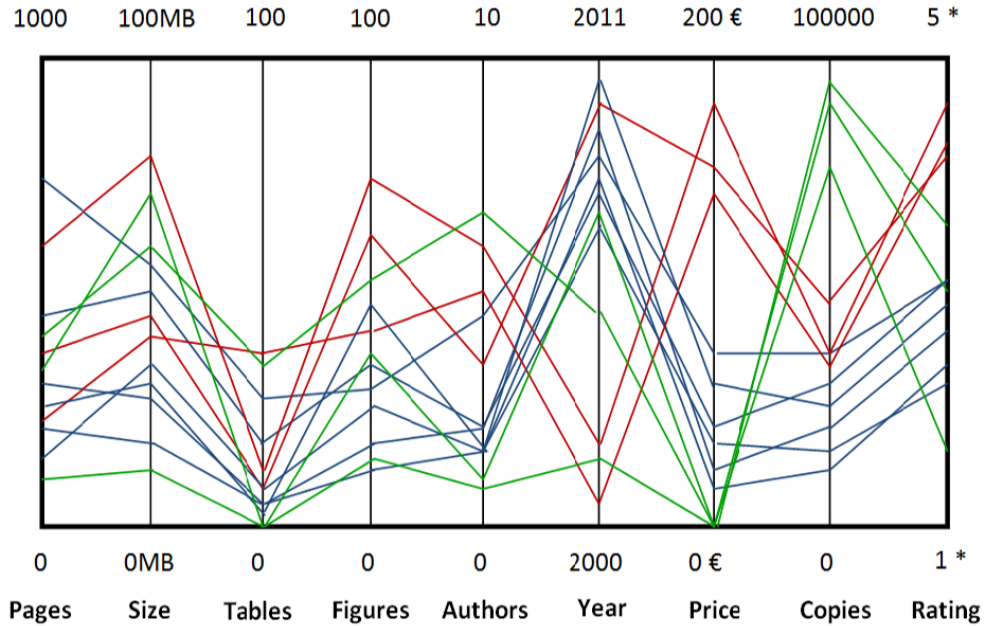


Figure 2.6: Parallel coordinates showing nine metadata types on parallel axes for eBooks (note that synthetic data is used).

and icon of the visualized items. To illustrate the idea a simple example is given in Figure 2.5: a collection of books is displayed with metadata mapped to following channels: publication year is mapped on the x-axis, the number of pages is mapped on the y-axis, rating is mapped on the size and authors are shown through different icons. The scatterplot was implemented within a Master-Praktikum [Kandlhofer 2008] supervised by me, by extending the scatterplot component of the *prefuse* visualization framework [prefuse 2007] with a coordinated multiple view capability (also see Section 2.7). While scatterplots make good use of screen real estate, their ability to visualize complex high-dimensional relationships is fairly limited. To a certain degree, this limitation can be addressed by combining multiple coordinated scatterplots in a single user interface. More advanced scatterplot implementations address scalability by making use of aggregation techniques to reduce clutter when a large number of visual items occupies a small area [Nowell et al. 1996].

2.4.2 Parallel Coordinates

When the number of dimensions which should be displayed is much higher than the number of spatial dimensions and visual channels which are dis-

playable on screen, the problem can be addressed by the parallel coordinates method [Inselberg & Dimsdale 1987]. Parallel coordinates visualization enables the visualization of datasets with very many variables and empower the user to discover relations in high-dimensional data. Dimensions are displayed as parallel vertical axes, and for every displayed object the values of each variable are displayed on the corresponding axis and connected with a polygonal line. Patterns and relationships between objects can be identified as their lines have, locally or globally, similar shapes. Although this visual representation can handle far more dimensions than a scatterplot, the number of axes is still insufficient for very high-dimensional data, such as text.

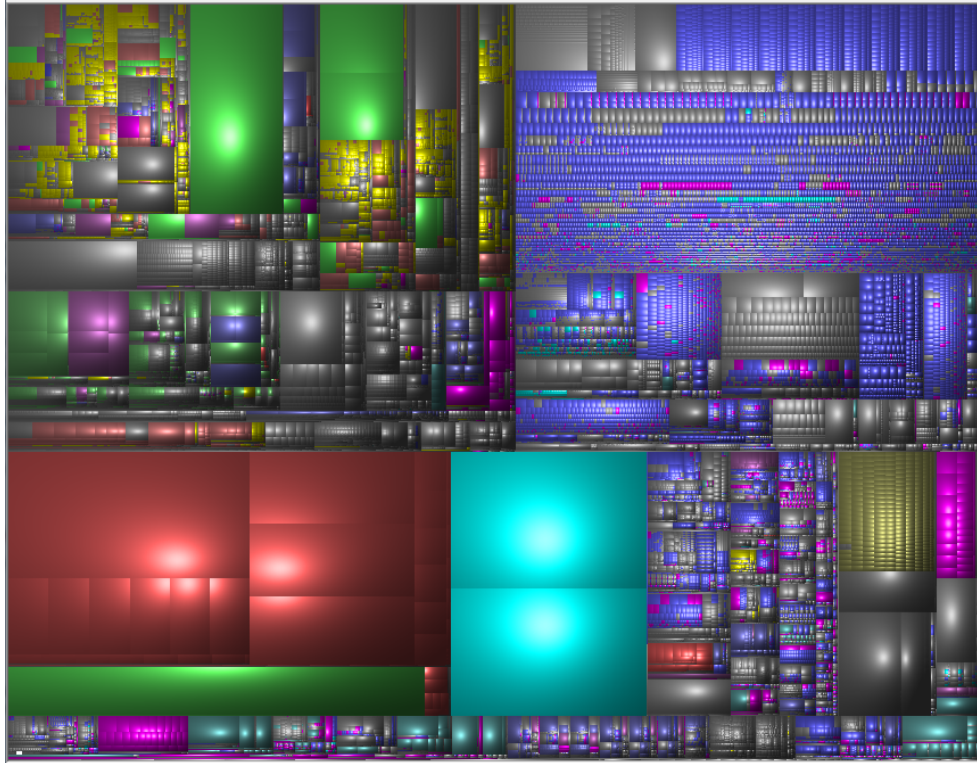
A parallel coordinates visualization in Figure 2.6 displays nine different types of metadata for a synthetic e-book data set, with the line color encoding different book publishers. It is easy to see that red e-books have high ratings and high prices, the blue e-books are cheaper and have lower ratings, while the green e-books are free of charge and achieve the highest delivery rates.

2.4.3 Treemaps

Treemap [Shneiderman 1991] is a common visual representation used to convey hierarchically structured data. Nodes of the hierarchy are represented as recursively nested rectangles, where the size of each rectangle corresponds to a selected property of the underlying data, for example the number of leafs in the corresponding subtree. Color of the rectangles is typically used to convey further properties of a node, and sometimes other representations fitting within a node's area, for example histograms or labels, can be used instead. Through size and color coding the user can easily spot patterns in data which would be less obvious to recognize in other hierarchy representations. A treemap is a scalable representation with efficient use of screen space, however various algorithms used to compute the layout of tree map, always produce a compromise between the following two properties which have an inverse relationship:

- aspect ratio of the rectangles, where high ratios translate to lower readability and visual appealing,
- the order and position of areas, which reflects the relationships within the hierarchical structure.

In Figure 2.7 a tree map of a hard drive, computed by WinDirStat tool [WinDirStat 2007] can be seen. The size of the directory is mapped to the rectangle area, while the color indicates the predominant file type. A short overview of treemap based techniques is available in [Shneiderman 1998-2009] and [Kerwin 2011].



Extensi...	Col...	Description	> Bytes
.dll	Application Extension	Application Extension	22.2 GB
.cfs	CFS File	CFS File	15.3 GB
.rar	WinRAR archive	WinRAR archive	9.0 GB
.sys	System File	System File	8.6 GB
.exe	Application	Application	4.7 GB
.pdf	PDF Document	PDF Document	4.0 GB
.jar	Executable Jar File	Executable Jar File	3.2 GB
.ppt	Microsoft Office PowerPoint ...	Microsoft Office PowerPoint ...	2.9 GB
.sv...	SVN-BASE File	SVN-BASE File	2.3 GB
.cab	WinRAR archive	WinRAR archive	2.2 GB
.zip	WinRAR ZIP archive	WinRAR ZIP archive	2.0 GB
.g...	GMP File	GMP File	1.8 GB
.jpg	JPEG Image	JPEG Image	1.8 GB
.msp	Windows Installer Patch	Windows Installer Patch	1.7 GB

Figure 2.7: Treemap showing the file hierarchy on a hard drive.

2.4.4 Cluster Map

Cluster Map [Clustermap 2011] visualization technique similar to Venn and Euler diagrams, for visualization of classified objects and their relationships, or light-weight ontologies in Cluster Map parlance [Fluit 2005]. The main purpose is to show if and how, i.e. through which features, these sets overlap. Other features include guided exploration, auto-generated suggestions for refinement, semantically rich interaction through filtering and refinement of metadata facets such as for example information type, people, location, time etc.

2.4.5 Information Landscapes

Information landscapes employ a geographic landscape metaphor for analysis of complex relationships in large, high-dimensional data sets by conveying relatedness in the data through spatial proximity in the visualization. Relatedness between a pair of data objects is typically defined as the similarity of the feature vectors describing the data objects, expressing for example topical relatedness, relatedness through authors or geographical location etc. Hills represent groups, or visual clusters, of related data objects and emerge where their count (density) is large. They are labeled by most significant features from the underlying data enabling the user to immediately identify clusters of interest. Hills are separated by sparsely populated areas which are represented as sea. Information landscape also conveys the size and cohesion of clusters through visual properties. The height of a hill is an indicator of for the number of data objects belonging to a cluster, while spatial compactness of the cluster is an indicator for higher cohesion, i.e. higher relatedness of its elements. The resulting visual representation provides an overview of the whole data by unveiling structures in the data which arise through relatedness and similarity. Following the labels of visual clusters users can interactively explore the data set and gain insight into more detailed relationships in the data. For example Figure 2.8 shows an information landscape created from several thousand text documents, while an application for visualization of multimedia data [Sabot et al. 2008b] can be seen in Figure 2.9.

Powered by modern graphics subsystems visualization of hundreds of thousand or millions of data objects becomes feasible, provided scalable ordination algorithms can project high-dimensional data objects into the visualization space. An overview and a brief history of using the geographic landscape metaphor for visualization of non-geospatial data is given in [Old 2002].

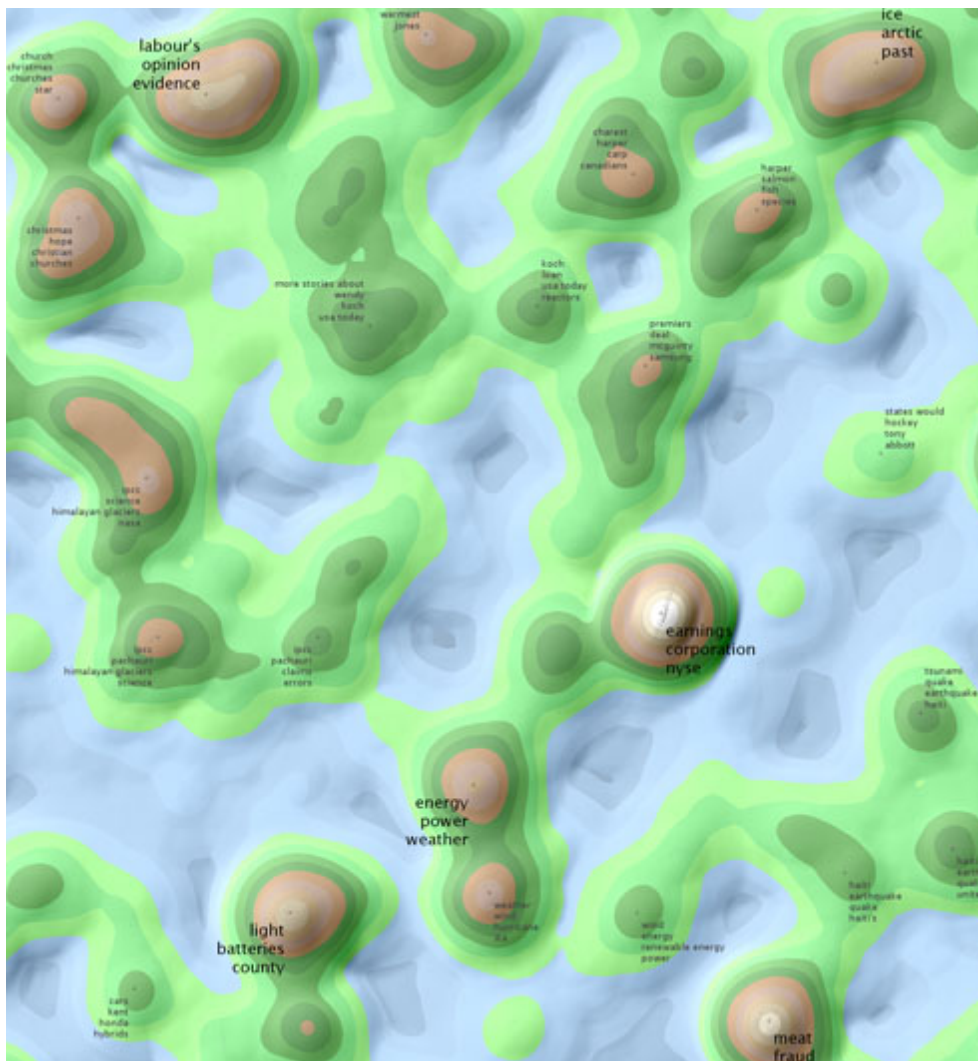


Figure 2.8: An information landscape built from several thousand documents on climate change [Sabol et al. 2008c] (see Chapter 4 for details).

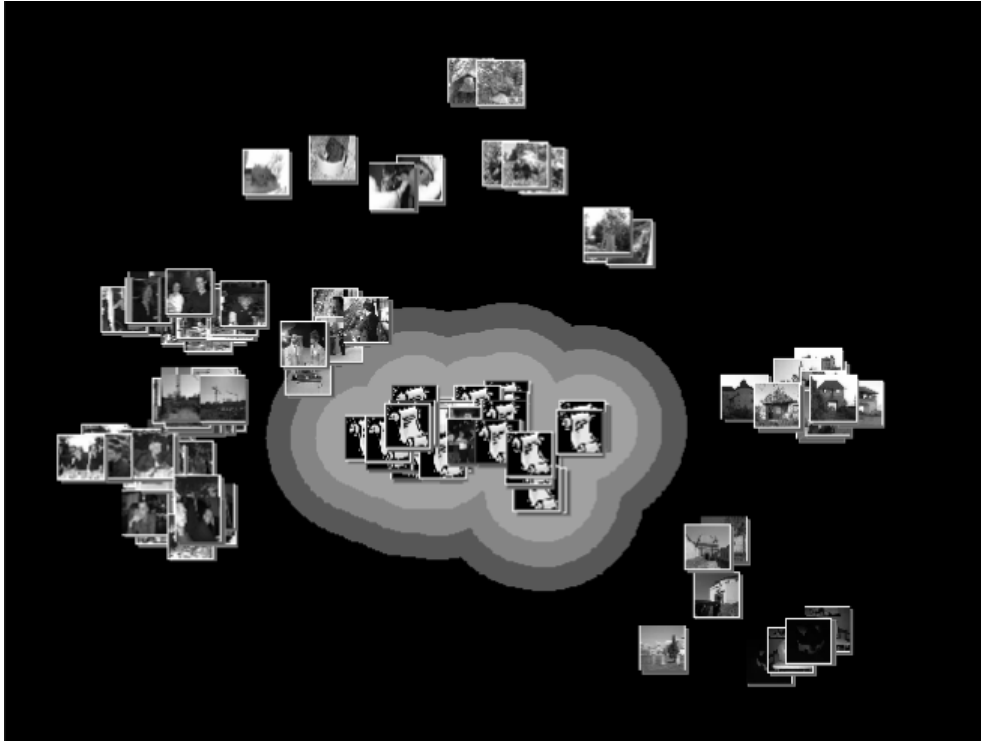


Figure 2.9: An information landscape showing a mix of images and text documents [Lux 2004].

2.4.5.1 Bead

Bead [Chalmers and Chitson 1992] is one of the first systems to use the landscape metaphor [Chalmers 1993] to visualize text data sets. Early versions employed a simulated annealing ordination method which was later replaced with a fast force-directed placement method [Chalmers 1996] with quadratic running time, allowing the system to visualize small to medium size data sets. It produces a so-called information terrain where topically similar documents formed visual clusters. In the first version of Bead, text documents were freely placed in the virtual 3D space, however the lesson learned was that due to occlusion effects in 3D and the complexity of navigation in 3D space placed an excessive cognitive load was placed on the users. To overcome this issue additional forces were used to pull the documents closer to a 2D plane effectively producing a 2D layout with smaller variations in height (also called a 2.1D representation). Triangulation of the layout produced an information terrain with peaks and valleys which helped the user orientating and navigating through recognition of terrain features. Usability of the resulting 2.1D

landscape metaphor proved superior than the full 3D representation.

2.4.5.2 SPIRE

The SPIRE (Spatial Paradigm for Information Retrieval and Exploration) system [Wise et al. 1995] is a set of tools developed for intelligence agencies targeted at users who need to analyze large amounts of new text documents in a short period of time. SPIRE automatically analyzes and visually presents large amounts of text content supporting the user in exploring, retrieving, categorizing and summarizing. The system offers two landscape visualizations: the Galaxies visualization uses the metaphor of the night sky with stars representing documents and star clusters, i.e. galaxies, representing clusters of topically related documents; ThemeScape representation is a further development of the concept into a more scalable 2.1D landscape metaphor, where document clusters are represented as elevations of a terrain. SPIRE generates a visualization in the following process [Wise 1999]:

- A text engine analyzes the documents, extracts discriminating terms and constructs term vectors. The process includes frequency-based term selection to filter out term with low discrimination power, and in-document term clustering (so-called condensation clustering value) as good terms tend to occur in bursts.
- Clustering of document term vectors is performed by a fast divisive clustering algorithm related to k-means clustering with furthest distance seeding strategy.
- Projection (ordination) into 2D space is performed by a scalable "Anchored Least Stress" algorithm, which projects the cluster centroids using a PCA-based techniques and then places the documents based on their high-dimensional distances to the projected cluster centroids only.
- Visualization generation is trivial for the Galaxies view (bright dots on dark background), ThemeScape visualization is constructed by layering of significant, high-discrimination power terms belonging to the document as sedimentary layers of certain thickness over each other to obtain the landscape height. Peaks which arise through this process are highest where the density of the documents and topical terms is high, symbolizing clusters of related documents

More details on clustering and ordination algorithms applied in SPIRE is available in [York et al. 1995]. SPIRE has been developed further with the goal of scaling to multi-Gigabyte data sets. The improved system, in the

mean time renamed to IN-SPIRE, is based on a parallelized text engine which was shown to scale well with Gigabyte-sized data sets, and with the CPU count (up to 32 CPU cores) [Krishnan et al. 2007].

2.4.5.3 VxInsight

VxInsight [Davidson et al. 1998] is a visual knowledge mining tool, employing a 2.1D terrain metaphor very similar to the SPIRE ThemeScape visualization. The system was built to analyze various types of high-dimensional data, not just text. It was successfully applied on gene expression data [Mosquera-Caro et al. 2004], patent databases [Boyack et al. 2000], and scientific and technology document sets [Boyack et al. 2002]. Early versions of the ordination algorithm employed a dual algorithm strategy: an eigenvalue decomposition based MDS method produced an initial layout which was globally optimal with regard to stress, but typically too tightly clustered to be useful for visualization; a heuristic method, which would normally converge to a local minimum, was used to refine the optimal configuration and produce a usable, visually appealing layout. Later versions of the ordination algorithm kept only the force-directed method, which was extended with methods for avoiding of getting stuck in local minima, such as introduction of stochastic jitter and barrier breaking techniques [Davidson et al. 2001]. Scalability of the force-based method was addressed by a grid based method which avoids comparison to all other vertices when computing the force. Scalability to millions of objects is expected by the authors, however it is not demonstrated.

2.4.5.4 Other Approaches

Other approaches to ordination and various other applications of information landscapes were proposed, for example: In [Skupin 2004] visualization of scientific publications was performed using a self-organizing map algorithm. The emphasis was placed on the design of the visual landscape representation which faithfully reproduces cartographic principles. A meta search engine for visualization of search results in the Web browser is shown in [Tianamo 2008]. In [Lux 2004] information landscapes are applied on multi-modal data sets containing images, sounds and text. In addition to multi-media data sets [Heilig et al. 2009] demonstrated application possibilities such as projection on a high resolution display (wall), collaboration on a Microsoft Surface multi-touch table, and zoomable object-oriented interfaces where zooming capability is extended to visualize contents of single documents within the visualization. An approach for supplying user feedback through modifications of the visual layout and readjusting (learning) the model was proposed in [Neidhart 2005].

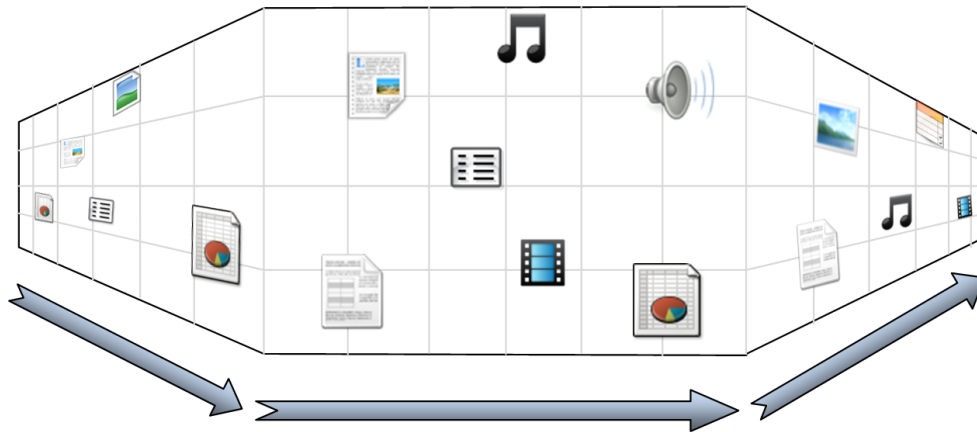


Figure 2.10: A mock-up illustrating the principle of Perspective Wall (arrows indicates the direction of time).

2.5 Visualization of Temporal Data

Time series analysis is of paramount importance in areas where the temporal aspect is central to the data, for example in signal processing. However, even in data sets which are not primarily concerned with time, time-related metadata, such as creation date or a particular time stamp, is very often present and constitutes an important additional aspect of the data. Also, many repositories are subjects to changes over time and understanding these changes can play an important role. Visualization of time-dependent data deals with visual metaphors and tools for capturing and understanding the changes and discovery of major trends in the data set. This section gives a brief overview of relevant techniques, for a more comprehensive survey see [Muller & Schumann 2003] and [Chin et al. 2009].

2.5.1 Perspective Wall

Perspective Wall [Mackinlay et al. 1991] is a technique for visualizing linear information featuring smooth integration of a detailed and contextual views. It is a 3D interactive animation of a 2D layout spanned over a 3D wall, where the information is placed along the time axis which flows from left to right side of the visualization. The wall consists of three regions:

- A central region, which faces the viewer, is used for viewing detailed information within a specified time interval.
- Two context regions, which are placed on the left and on the right of

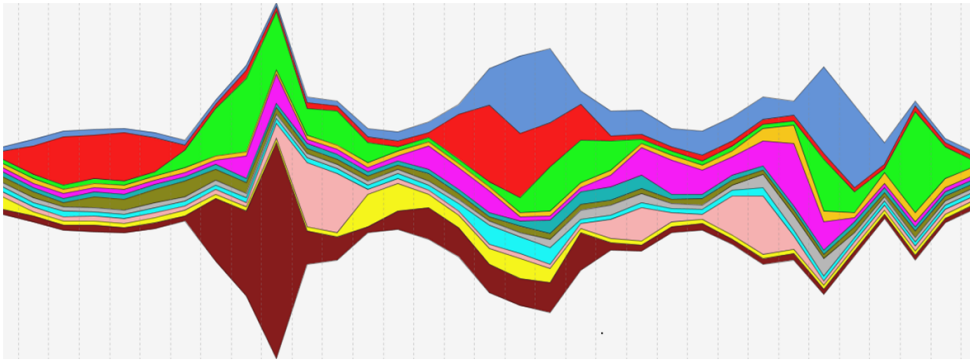


Figure 2.11: An example illustrating the idea behind ThemeRiver.

the central region, and bent in 3D perspective towards the rear part of the scene. Information before and after the interval shown in the central region is shown on the left and right context region, respectively.

Through the use of 3D perspective for the context regions, a visual distortion is introduced which shows temporally closer information is larger and in more detail, while temporally distant information is scaled down and displayed at low level of detail. The idea is illustrated in Figure 2.10. The geometry of the representation allows for visualizing wide time intervals and provides efficient use of screen real estate. Smooth transitions of views are possible by sliding the time axis over the wall.

2.5.2 LifeLines

LifeLines [Plaisant et al. 1996] is a visualization of personal histories, with applications including for example professional histories or medical records [Plaisant et al. 1998]. LifeLines provide an overview of a personal history minimizing the chances of oversight, and facilitates discovery of unexpected trends or anomalies. The visual representation places multiple facets of person's records on the y-axis, in case of medical records for example consultation histories, symptom manifestations, treatments, hospitalizations etc. Each facet is displayed as an individual horizontal time line extending over the x-axis (the time axis). Color and thickness of the line represent relationships or significance, while icons placed along the line represent events, for example consultations with a particular physician or administration of medicaments. Filtering and zooming functionality is also supported.

2.5.3 ThemeRiver

ThemeRiver [Havre et al. 2002] uses a river metaphor to visualize topical changes in large documents collection. Design of the visual representation was based on discussions on perceptual processes in [Hoffman 2005], on how human mind perceives images features such as symmetry, proximity, continuity, closure and similarity, and on the fact that continuous shapes are easier to perceive than those containing abrupt changes and discontinuities [Ware 2004]. Topically related groups of documents are visualized as color coded currents flowing along the x-axis which represents the flow of time. The thickness of a current at the given x-coordinate represents the strength of the topic at the corresponding moment in time. Change of the width represents the change of topic's strength allowing the discovery of topical trends: narrowing or broadening of current's width indicates decreasing or increasing of the strength of the corresponding topics. Various topical currents, each shown in different color, are stacked over each other to produce a single river of topics, where relationships and correlations between various topics can be identified. Figure 2.11 illustrates the idea behind ThemeRiver. ThemeRiver and numerous similar visual representations have been applied on data sets other than document collections. One of them is, for example, NameVoyager [NameVoyager 2011], an Internet-based visualization of baby name trends and changes in name popularity.

2.5.4 More Temporal Data Visualizations

Commonly, the the flow of time is displayed along a straight axis. A spiral axis, such as used in [Weber et al. 2001] provides certain advantages, for example, it is suitable for detecting cycles and recurring events, and it allows for displaying long time intervals with high temporal resolution even when the available screen real estate is not large. TimeWheel and MultiComb [Tominski et al. 2004] are examples of temporal data visualizations for multi-dimensional data. They employ an axes-based principle, such as parallel coordinates (see Section 2.4.2) to visualize multiple dimensions. While parallel coordinates shows the variables as parallel axes, MultiComb places the axes radially in the x-y plane. The time flow is represented by a centrally placed z-axis which is orthogonal to the x-y plane. In the resulting 3D visualization a time-dependent graph plot is obtained showing the development of each dimension.



Figure 2.12: A visualization showing a global tag cloud (central area) and tag clouds for six different categories (surrounding areas).

2.6 Other Visualizations

After providing a survey on visualization of topical and other high-dimensional patterns, and on visualization temporal data and temporal developments [Sabot et al. 2008a], a short introduction to visualization of other aspects of the data, such as geospatial information, keyword and concept information, and relationships between the concepts, shall also be given. As these visualizations are not the focus of this work only a brief overview is given here.

2.6.1 Tag Clouds

Tag clouds are an increasingly popular visual representation consisting of keywords and short phrases which describe the content of a text document collection. Displayed terms are extracted from document content using information extraction [Kaiser & Miksch 2005] and statistical techniques. Size, color and layout of the displayed terms are driven by their importance as well as by aes-



Figure 2.13: Geo-visualization of Austria showing locations referenced in news articles. Location references are shown as cones, with the size corresponding to the number of articles referencing that location.

thetic and usability criteria [Seifert et al. 2008]. Figure 2.12 shows a tag cloud where terms are subdivided into several categories. The assignments of terms to categories is either manually defined or computed automatically using text classification methods. Each category is assigned an area which is generated by constructing a Voronoi diagram [Okabe et al. 2000, Aurenhammer 1991], whereby the generator points are either set manually or are computed by an ordination algorithm depending on the topical similarity of the category content.

2.6.2 Geovisualization

Since the advent of Google Maps, geovisualization has definitely become mainstream. Visualization of geospatial metadata is a natural fit for various geovisualization approaches [Dykes 2005]. Geospatial information automatically extracted from text can be shown on geographical maps [Scharl & Tochtermann 2007] to reveal where something is happening. For example, in Figure 2.13 a geovisualization of locations extracted from German news articles [Lex et al. 2008] can be seen. The extracted locations, shown on a map of Austria, are represented by cones where the size of a cone corresponds to the number of news articles referencing a particular location. Clicking on a cone triggers a filtering mechanism which restricts the list view on articles mentioning the corresponding location.

GeoTime [Kapler & Wright 2005] is a geovisualization which fuses the temporal component together with the geospatial. It employs a 3D visualization consisting of a geographic map shown in the x-y plane, where the flow of time is orthogonal to the map, i.e. shown along the z-axis. GeoTime facilitates

2.7 Coordinated Multiple Views

Properties and capabilities of each visual representation are designed to target the characteristics of the specific data type [Shneiderman 1996], such as temporal data, geographic information, hierarchical or graph structures etc. In complex, heterogeneous repositories which contain various data types, the analysis of a data set necessitates considering more than just a single aspect of the data. In application domains with data sets characterized by patterns not only within each separate data aspect, but also by patterns between the different aspects, these patterns may resist analysis unless all aspects can be analyzed simultaneously. Visual representations employ a specialized metaphor which is restricted to revealing patterns only for one, or a small amount of data aspects. As a consequence, a visual user interfaces capable of simultaneously conveying different aspects of the data will usually employ several specialized visual components. In such a complex user interface should behave in a coherent, unified way is necessary that different components act and function in a harmonious, coordinated fashion.

Coordinated multiple views (CMV) is a paradigm for systems which use two or more distinct views to support the investigation of the same conceptual entities. User interfaces built along the lines of CMV paradigm achieve the tight coupling of several components effectively fusing them into a single coherent user interface where interactions performed in one component are immediately reflected in all components within the GUI. In[Baldonado et al. 2000] a collection of guidelines for using coordinated multiple views in information visualization are given. These boil down to the recommendation than in order to reduce the cognitive load on the user multiple views should only be used when the diversity of data aspects in complex data sets can not be handled by a single view. Care should be taken to balance the costs and benefits of using multiple visual representations. Also highlighted are the necessity to use complementary views, which need to be kept in consistent states at all times employing a well-defined, consistent coordination model.

A Snap-Together, a system backed by a relational data model, was proposed in [North & Schneiderman 1999]. It allows the user to dynamically combine and bind different visualizations to produce customized user interfaces, where coordinations are specified by joining different data properties, effectively "snapping" various visualizations together. Users can construct and apply different coordination actions between views themselves, such as for example brushing-and-linking, drill down, overview and detail, and synchronized scrolling. The concept was developed further in [North 2000] and was applied in a commercially available SpotLite DecisionSite tool [DecisionSite 2001],

which is an enterprise visual analysis suite composed of a series of coordinated visualizations including scatter-plots, pie charts, bar charts, function graphs, geographic maps, lists, tables etc.

Another approach to view coordination employs the well-known model-view-controller architecture, as proposed in [Pattison & Phillips 2000]. Model-View-Controller (MVC) is an architectural pattern applied in user interface software engineering which focuses on separation of concerns between the model, which manages and provides access to the data, the view, which is concerned with rendering and interaction and the controller, which implements the domain logic. Comprehensive information on various approaches, methods and models used in coordinated multiple views based user interfaces can be found in [Müller 2005] and [Roberts 2007].

Chapter 3

Early Contributions

This chapter describes my early research and development results, including new ideas, concepts and methods, which are important and relevant to this work. Concepts and methods introduced in this chapter are:

- Incremental clustering and ordination algorithms and the information landscape with dynamic topology are introduced in the WebRat system for topical visualization of search result sets.
- A scalable system for similarity-driven visualization of large, hierarchically organized document collections containing up to millions of documents was realized in the InfoSky project.
- Tools and methods for temporal visualization were introduced in the following projects: a visual application for discovery of communication patterns within a meeting was developed in MISTRAL project; a prototype search result analysis application, developed in OnAir project, includes a temporal analysis component.

These results were realized in separate projects and systems, and were only later extended and combined into a consistent visual set of tools, described in Chapter 4, for addressing the goals defined in Section 1.2.2.

3.1 Starting Point

Visualisation Islands [Sabol 2001], developed during my Master's Thesis as one of the visual front ends for the xFIND search engine [Andrews et al. 2001], is a simple visual system, implemented in Java, for topically organizing search result sets. The system clusters and visualizes search results in the form of an explorable, topically organized information landscape (see Figure 3.1).

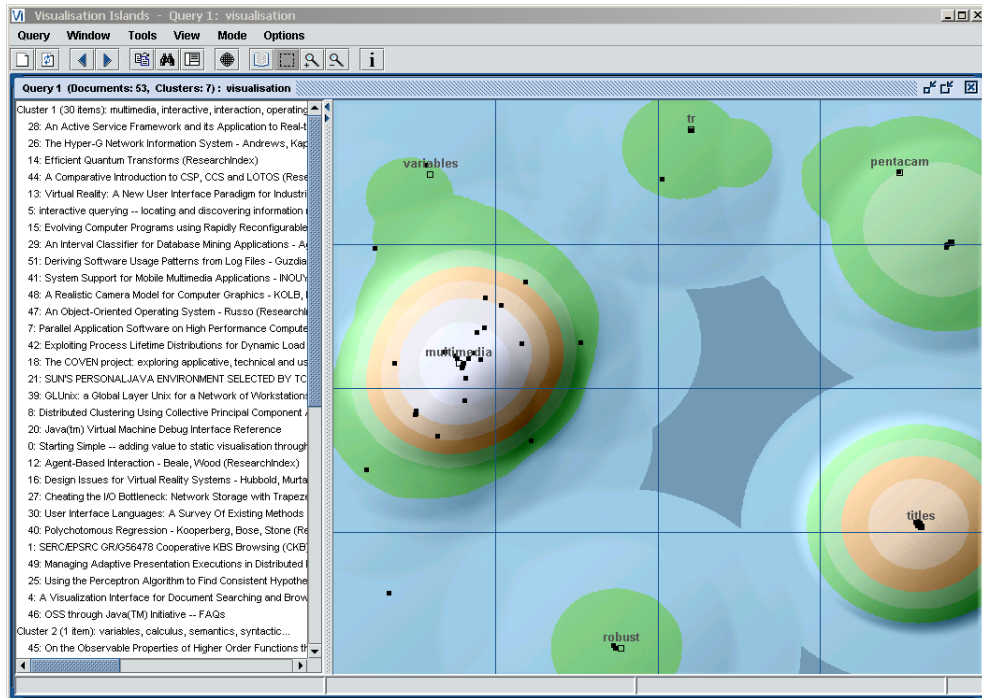


Figure 3.1: Visualisation Islands showing clustering and an information landscape for search results returned for query "visualisation".

The topical map visualization is constructed by first applying k-means or hierarchical agglomerative clustering algorithms on document keyword vectors which are returned by the xFIND engine. Document vectors are subsequently projected into the 2D visualization space according to their topical similarity using a force-directed placement algorithm with stochastic sampling [Chalmers 1996]. A topographic background image is computed based on 2D document density. The system supports zooming, selection, keyword filtering and iterative scattering and gathering of results along the lines of [Cutting et al. 1992]. However, the system is very limited with respect to scalability (up to 1000 documents), handling of high-dimensional data (up to 10 keyword per document), can not deal with dynamically changing data collections, provides no possibilities for visualization of rich metadata, has no extension mechanisms for support of additional visualizations (other than the information landscape), and no view coordination.

This work continues where Visualisation Islands has stopped, gradually introducing new ideas, algorithms and visual techniques capable of handling large, unstructured, complex, dynamically changing data sets.

3.2 Incremental Visualization of Search Results

WebRat [Sabot et al. 2002a] is a light-weight, interactive system for visualizing and refining search result sets collected from several search engines. Rather than presenting a linear ranked list of search results, documents matching a query are clustered on-the-fly and visualized as a dynamic information landscape. The main feature of WebRat is, that as documents returned from search engines are pouring in, thematic clusters and visual representation are built, analyzed, and visualized incrementally and in real time. This is an agile approach to search result set exploration, allowing the user to start exploring search result sets almost immediately after the search query was executed, in contrast to traditional meta-search engines which let the user wait until a certain amount of results was collected. This incremental capability makes WebRat one of the first (if not the first) systems employing an animated information landscape with fully dynamic topology.

WebRat was developed at the Know-Center's [Know-Center 2011] Knowledge Discovery Division team. My roles in the project involved the overall conception, the ordination algorithm, and to a certain degree the clustering algorithm and the map generation algorithm.

3.2.1 Goals

The standard web search interfaces of today differ very little from the interfaces of the first full text searchable databases decades ago. Users type in one or more query terms and the results of searches on billions of web pages are presented as simple linear, ranked lists of matching documents, in decreasing order of relevance. Such representations are incapable, however, of expressing the manifold topical dimensions contained in typical search result sets. On closer examination, the several drawbacks of traditional ranked lists stand out:

1. Only a small subset of the matching documents can be displayed on a screen at a time.
2. To find relevant documents, users may have to scan a large amount of textual information word by word, narrow down the search query, or both.
3. Only recently search engines began provided the user with keyword suggestions to use for query refinement, however not within different topical contexts present in the data.

4. Topical interconnections or clusters present in the result set are hidden from the user. Relationships between information entities which should be obvious and easy to discover are obscured.

Hence, users may be hindered in two of their fundamental tasks: finding relevant information, and placing information into context. While many retrieval and visualization systems have attempted to address these issues, most rely on pre-calculated schemes, require sophisticated graphics hardware or can only operate in a powerful server environment. However, most users have standard office PCs at their disposal, and any requirements beyond that restrict the usefulness of any solution. The WebRat system, designed to address these problems, is built around a framework capable of:

- querying various web data sources in the fashion of a meta search engine;
- dynamic, incremental clustering and ordination of search results;
- extracting keywords describing topics and using these as cluster labels;
- interactive visualization of results.

All calculations can be performed on standard office machines, visualization works with low-performance graphics hardware, and no dedicated server or service environment is necessary.

3.2.2 Concept

WebRat supports the identification of topical clusters of search results through dynamic, incremental clustering, ordination and visualization. A thematic landscape of matching documents is generated and updated on-the-fly as search results arrive. WebRat also simplifies query refinement, by labeling thematic clusters with automatically extracted discriminating keywords. Figure 3.2 shows a typical WebRat visualization (left), zoom in on a cluster (up-right), and query refinement using cluster labels (down-right).

WebRat supports the retrieval process by dynamically identifying and visualizing clusters of similar documents, and by offering means for search query refinement. Users start the query process by entering a number of query terms which are sent to various data sources. The results form islands on a virtual contour map which are labeled with the according keywords. The thematic landscape of matching documents is generated and updated at regular intervals as more and more search results arrive. As the generated thematic landscape is getting more stable as the number of processed search results increases, the user is given the possibility to zoom in to reveal more details, and navigate

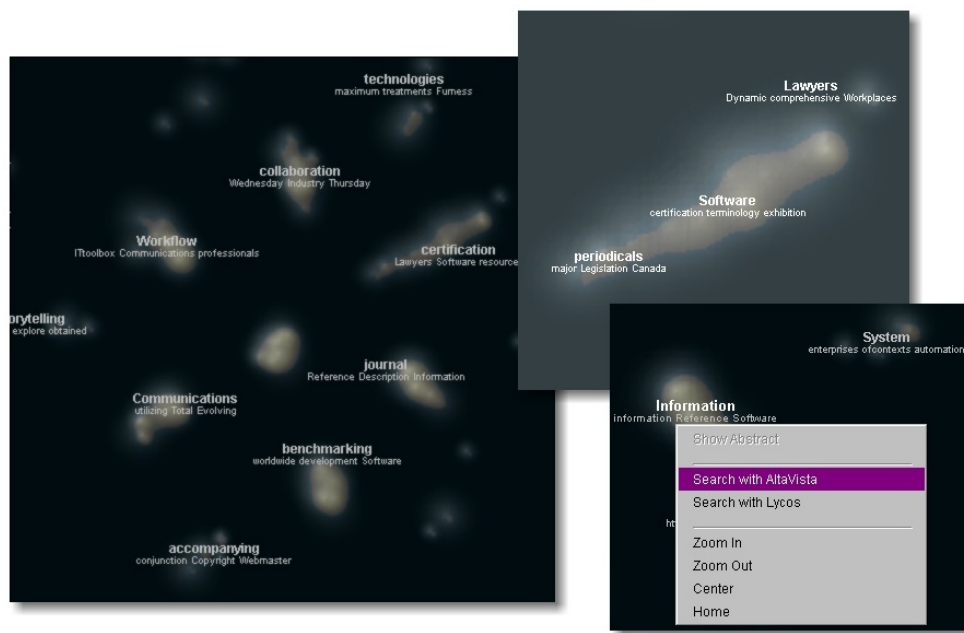


Figure 3.2: Visual query refinement with WebRat: 1) overview for query "Knowledge Management", 2) zoom in on "certification" cluster, 3) Refining the query with the term "Information".

to different regions of the map freely, even while the landscape generation is still in progress. Query refinement is simplified by labeling thematic clusters with automatically extracted keywords. Labels are calculated on the fly, so as to always describe the most obvious concentrations of documents (the largest islands) in the visualization. Users can invoke a context menu for each label which allows to re-query any of the used data sources by refining the original query with the label keywords. In this way, the system supports the user in gaining an overview of the data, discovering major topical clusters and narrowing down the result set by formulating a refined search query.

Therefore, a typical retrieval process using WebRat consists of the following steps (see Figure 3.2):

- The user enters some initial query terms, probably of a general nature.
- The user examines the visualized result set for topics of interest.
- The user zooms in on the chosen topical cluster to reveal more detailed topical keywords.
- The user launches a new search through a topic of choice, refining the

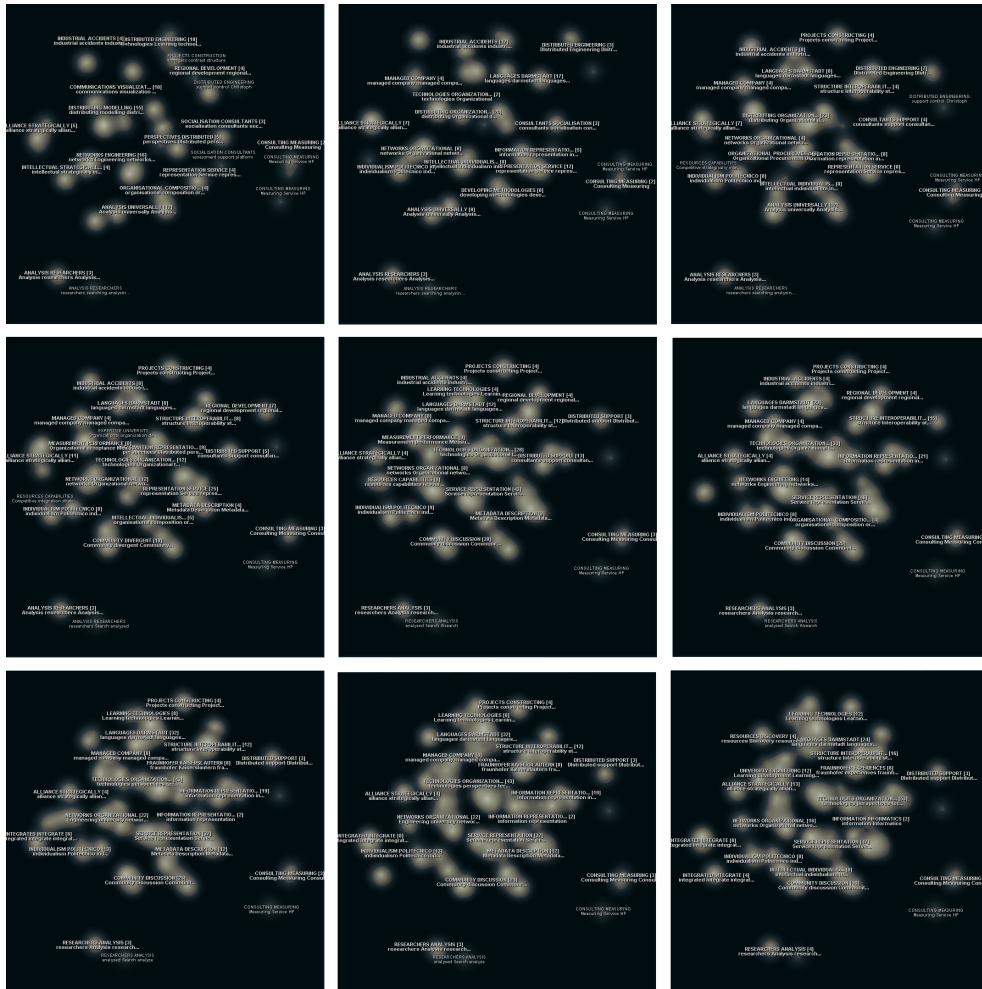


Figure 3.3: A series of nine snapshots of WebRat incrementally processing about 300 scientific paper abstracts.

initial search.

- The user browses the refined result set from the specified search engine(s).

3.2.3 Incremental Visualization

The incremental capability is illustrated in Figure 3.3 where a set of about 300 scientific paper abstracts on knowledge management is incrementally clustered, projected and visualized. On the top-left a first snapshot can be seen which contains about a quarter of the whole data set. Document count increases line-by-line from left to right, with a complete data set visualized in

the snapshot shown on the bottom-right. It can be seen that main topical structures mostly remain stable during the process, giving the user the possibility to explore the search results set before all data has been retrieved and processed. Although smaller changes and adjustments do occur, they are incorporated gradually and smoothly so that the user can follow the movement of the landmarks. In this way the user does not lose orientation and can retain recognition of his points of interest.

3.2.4 Architecture

The WebRat framework consists of four main components, shown in Figure 3.4:

1. **High-Dimensional Component:** Retrieves the search results from the search engines and creates a high-dimensional representation for each result. It includes the following subcomponents:
 - The Grabber sends search queries to different search engines, retrieves the document-snippets from the result lists and adds them to the global document pool.
 - The Vectorizer analyses document snippets with a language-independent method known as n-gram decomposition, to create high-dimensional representations of the retrieved documents. Subsequently a TF-IDF weighting scheme is applied. A cosine similarity function for comparing the vectors is provided.
 - The High-Dimensional Centroid Computation Unit computes the high-dimensional centroids for new clusters and continuously updates the existing clusters as new documents are inserted
 - The Keyword Extractor applies a weighting scheme to identify terms (n-grams) which best describe clusters and/or regions of interest in the 2D layout. Subsequently keywords are extracted by identifying the words and text segments which were the sources of the n-grams with the highest rating.
2. **Mapping Component:** performs the dimensionality reduction from the high-dimensional term-vectors to the 2D visualization space. It comprises two sub-units:
 - The Force-Directed Placement (FDP) Unit performs multidimensional scaling of the data set: based on high-dimensional term vector similarities, 2D document positions are computed preserving

the high-dimensional relations as far as possible. The FDP algorithm can operate in cluster-oriented mode to improve performance and layout separation.

- The Low-Dimensional Centroid Updater continuously recomputes the low-dimensional cluster centroid positions as the clusters children are repositioned by the layout algorithm. As documents are added to clusters their weights are adjusted to ensure that the FDP interactions between document and clusters are performed with correct strength. This scheme significantly improves the performance of the FDP algorithm.
3. Low-Dimensional Component: consists of a user interface, a landscape generator and a 2D clustering module:
- The User Interface sub-component presents the visualization and handles the interactivity and navigation. It dispatches events and tasks to other components depending on user actions.
 - The Landscape Generator computes a shaded islands landscape based on the 2D document density. Labels describing different landscape regions are computed dynamically from the extracted keywords depending on the zoom level and context.
 - The 2D Clustering Module analyses the landscape to identify positions of the 2D document density maxima created by the FDP algorithm. These are used as cluster seeds. Clusters are then created by assigning each document to the nearest seed. In doing so, a feedback loop is created to the FDP algorithm, which uses the created clusters to improve performance and layout quality.
4. Shared Component: consists of document and cluster pools, high-dimensional and low-dimensional metric definitions, threading and control logic facilities, an event model, as well other components shared by the three processing components.

For maximum adaptability to various data sources and visualization metaphors, each component is separately configurable and exchangeable. The components operate as Java threads, communicate with each other through a shared data pool, and use an event model for exchanging messages or requesting specific operations.

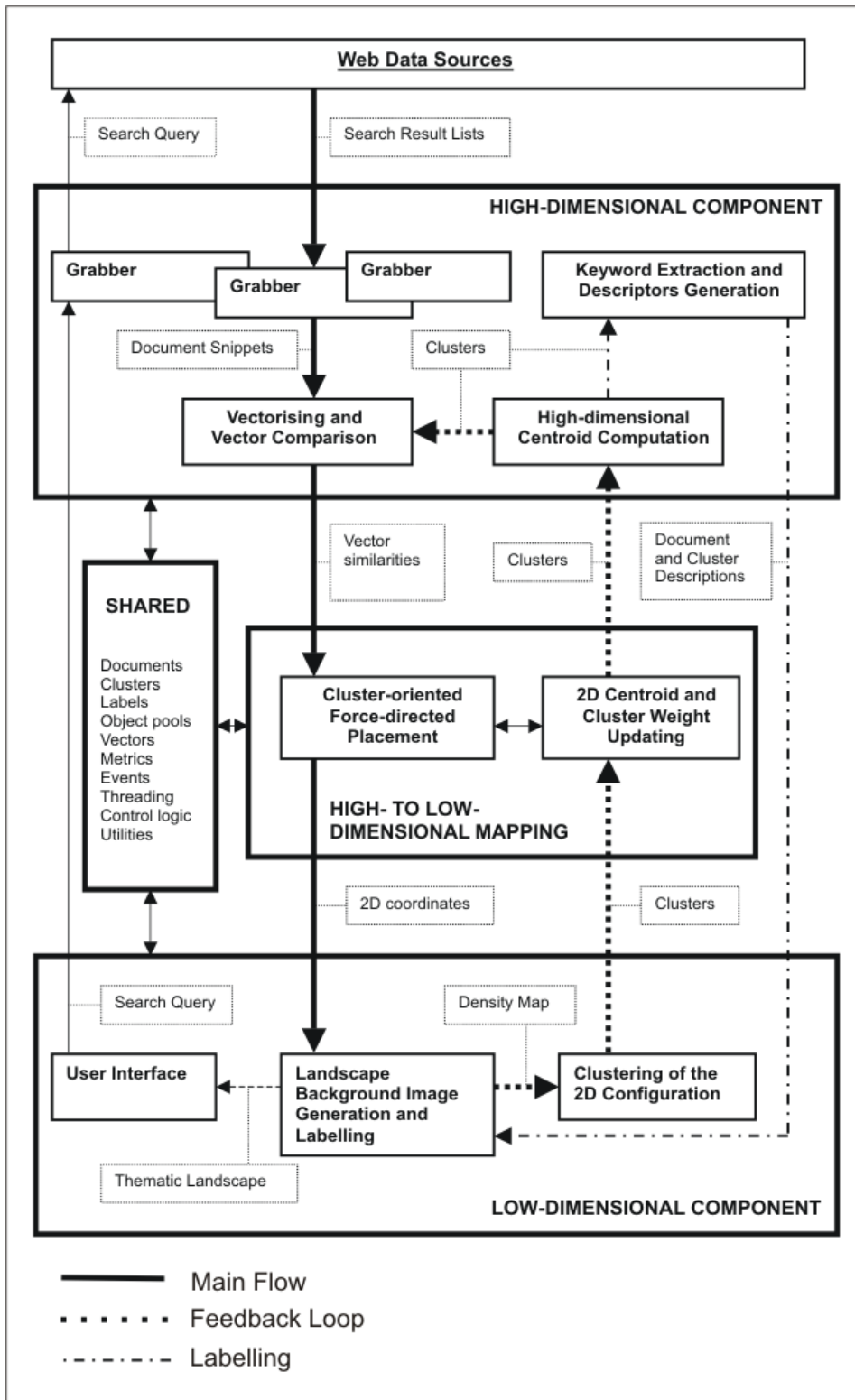


Figure 3.4: The system architecture of WebRat.

3.2.5 Integrated Ordination and Clustering Algorithm

A selection of most important algorithmic details is described in this Section. Some techniques deemed less relevant for this work, such as reconstruction of uniwords from n-grams needed for labeling, or optimizations of the map image generation, are only outlined (more details can be found in [Sabot et al. 2004]).

In WebRat, a cosine similarity coefficient (3.1) is used to compare high-dimensional document vectors \vec{v} and compute a similarities between them. Topical similarities between pair of documents ($sim(d_i, d_j) \in [0, 1]$) are needed by the ordination algorithm to compute 2D document positions $\vec{p} = (x, y)$. A force-directed placement algorithm was chosen as ordination method because it produces visually appealing layouts and because it is incremental - new documents can be added to an already computed 2D configuration without disrupting the layout and without the need to recompute the layout from scratch. The algorithm iteratively computes document positions by letting each document interact with all other documents in the set in every iteration.

$$sim(d_i, d_j) = \frac{\vec{v}_i \cdot \vec{v}_j}{\|\vec{v}_i\| \|\vec{v}_j\|} \quad (3.1)$$

$$dist(d_i, d_j) = \|\vec{p}_i - \vec{p}_j\| \quad (3.2)$$

The force (3.3) between two documents has three components: An attractive component proportional to the high-dimensional similarity between the two document vectors (3.1), a repulsive component inversely proportional to the Euclidean distance (3.2) between these two objects in 2D, and a weak constant gravitational component *grav*. The first component is an attractive component which pulls objects with similar content together with the strength proportional to their similarity. The second component is a short-distance, exponential repulsive component which pushes two objects apart and prevents them from ever coming too close. Excessively high repulsive forces can be clipped, allowing very similar objects to create tight visual clusters of thematically strongly related documents. The third component is a weak but constant gravitational force, which provides cohesion to the layout. It ensures that even very dissimilar objects are attracted to the rest of the data set once they become very distant, preventing outliers from drifting into infinity.

$$force(d_i, d_j) = sim(d_i, d_j)^a - \frac{1}{dist(d_i, d_j)^r} + grav \quad (3.3)$$

It should be noted that this force function does not yield 0 when $dist_{HD}(d_i, d_j) = dist_{LD}(d_i, d_j)$, i.e. the FDP algorithm using this function does not strive

to exactly reproduce the original high-dimensional (HD) distances in low-dimensional (LD) 2D space (a force function that would do that is as simple as $force(d_i, d_j) = dist_{LD}(d_i, d_j) - dist_{HD}(d_i, d_j)$). The reason being, that striving to reproduce HD-distances exactly, would in many cases yield a 2D layout which is not suitable for visualization [Davidson et al. 1998]. Our numerous experiments confirm this fact: distances between various visual clusters may vary by a large margin, and many clusters become too tight for any structure to be discernible. After experimenting with different approaches, the particular force model in 3.3 was chosen, because it is simple and fast to compute, and it produces layouts which reflect high-dimensional relationships faithfully (though not exactly) while at the same time being visually appealing and usable. Another important advantage is that a degree of control over the final layout is provided, in particular:

- Exponent $a \in (0, \infty)$ in the first term is normally 1, but when the average similarity in the data set tend to be too low, its value should be decreased resulting in higher, more discriminative similarity values. If the average similarity tends towards 1 the exponent a should be increased reducing the similarity values. Choice of a is data set dependent and already a very simple heuristic for adjusting this value will has a significant positive impact on both the separation power and the convergence speed of the algorithm.
- Exponent $r \in (0, \infty)$ in the second term has a default value 1, but increasing it will produce tighter visual clusters while decreasing it will result in a more uniform layout. This allows for fine-grained control over the amount of separation in the layout.

2D coordinates of every document are calculated by letting it interact with all other $N-1$ documents in the data set, N being the number of documents inserted in the pool so far, and subsequently averaging the results over all interactions. For example, $Di.x$, a new x-coordinate of object d_i , is calculated with the following equation:

$$d_{i.x} = d_{i.x} + \frac{1}{N-1} \sum_{j=1, j \neq i}^N force(d_i, d_j)(d_{j.x} - d_{i.x}) \quad (3.4)$$

The y-coordinate $d_{i.y}$ is calculated analogously.

Thus, at each iteration of the algorithm a new position is computed for every object. The iteration continues until a termination condition is satisfied. The commonly used termination condition is that of mechanical stress, which is is defined as follows:

$$stress = \sum_{i,j=0, i \neq j}^N (dist_{HD}(d_i, d_j) - dist_{LD}(d_i, d_j))^2 \quad (3.5)$$

However, computation of stress is quadratic in time with N and therefore computationally too intensive for practical purposes. Also, the above definition of stress is not applicable to the used force function (3.3) directly, because it does it is not designed to reproduce high-dimensional distances exactly. Therefore, a more light-weight, adaptive condition is used, which can be summarized as: execution terminates when all documents have been retrieved, and when object positions have stabilized sufficiently, i.e. the average object speed has sunk under a certain threshold.

The time complexity of the described approach is $O(N^2)$ per iteration. As the size of the pool grows comparing every object to all others becomes very slow. To overcome this limitation the implemented FDP algorithm uses two important optimizations to improve performance and enhance the layout quality: document skipping and clustering.

Document skipping optimization, which proved to be very useful in incremental environments, involves tracking of speeds on a per document bases. Documents which have been processed by the layout algorithm for a longer period of time, may have reached a (possibly temporary) stable position. Their positions need not be recomputed in each FDP iteration. In effect, they can remain dormant for a number of iterations, reducing the computational complexity significantly. A heuristic (based on testing and experience) determines when the movement of a document can be considered as "almost still", and assigns a sleep period to the document depending on two parameters: how small its last movement was, and how long the previous sleep period has been.

Use of a clustering technique has a significant impact on time complexity of the algorithm. Once the number of documents which being processed has exceeded a predefined limit, the computed 2D layout is used to identify the coordinates of document density maxima in 2D which are used as seeds for clustering in the 2D space (more details on map rendering, seeding and clustering follow bellow). In the following iterations, FDP algorithm operates in a special cluster-mode: document positions are computed by letting each document interact only with thematically and spatially neighboring documents, i.e. those belonging to the same cluster, and with all other cluster centroids, which represent thematically and spatially distant documents. When computing new document positions each cluster influences its non-members with the weight of all documents it contains. 2D cluster centroids themselves are continuously updated as the documents they contain are moved. If the number of created clusters is approximately \sqrt{N} , then the per-iteration computation complexity

is reduced from $O(N^2)$ to roughly $O(N^{1.5})$. This increases the scalability of the algorithm notably, however when the data set size grows beyond 1000 documents the algorithm becomes too slow for real time incremental computation on a standard desktop PC (2GHz CPU).

Two different strategies for inserting new documents into the cluster partition and the layout were evaluated. The first strategy is to identify the cluster most similar to the document in high-dimensional space, insert the document into this best fitting cluster, update the cluster's high-dimensional centroid, and initialize the document's 2D position with the cluster's 2D centroid position. However, scheme has serious problems with outliers. FDP algorithm has the property of pushing outliers towards the outer limits of the layout. Inserting an outlier into a compact cluster containing very similar document distorted the cluster's high and low dimensional centroids by a large degree. The second strategy is to insert new documents into the layout but not into any cluster. By leaving them "unclustered" for a while, their 2D positions computed by the FDP algorithm are influenced by cluster centroids only. In the course of FDP execution the document's position stabilizes in the vicinity of the most similar cluster - if the document is an outlier it will be pushed by FDP close to an "outlier cluster" (i.e. a topically not very coherent group of outlier documents) on the outer limits of the layout. Once the document set has been incrementally extended with new documents by a pre-defined amount, the whole 2D configuration, comprising old and new documents, is partitioned anew by the 2D clustering routine. During that operation new documents are integrated in the spatially closest cluster, and clusters representation are updated by computing the high- and low-dimensional cluster centroids.

When new documents are added to an already existing group, the positions of documents inside the group experience strong movements as the new documents seek their positions within the group. The groups tend to maintain or even increase their cohesion and only a very small number of documents leave a group and move toward another one. However, those that do so will be integrated in another group once the 2D clustering routine is re-triggered again. This strategy resembles a k-means clustering algorithm with cluster seeds being density hotspots computed by FDP. Seeding and clustering is performed periodically in the 2D space, and the partition is incrementally extended as new documents arrive.

Landscape background image generator computes a topographic map image based on the 2D document coordinates computed by the FDP algorithm. To generate the landscape image an elevation matrix is first computed, with dimensions corresponding to the image resolution (usually 500 x 500). Each

document can be thought of as a small peak having the shape of a spherical cap with a height proportional to its relevance. Objects are placed on the elevation matrix, so that in areas where document density is large the peaks overlap and their heights are superimposed adding to the elevation values of the underlying matrix cells. In the resulting matrix every cell contains the height of the accumulated mountain above it.

To obtain a topographic map-style image the algorithm writes color values into pixels depending on the height in the corresponding cell of the elevation matrix. With the appropriate choice of colors the resulting image resembles a geographic map with peaks at areas where document density is large, while low density areas will be represented as oceans or valleys. A 3D style image can be computed by computing a slope value for the matrix cell using values in the neighborhood cells. Color shade of the corresponding pixel is modified depending on the slope, where a lighter shade is assigned if the slope is positive and a darker one if the slope is negative. For a fixed resolution of the image, the method scales linearly with the number of documents, however to make it reasonably quick for real-time environments the resolution of the used elevation matrix had to be smaller than the image resolution. Further optimizations were proposed and tested by colleagues such as using various lookup tables, using pre-calculated images for document peaks which are alpha blended at runtime, using kernel-based filtering of images for lighting etc.

WebRat detects density maxima positions in the elevation matrix, which correspond to clusters in the landscape image the user sees, by means of image processing. These maxima serve two purposes: they are used as 2D cluster seeds in the accelerated, cluster-oriented version of the force-directed placement algorithm, and they serve as anchor points for labeling the visualization with appropriate keywords. Labels are computed from the highest weight n-grams from the high-dimensional centroids. As n-grams are meaningless to the users, the words from which the n-grams originate are computed using a statistical mapping technique developed by team colleagues.

It should be emphasized that a unique feature of WebRat is the feedback loop created between the low-dimensional and high-dimensional representation of the documents, by the FDP algorithm driving the density maxima detection and clustering, while the clusters are fed back into FDP to improve its time complexity.

3.2.6 Evaluation

WebRat has been subjected to evaluation using an on-line questionnaire after a two-week test phase. For assessment of acceptance of WebRat as a

search engine, as well as for obtaining information about features that could be improved a simple questionnaire was composed. A group of seven subjects recruited at the Know-Center was asked about several aspects concerning data sources to be searched and the handling of the tool. Subjects had to answer questions on a seven-point scale as well as open questions.

A summary of the results is as follows: Subjects were asked about the importance of several data sources queried by WebRat. The organization of individual information such as personal documents and emails as well as the organization of newsgroup archives was announced to be most important. Moreover, subjects suggested WebRat to be useful among others to search item pools, diagnostic manuals or clinical records. Regarding search functionalities subjects claimed thesaurus and synonym support to be helpful and missed the possibility for cross media queries. Altogether, they stated the search functionalities to be sufficiently described, the search site to be well manageable, labels to be significant, navigation functionalities to be sufficiently described, and results easy to interpret. Concerning result functionalities subjects missed a back button and an indication about how much time is left until the search process has been completed.

It can be concluded that WebRat was found to be most helpful when searching a domain one is not familiar with, since it gives a good overview and hence an entry point to the domain. Moreover, test users positively judged that WebRat gives insight into the vocabulary used in the searched knowledge domain as well as how search results relate to each other. Although superficial, these results provided valuable input which already resulted in some improvements in the user interface.

3.2.7 Other Applications

The characteristics of environmental information make it a challenging field for search engines and query refinement tools. Environmental information is typically made up of a variety of different data types, and is typically enriched with meta-information. The environmental context is saturated with abbreviations and multiple meanings of words, rendering the snippet information returned by standard web queries mostly useless. WebRat was applied as a retrieval tool for querying environmental data catalogs (Austrian and German Environmental Data Catalogue: UDK, www.umweltdatenkatalog.de) [Tochtermann et al. 2002] and [Tochtermann et al. 2003]. Meta-information returned by these systems was incorporated and given priority compared to snippet information. As environmental metadata is usually of high quality WebRat was able to deliver higher quality clustering and visualization.

not having any nuclear power plants, is mainly concerned with civil protection. Therefore the query reveals topics at the bottom of Figure 3.5 such as "International Cooperation" or "Bezirkshauptmanschaft" (an Austrian state administration unit which, among other tasks, deals with the protection of Austrian citizens). On the other hand, results from the German UDK cover more technical aspects such as atom absorption spectroscopy and circulation processes (both in the top left corner), laws for radiation protection (middle) and nuclear crime (middle right). Clearly the layout splits into a technical part in the top-left, which is covered mainly by the German UDK, and a jurisdiction and administration part in the middle and bottom parts of the layout, which are covered by both services. There are also a few overlapping regions in the middle-top which are related to ground and radioactive waste, issues concerning both Austria and Germany.

Hyperwave Information Server (HIS) [Hyperwave IS/6 2011] is a knowledge and document management system which stores documents and other objects in a hierarchically organized repository resembling a file system directory structure. The standard Web-interface displays a tree representation of the hierarchically structured repository on the left side, and a list of documents in the currently selected collection on the right side. However, this conventional view does not offer the possibility to get an overview of the knowledge contained in all documents and sub-collections belonging to the current collection, not does it display thematic relationships between these knowledge entities. Hyperwave Web-interface was extended with the WebRat to provide insight into complex thematic relations in the repository and to aggregate the knowledge present in several hierarchy layers to provide an overview to the user [Kienreich et al. 2003a]. The integration of WebRat into Hyperwave Web interface was accomplished by integrating a WebRat-based Java applet in distinguished HTML page on the Hyperwave server - the Collection Head. When a collection is selected in the tree view, WebRat recursively scans all documents present in that collection and its sub-collections, and processes them on-the-fly to generate a visual aggregation of the knowledge present in the chosen sub-hierarchy. In this way the WebRat gives an topical overview and, in addition to hierarchy navigation via the collection tree, provides means for topical exploration and navigation of the documents. That provides a benefit in cases when the user is unfamiliar with the document corpus or when the hierarchical structure of the repository does not reflect topical structures within the corpus.

3.3 Visualizing Hierarchical Document Collections

InfoSky [Andrews et al. 2002] is a system for exploration of large, hierarchically organized document collections, providing a visual representation capable of providing an overview of the whole data set at once, and a deeper, more detailed insight at any level of the hierarchy. It employs a metaphor of the night sky viewed through a telescope, where topically similar documents are placed close to each other and visualized as stars, forming galaxy-like clusters with distinct, recognizable shapes. Hierarchy of collections is visualized as nested polygonal areas whose size is a measure for the number of documents they contain. Navigation through the hierarchy is animated and provides a seamless zooming transitions between the overview and detailed views at the lower hierarchy levels. Labels showing collection names and a summary of their content are displayed to provide orientation. They are dynamically displayed and hidden during navigation, automatically adjusting to the chosen level of detail (i.e. zoom level). Searching is supported through highlighting of hits in different colors allowing the user to immediately see the distribution of hits over the hierarchy and correlations of different searches. The layout algorithm employs a combination of force-directed placement and Voronoi area subdivision, and exploits the hierarchical structure to achieve scalability.

An important feature of InfoSky is that it scales to large data sets, something not possible with the previously described system, the WebRat. However, it should be noted that the algorithms for computing the visualization geometry require a hierarchically organized structure, which is exploited to achieve scalability. "Flat", unstructured data sets are not supported by the system.

My contribution to the development of InfoSky are as follows: design and implementation of the user interface component and the client server architecture, significant contributions to the scalable layout algorithm, and involvement in the usability testing of the interface. I am registered as inventor, along Frank Kappe and Wolfgang Kienreich, on pending EU and US patents: EU Patent Application Number 020077426 (5.4.2002), US Patent Application Number 60/376474 (29.4.2002). InfoSky system was developed for Hyperwave AG [Hyperwave 2011] by Know-Center GmbH [Know-Center 2011], in collaboration with the Institute for Information Systems and Computer Media [IICM 2011] of Graz University of Technology. The system was developed over a period of three years with yearly deliveries of new, improved versions [Kappe et al. 2003], [Andrews et al. 2004].

3.3.1 Requirements

Using structures to organize large amounts of data is common in information systems. User understand hierarchical organization of things from everyday life, which makes exploring and navigating hierarchies more familiar. As a consequence hierarchical structures are one of the most commonly used ways of organizing data, such as for example in file systems. Exploration of hierarchically organized data sets by following parent-child relationships allows users to quickly narrow down the scope and access data even in very large collections. However, typical hierarchy representations, such as tree widgets, do not communicate any other properties in the data other than the hierarchical organization. Information on size, topical similarity or an overview of the data set require the use of additional components.

Requirements on InfoSky were to eliminate these shortcomings and deliver the following additional capabilities:

1. Scalability: visualize large, hierarchically structured document repositories containing millions of documents.
2. Topical similarity: Along the hierarchical structure, topical relationships in the data set should be represented, all within a single visualization.
3. Size: Estimation of the amount documents present in branches of the hierarchy should be supported through visual properties.
4. Overview and detail: Provide both a global overview and a possibility to seamlessly navigate down to the leaf level for a detailed local view.
5. Unified frame of reference: Use a metaphor which promotes visual recall and recognition of features, and provides a single, consistent view for all users to promote communication and collaboration.
6. Exploration: Provide simple, intuitive facilities to browse and search the data set.

TreeMap is visual representation 2.4.3 which comes closest to InfoSky, however TreeMaps can not deliver all of the requirements listed above. While TreeMaps can visualize large hierarchies they do not display leafs, thus lacking the possibility to provide an overview of the whole data set and deliver detail all the way down to leaf level. Also, due to use of rectangular areas, conveying both size and relationships, while a the same using all of the available area is, subject to significant compromises in TreeMaps, often leading to rectangles with extreme height to width ratio.

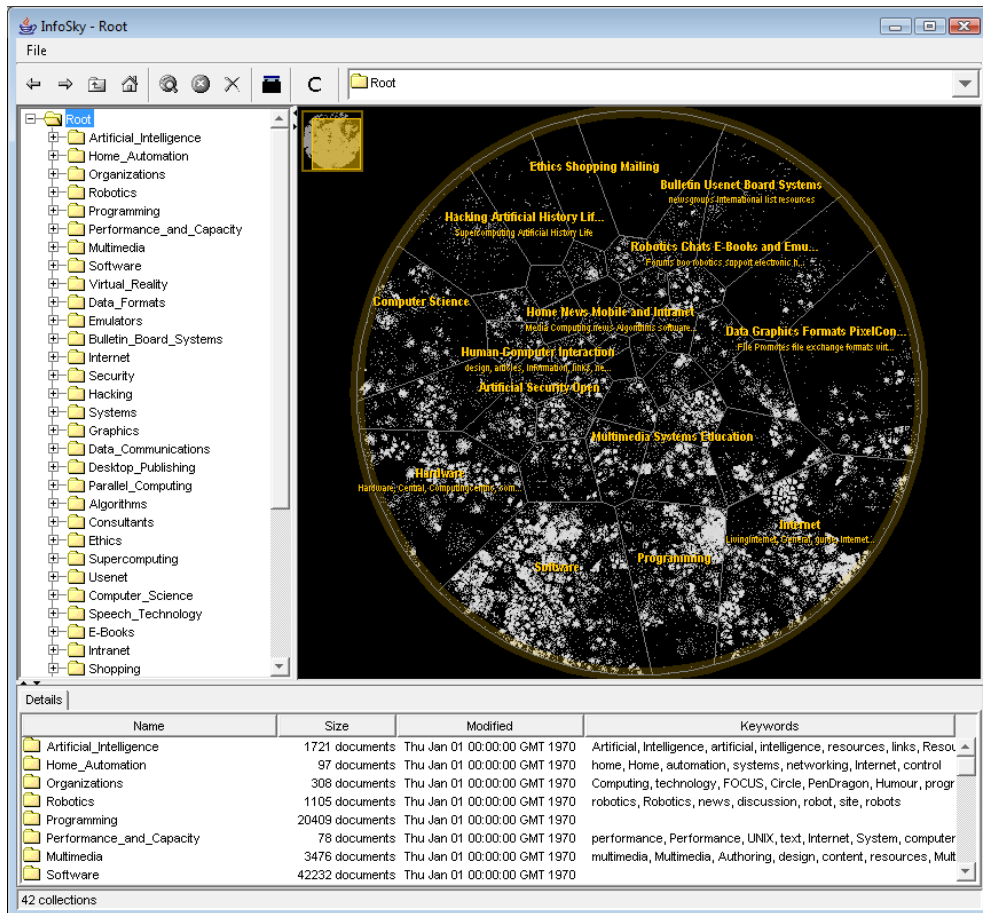


Figure 3.6: InfoSky user interface consisting of a visualization component, a tree and a table, all showing the top level collections.

3.3.2 Visual Interface

InfoSky requires that the document repository is organized as a hierarchy of collections, where each collection can contain documents and further sub-collections. Each document and collection can be members of more than one parent, i.e. the system is not limited to tree structures but can also handle directed acyclic graphs. To visualize the data set InfoSky visual employs the metaphor of a night sky viewed through a telescope. Collections are visualized as nested polygonal areas, which contain further areas (sub-collections) and document visualized as stars. At every level of the hierarchy both collections and documents are positioned in such a way in the 2D space that topically similar object are placed close to each other, while dissimilar ones are positioned far apart. Also, the size of collection areas and the density of stars provide the

possibility to visually estimate the size of a hierarchy branch. The resulting distribution of points and areas resembles stars organized into constellations and galaxies, where zooming in reveals deeper and finer structures, similar to increasing magnification when exploring the sky with a telescope. The direction the telescope is pointing to can be modified to shift the focus and show other regions of the visualization.

Figure 3.6 shows the InfoSky user interface visualizing a collection of 149,195 files in 7643 hierarchically organized collections from the Dmoz [dmoz 2004] "Computers" hierarchy (retrieved in September 2004). The user interface consists of five main components:

- InfoSky visualization, on the right hand side, shows an overview of the data set. Magnification level and direction of the telescope are automatically adjusted to provide optimal viewing size for the currently chosen collection. A small map in the right upper corner reveals which part of the visualization is currently visible on screen.
- A tree on the left hand side shows the hierarchy, with the currently chosen collection ("Root") highlighted.
- A table on the bottom shows metadata, such as size, date and keyword, for direct children of the currently chosen collection.
- A tool bar on the top of the window offers navigation and searching functionality, and provides an address box showing the current location in the hierarchy.
- A status bar at the very bottom reveals the child count (documents and collections separately) for the currently chosen collection.

3.3.3 Searching and Highlighting

Searching and highlighting of search hits within the visualization unveils the distribution of hits over the collections. In Figure 3.7 three searches have been executed: "linux" shown in magenta, "windows" shown in green, and virus shown in red. While the two big clusters seen on the lower left are obviously the "Linux" and "Microsoft Windows" collections (also seen in Figure 3.8 on the bottom), it is interesting to see how search hits distribute over other collections. For example, in the "Consultants" collection (center of the image) "Linux" is obviously mentioned more often than Windows, while the largest cluster of "Virus" related hits can be found in the "Security" collection.

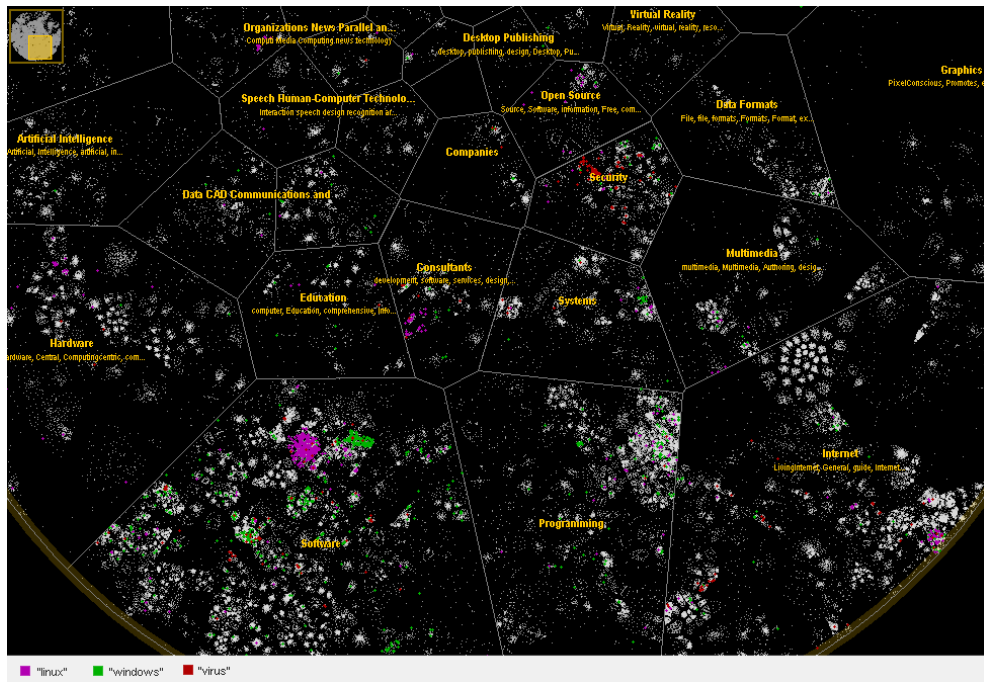


Figure 3.7: Search results for "linux" (magenta), "windows" (green), and "virus" (red).

3.3.4 Navigation

Navigating the hierarchy is as simple as clicking on the labels which cause the system to automatically shift focus to the chosen region and magnify it to optimal viewing size. When navigation deeper in the hierarchy more details are revealed: areas and labels representing collections at deeper levels of the hierarchy will be shown providing a level of detail appropriate for the chosen zoom-level. Figure 3.8 shows an example where, starting from the root collection, the user navigates deeper clicking first on "Software" and then "Operating Systems" collections. Support for view coordination in InfoSky is provided for navigation: the user can choose a collection in the visualization, tree, table of the address bar, and all other views will adjust accordingly. Free navigation in the visualization is also supported to provide fine-grained control of magnification factor and telescope direction. The user can zoom in and out using the mouse wheel, and pan by dragging while keeping the left mouse button pressed.



Figure 3.8: Navigation the hierarchy collections "Software", "Operating Systems", "Linux" (highlighted with gray overlay).

3.3.5 Algorithms

InfoSky is a server-client system implemented in Java. Given a hierarchically structured document set the geometry of the visualization is entirely computed on the server stored there on the disk. When a client connects to the server all 2D document coordinates transferred immediately, but the rest of the geometry and metadata is delivered only on demand while the user is navigating the hierarchy. This allows the client to immediately provide an overview of the whole repository, while the amount of transferred data remains within limits even for large data sets. Another advantage of dynamic loading is, that the amount of geometry and metadata present on the client at each moment is only a fraction of the geometry and metadata for the whole hierarchy: besides 2D document positions for the whole data set, only a single hierarchy branch is present on the client. As a result, even for data sets including more than a million documents and tens of thousands collections, client memory consumption remains low and rendering performance sufficient even for small desktop machine.

The algorithm executed on the server to generate the geometry of the hierarchy performs recursively from top to bottom in such a manner, that at each moment only one parent collection and its children, including documents and sub-collections, are loaded in memory. Provided there are no collections with excessively large number of children (i.e. max a few thousands), the algorithm can compute the geometry for millions of documents within several hours on a small server or even a desktop machine. The algorithm has two main phases:

1. The first phase, which proceeds in a bottom up fashion, begins by parsing the documents of a collection, transforming them into a term vector representation and adding them together to compute the collection centroid. Document vectors and the collection centroid are stored on the disk, and the process continues by moving one level up in the hierarchy, where the centroids of parent collections are computed by adding their children's vectors. Note: if a collection contains both sub-collections and documents, an additional synthetic sub-collection is created to hold the collection's documents (i.e. the hierarchy is transformed so that each collection contains either sub-collections or documents). The first phase is completed when centroids of top level collections have been computed and stored.
2. The second phase recursively processes the hierarchy in a top down manner, and uses collection centroids and document vectors computed in the first phase to generate the 2D geometry. For every collection, beginning

with the root, following steps are executed:

- (a) Ordination: Centroids of the collections's sub-collections (or documents) are positioned in the 2D space so that topically similar objects are placed close to each other and dissimilar ones are placed further apart. The ordination is computed using a force-directed placement algorithm with a force model defined in equation 3.3. Stochastic sampling using neighbor and random sets, along the lines of [Chalmers 1996], was used to improve the running time and add a source of jitter to reduce the probability of getting stuck in local minima.
- (b) Inscribing: The layout computed in previous step is first normalized to occupy $[-1, 1]$ 2D space, with the origin o in $(0, 0)$, the square bounding the normalized space being B . Then the layout is inscribed into the convex polygonal area A of the parent collection using the following simple geometric transformation:
 - i. Point c , being the center of gravity of the polygon A , is aligned with the origin of the normalized $[-1, 1]$ space o .
 - ii. For each point p_i in the normalized space an infinite ray r_i is cast from o to p_i intersecting the polygon A at point a_i , and the square B at point b_i
 - iii. The inscribed point p'_i is placed on r_i so that $dist(c, p'_i) = dist(o, p_i) \frac{dist(c, a_i)}{dist(o, b_i)}$
- (c) Voronoi area subdivision: The collection's polygon is subdivided into nested sub-polygons by assigning a polygonal area to each sub-collection. This is performed by applying a variant of additively weighted power Voronoi subdivision [Okabe et al. 2000], with inscribed sub-collection centroid positions as control points. The resulting size of each created area is related to the total number of documents contained in the subtree starting at that sub-collection. Note: if the collections contains documents, i.e. the bottom of the hierarchy has been reached, then no area subdivision is performed and the recursion stops.

3.3.6 Evaluation

InfoSky was developed over a course of more than tree years. After about a year and a half a first complete prototype was ready and was tested in a preliminary usability study [Andrews et al. 2002]. This study revealed various deficiencies which were fixed in the next two versions of the tool. A



Figure 3.9: InfoSky evaluation setup.

second evaluation was performed on the final InfoSky version in spring 2004 [Granitzer et al. 2004]. Both tests consisted of formal experiments which were performed in an environment as shown in Figure 3.9. For both tests users were given a short introduction to the features of the tested user interface and received a two minute training to become familiar with the interactivity. After the test each user was interviewed to collect additional feedback. The test were recorded on video providing the possibility to analyze user reactions extract the exact timings for each task. For all tasks the application window was maximized to make use of the full screen area, and the search functionality was disabled.

3.3.6.1 Preliminary Evaluation

In the first formal experiment performed in 2002 a comparison between the InfoSky visualization and the standard tree widget was performed. Users could perform tasks in the visualization or in the tree, use of both components simultaneously was prohibited. The test dataset used in the test consisted of 110.000 newspaper articles from the German Sueddeutsche Zeitung. Two

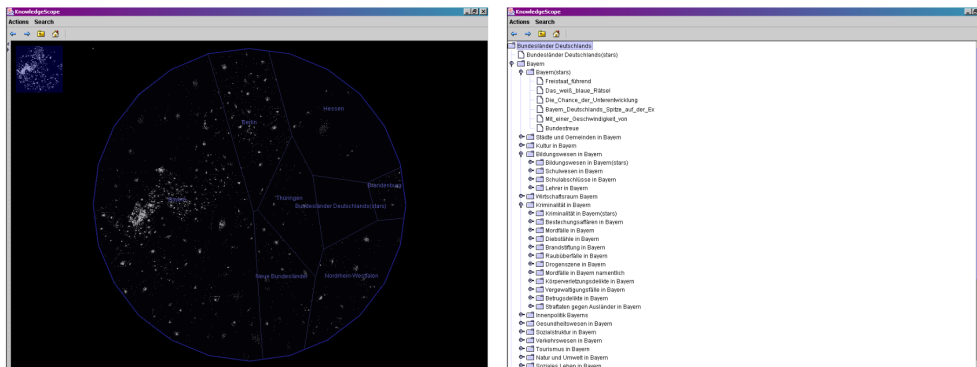


Figure 3.10: InfoSky test conditions: V on the left and T on the right.

sets of tasks, A and B, were formulated consisting of five pairs of equivalent tasks. Pairs of tasks are equivalent in the sense of their complexity, i.e. their solutions lay at the same hierarchy depth and involve approximately the same number of inspections and choices. Eight test users were divided randomly into four groups of two. Half of the users began with the visualization and then used the tree, condition V, the other half used the tree view first and then the visualization, condition T (see Figure 3.10). Within each condition two users began with the task set A and the other two with the task set B.

Without going into details of each task, they can be summarized as follows (ordered in ascending level of difficulty):

- Tasks 1 and 2: Find a sub-collection within a collection, with task 2 being slightly more difficult to complete.
- Task 3: Estimate which of the two sub-collections contains more documents.
- Task 4: Give the exact number of documents in a sub-collection.
- Task 5: Find a document on a particular topic within a given subtree.

Evaluation revealed that the tree performed better than the visualization on average, and that the results were statistically significant ($p < 0.05$, paired samples t-test, 39 degrees of freedom, $t = 3.038$). Besides users' familiarity with the tree, following issues could be identified as reasons for the result:

- A portion of Voronoi polygons, typically those in the center of each collection, were too small.

- At the bottom of the hierarchy, where documents are shown, labeling techniques is not adequate. Due to potentially large number of documents only titles of documents near to the mouse pointer were shown in what was perceived by users as a list. Two problems were associated with this labeling style:
 - When more than a few document titles were displayed the display still became cluttered with text.
 - Users tried to scan through the "list" which changed the moment the mouse was moved. This was perceived by users as "jumping around" of document titles.

From interview results it was clear the users felt more familiar with a tree view than with the visualization, although they liked the overview provided by the visualization and could imagine themselves using it for exploration of a document corpus. Users also indicated that a combination of tree and visualization could be a powerful solution exploiting the advantages of both representations.

3.3.6.2 Main Evaluation

InfoSky visualization and user interface were subsequently redesigned under consideration of the above evaluation results. This included the following changes:

- Voronoi subdivision and force-directed placement algorithms were modified and tuned in order to prevent creation of excessively small collection polygons.
- Labeling of documents in the visualization was completely overhauled:
 - Keywords are extracted for both collections and documents. A label now consists of an upper row showing the title and a lower row showing the keywords. This is useful in cases when titles do not provide a topical description of the object.
 - A zoom-factor sensitive label merging algorithm avoids clutter by ensuring that the amount of text displayed on screen is limited. Displayed labels are always adjusted to the current level of detail.
- The full list of children (i.e. documents and sub-collections), including metadata, belonging to the currently chosen collection is shown in a table.

- A navigation bar is added on the right of the tool bar, showing the currently selected collection (i.e. location) in the hierarchy. A drop down menu reveals collection's parents up to the root collection.
- A variety of smaller issues and annoyances was corrected, interactivity was polished and animated transitions in the visualization became smoother.

The resulting interface, shown in Figure 3.6, resembles typical file system navigation tools, such as Windows Explorer, in look and feel. Design of the interface was driven by the wish to provide users with an interface they are familiar with, extended by the InfoSky visualization.

The test setup differed from the previous one, as in addition to testing the visualization and tree separately, combination of the visualization and tree was also tested. Other components, such as the tree and the navigation bar were available to test users all the time (with the exception of searching). Three sets of tasks, A, B, and C were formulated consisting of six triples of equivalent tasks. As in the previous test, the tasks were designed to be equivalent among the three sets in the sense that their solutions lay at the same level of the hierarchy and involved inspecting approximately the same number of choices at each level. Nine test users were recruited for the study and divided into three groups of three. Users of the first group began with the visualization (condition V), then used the tree (condition T), and finally executed the tasks with both views available (condition VT). Other two groups used alternating ordering of test conditions. Data set used for testing consisted of 80.000 newspaper articles from the German *Sueddeutsche Zeitung*.

Tasks performed by the users included the following:

- locating a collection within the hierarchy,
- locating a document within the hierarchy,
- estimating and comparing the number of documents contained in two collections,
- counting the number of documents which are direct children of a collection,
- for a given document, locating and counting the number of topically similar documents within the same collection.

It should be noted that majority of the tasks in this usability study were significantly more complex than tasks in the first study, and demanded navigation

into deeper parts of the hierarchy. Also, some tasks, such as locating topically related documents, were not performed at all in the first study.

Statistical evaluation of the times required to performed the tasks indicate that there are significant differences between the three test conditions. Note that when a test user could not complete the task within a predefined maximum time, a time-out was recorded and this result was not included in the statistical evaluation. A summary of the results is as follows:

- Combination of the visualization and tree (condition VT) performed significantly better than the visualization alone (condition V). Statistical results are confirmed by comments given by users in the follow-up interviews, where the value of the visualization was recognized as an overview and orientation tool useful for avoiding getting lost when navigating deep in the hierarchy.
- Combination of the visualization and tree (condition VT) still did not perform as good as tree alone (condition T).
- However, when using the tree browser (condition T) seven time-outs were reported, indicating that the task could not be solved in reasonable time, but at the same time only four time-outs occurred when using the combination of visualization and tree (condition VT). Majority of time-outs were reported when users got lost in the hierarchy and were unable to find a path of navigation towards the desired destination. This is a strong indication that the visualization is useful for understanding both the overall structure of the hierarchy and the context of the current position.
- When using visualization only (condition V) or tree only (condition T), about half of all time-outs occurred in task where the user was asked users to navigate to an item deep in the hierarchy. However, in combined view (condition VT) only one user reported a time-out. This further underlines the having both provides advantages for navigation of the hierarchy.
- In interviews the users consistently described the combination of visualization and tree as more satisfying than any of the two alone.
- Issues with labeling were rarely reported. The overhauled labeling strategy removed the majority of problems identified in the first study.
- Far less small usability issues and annoyances were identified than in the first study. Obviously, this version of InfoSky is significantly more mature and more polished than the first one.

The conclusion is that the combination of visualization and tree is a promising approach which was well accepted by the users. It delivered results more often than the tree alone, but was still slower on average in all tests. However, this might be the result of the fact that users are far more familiar with a tree. It is likely that for tasks where context and overview can be exploited, the combined view would come closer to the performance of the tree view, given the comparable amount of training and experience. On the other side for users which are familiar with the hierarchy the visualization will probably not offer any advantages.

3.4 Temporal Visualization

Addressing analysis of temporal aspects, in addition to topical relationships, is a frequent requirement, for example in understanding the history of search result sets. For some data types, such as audio and video, temporal information is not just an additional metadatum but an essential ingredient of that data. This section introduces two project which apply visual methods for analysis of temporal aspect of the data set.

3.4.1 Temporal and Topical Analysis of Search Results

In the OnAir project [Kienreich et al. 2005a], [Kienreich et al. 2005b] an intelligent retrieval system was developed including a prototype client solution for visual presentation of search results. It was implemented in cooperation with the APA DeFacto [APA-DeFacto 2011], a subsidiary of the Austrian Press Agency which manages the largest media archive in Austria. Although a significant effort was invested into retrieval technology, this section focuses on visualization features and related algorithms only. My focus in the project was predominantly on ordination and clustering algorithms, and also included conception of the user interface.

An important feature of OnAir is that in addition to visual analysis of topical relationships, it introduces analysis of temporal development of topical clusters. Exploiting of rich metadata present in the search results was also one of the goals of the project. However, it should be noted that OnAir is targeted at non-expert users. Therefore, visualizations employed in OnAir are ment to provide a quick (ad-hoc) overview of a smaller amount of highly relevant search results, they are not designed for into-deep exploratory analysis of large data sets. Also, to avoid overloading the user with a complex user interface, the interface permits the visualizations to be viewed only one at a time, and does not provide support for coordination between them.

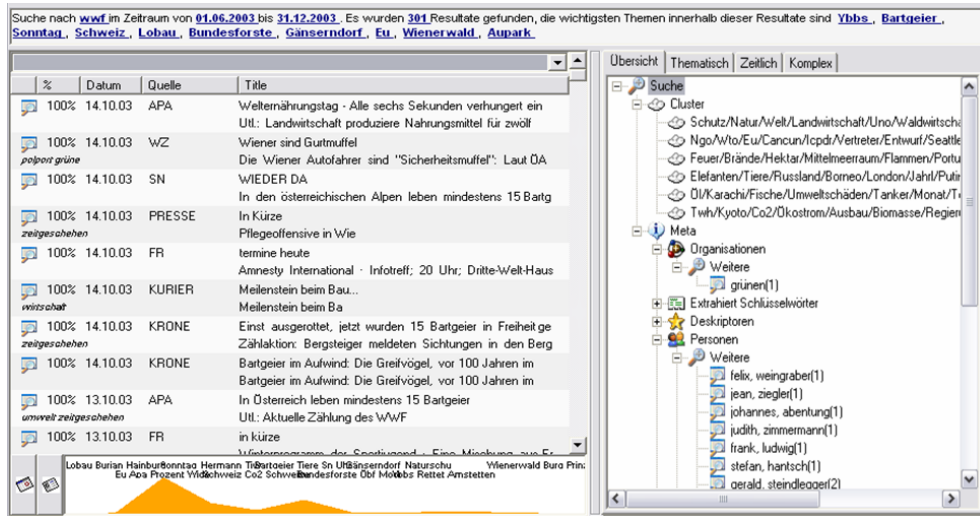


Figure 3.11: OnAir user interface.

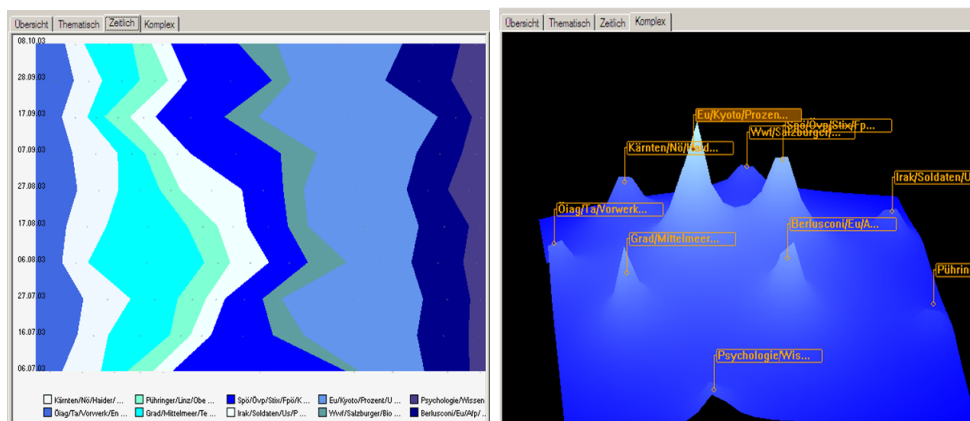


Figure 3.12: Visualization of cluster temporal development (left) and topical relationships (right).

The system was implemented in a full 3-tier architecture including:

- a database backend and retrieval engine
- a middleware component, implemented in Java, provides ordination and clustering algorithms, metadata collection and aggregation, and control logic
- a user interface component, implemented in .NET, provides a search result list and multiple visual components

News returned as response to a search query are analyzed on-the-fly and, in addition to a ranked list presentation, are shown in several visualizations. Visualization and corresponding on-the-fly analysis techniques were tuned to process a relatively small amount (typically up to several hundreds) of most relevant search hits as quickly as possible, so that the user can begin exploring them as soon as possible. Scalability to large data sets was not the goal of the project. With this in mind the following algorithms were applied:

- Hierarchical agglomerative clustering (complete link) was used to extract topical clusters from search results. Although not scalable, this methods produced topically coherent clusters and proved fast for small data sets. A k-means variant was used only for data sets above a certain size.
- Force directed placement was used as ordination algorithm. For speed reasons single search results were not processed, only cluster centroids were projected to provide an overview of major topics and their relationships.
- Rich metadata present in search hits, such as persons, organizations, locations, and various descriptors, were collected and aggregated, and presented as a tree of faceted metadata.

Figure 3.11 shows the OnAir user interface consisting of following components:

- Top: A summary of the search query.
- Left: Search result list displaying the title, source, date and relevance of each result.
- Bottom-left: a simple visualization of clusters sizes
- Right: A tab pane containing several views:

- A tree view showing the clusters and the faceted metadata tree (visible in Figure 3.11). In faceted metadata tree (starts at node "Meta") nodes in the first level represent the type of metadata, such as organizations, descriptors or persons. Hanging on the metadata type nodes are the particular instances of that type, such as names of particular organizations or places. Clicking on these nodes filters the search result list to show only hits containing that particular instance.
- A simple graph view of the clusters (not shown).
- A visualization of temporal developments of topical clusters (shown in Figure 3.12 on left). The visualization resembles closely the ThemeRiver representation (see Section 2.5.3), whereby cluster sizes are relative (as percentages), not absolute.
- A landscape visualization (shown in Figure 3.12 on right), showing topical clusters, their sizes (visualized as height) and their topical relatedness (conveyed by spatial proximity between them).

3.4.2 Visualization of Communication Patterns in Meetings

Multimedia data has a complex structure consisting of inter- and intra-document relationships (for example text refers to images in the same document, an audio references an external documents etc.). To exploit the potential of these implicitly present relationships methods for semantic extraction and cross-media exploration need to be devised. In the MISTRAL project [Sabot et al. 2005] an architecture for measurable, intelligent and reliable semantic extraction and retrieval of multimedia data (MISTRAL) was developed. The system extracts a variety of semantically relevant metadata from one media type and integrates it with concepts derived from other media types. Semantic extraction is the ingredient which differentiates MISTRAL from approaches which predominantly focus on low-level feature extraction.

In the context of the project several client applications for semantic retrieval and analysis of multimedia data were built. My contribution to the project was primarily on a visual application for discovery of communication and conversation patterns in a meeting. Visual Conversation Analysis (VCA) [Sabot et al. 2007] is a MISTRAL client application for visual browsing and analysis of communication patterns between participants of a meeting. A prominent feature of VCA is that it introduces several components supporting visualization and browsing of temporal data, which are integrated into a complex user interface using a framework for view coordination.

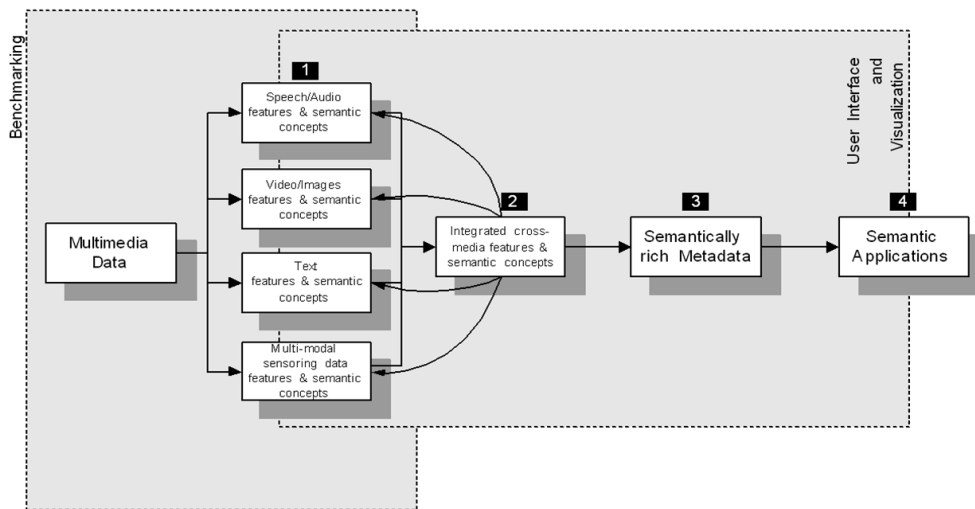


Figure 3.13: MISTRAL modules.

3.4.2.1 MISTRAL Overview

Due to resource and technological considerations developing of a fully generic system would be overly complex and impossible to realize in a realistic amount of time. In order to focus the development of extraction algorithms the project targets extraction of semantic data from meetings. "Meetings" application domain concentrates on video materials showing one or more persons talking or discussing a particular subject. Additionally to recorded video materials accompanying documents, such as PowerPoint slides, are also considered and analyzed by the system.

Main building blocks of MISTRAL are shown in Figure 3.13. The capabilities of main modules are briefly described without going into algorithm and implementation details:

- Video unit detects, tracks and recognizes multiple person faces from images and recorded video data in real time.
- Audio unit processes multi-channel audio signals recorded by a linear microphone array. After estimating and suppressing background noise it detects voice activity, speaker's position and gender, and performs speaker indexing (i.e. speaker recognition).
- Sensory data unit collects user input, such as mouse clicks and keystrokes, and tags them with time stamps and active application information. Also, text data currently displayed by the active application is grabbed for further processing by the text unit.

- Text unit employs text mining and retrieval techniques to extract named entities, topics and concepts, and to provide searching functionality for textual data.
- Multi-modal merging unit fuses features extracted from different modalities (video, audio and text) and produces a combined extraction result in a MPEG7-style [MPEG7 2004] format.
- Semantic enrichment unit provides inference capabilities capable of deriving new semantic concepts from the extracted ones. It also detects and resolves inconsistencies and contradictions resulting from merging the results of extraction units.
- Integration of components is achieved through a shared MPEG7 data structure and a data storage component, including a benchmarking system and client for data access and annotation.
- Applications: five different client applications were implemented for retrieving and browsing results generated by MISTRAL components.

See MISTRAL project homepage [MISTRAL 2005] for more detailed information on the system.

3.4.2.2 Visual Conversation Analysis Tool

Visual Conversation Analysis (VCA) tool is a user interface for visually supported navigation and browsing of meeting recordings, including video, audio and text materials, and extracted semantic metadata. Input data is provided by MISTRAL extraction modules, extended manually using the annotation application. Requirements on VCA were to enable the user to accomplish the following:

- Viewing speaker activity over time to discover when participants were holding a monologue, or were engaging in dialogs and discussions.
- Viewing topics discussed and entities (such as persons or organizations) mentioned over the duration of a meeting. Also included are topics and entities extracted from materials presented at the meeting, such as PowerPoint slides.
- Discover the relationships between speakers and topics, i.e. discover which participants were actively discussing on which topics.

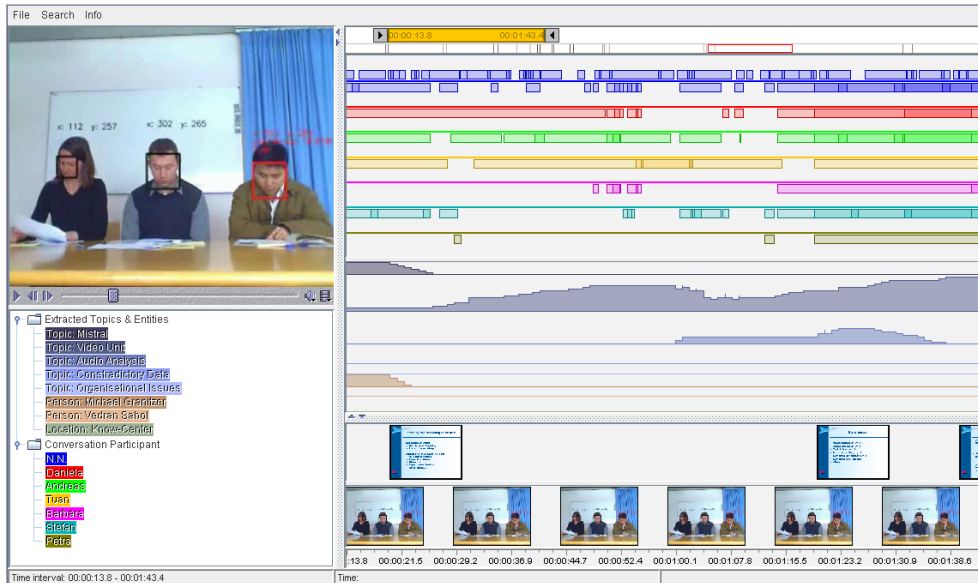


Figure 3.14: Visual Communication Analyzer (VCA).

- View the recording of the meeting in a video player., accompanied by a visual storyboard of the meeting including captured video frames and presented materials (PowerPoint slides).

VCA application, shown in Figure 3.14, is a complex user interface consisting of nine components, with the majority addressing temporal navigation and temporal visualization of data:

- A timeline component, on the bottom-right, displays the time along the x-axis. All components above the timeline share the same temporal coordinates, with the sole exception of the time interval selection bar.
- Interval selection bar, on the top-right, is used for temporal navigation and zooming purposes. Choosing of a time interval, which is equivalent to temporal zooming, is achieved by dragging the scroll buttons on the edges of the scrollbar. The chosen temporal window can be smoothly moved back and forward in time by sliding the scrollbar left or right. All other temporal components (placed underneath the bar) will adjust their content depending on the chosen time interval.
- Search hit bar shows search hits, i.e. events and actions returned by MISTRAL in response to a search query, positioned along the time axis. Position and width of the rectangle depend on the time and duration of the event (one found event is highlighted in red due to a mouse-over

effect). Clicking on a result triggers a smooth, animated navigation to the time interval covered by the event.

- Temporal activity view (on the upper right), which resembles LifeLines representation (see 2.5.2), visualizes presence or absence of actions and events, in this case speech activity of meeting participants. Participant are color-coded, color assignments are visible in the legend (see bellow).
- Temporal intensity view (in the center-right) visualizes the intensity of an activity (or the number of simultaneously occurring events), in this particular case mentioning of a topic or an entity (such as a person or organization) during the discussion. Topics and entities are color-coded, color assignments are visible in the legend (see bellow).
- Component for temporal visualization of time-stamped icons or image thumbnails, places the images at their position along the time axis. Two components are present in the user interface (both on the lower right):
 - The upper component displays PowerPoint slides at the time point when they were shown during the meeting. Clicking on a slide thumbnail will show the slide magnified.
 - The lower component displays the storyboard of captured video frames. Clicking on a video frame thumbnail will make the video player jump to the corresponding position in the video.
- A video player, on the upper-left, plays the recorded video of the meeting.
- A legend, in the form of a tree (on the lower-left), is used to specify the color-coding for the meeting participants, and for the extracted topics and entities. Meeting participant are coded by bright colors, while topics and entities are coded by pastel colors, where different shades of the same color are used for different entities of the same type.

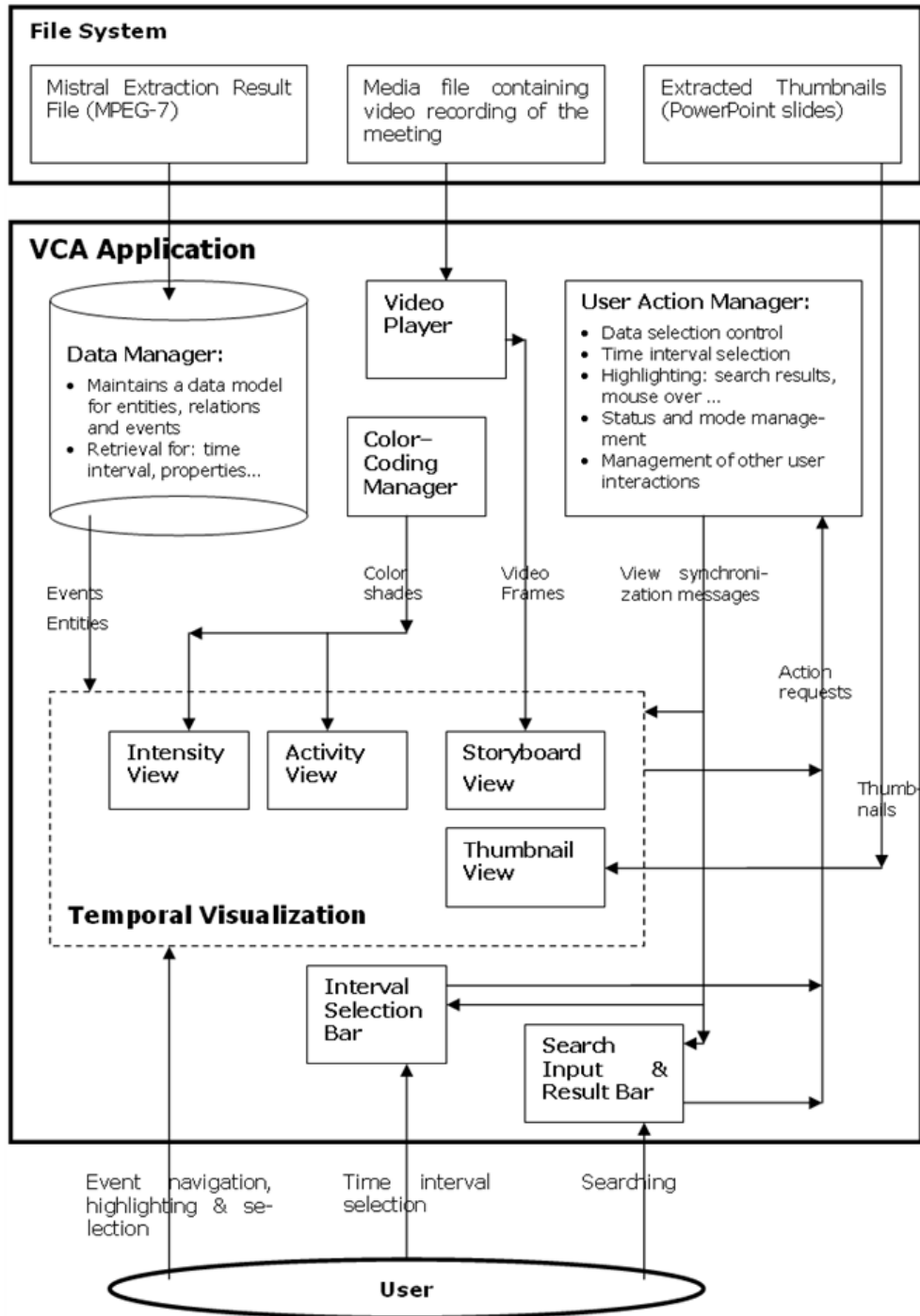


Figure 3.15: VCA architecture.

VCA is a complex user interface with a complex software architecture (see Figure 3.15). To ensure that the GUI behaves in a consistent manner, a framework for coordination of multiple views was introduced and realized along the lines of the model-view-controller paradigm (see Chapter 2.7). To provide fast coordinated filtering, selection, zooming and navigation an event dispatching model based on temporal actions and events was implemented to notify all views that an update in the visual representation is required in response to a user action. A shared data manager ensures that all views maintain a coherent data state and notifies them when the model has changed. The data, including extraction results (in MPGE7 format), recorded video materials and grabbed screen data (mainly PowerPoint slides) are provided by MISTRAL as files (later an optional Webservice interface was added). VCA is implemented in Java and makes use of Java Media Framework (JMF 2.1.1e) for video playback and frame grabbing.

Chapter 4

Algorithms and Visual Techniques

This chapter introduces visual methods and tools supporting exploratory analytical processes in large, weakly structured, complex, dynamic data sets. The described technologies are developed to address goals and requirements defined in Section 1.2.2 (also see Section 1.1 for a more in-depth discussion on motivation).

The chapter begins by providing an overview of the developed visual techniques and components, and by outlining the used approach of building upon my earlier contributions (described in Chapter 3). Each of the developed methods is described into detail over the several following sections. The chapter concludes with sections introducing two prototype applications built by combining and applying the developed visual techniques.

4.1 Approach and Overview

Driven by the needs of application domains such as business intelligence and or media analysis, a set of algorithms and visual techniques were developed targeting expert users and analysts who need powerful tools providing both an overview capability for familiarizing with new data, as well as into-deep analysis functionality. Building upon and extending the results described in Chapter 3 the following strategy was applied to develop visual techniques and tools capable of addressing the above stated goals:

- Combine the landscape metaphor (see WebRat, Chapter 3.2.2) with the night sky metaphor (see InfoSky, Chapter 3.3.2) into a representation for analysis of relatedness (such as topical similarity), which includes

the advantage of both approaches:

- The well-known geographic map representation provides information on the amount, density and cohesion of data elements, and includes "landmarks" aiding recognition and orientation.
 - The night sky visualization displays single data elements offering the possibility to code features and metadata through visual channels such as colors and shapes, and to interact with and manipulate the data set at low level of detail.
- Implement a scalable ordination algorithm applicable on large, unstructured data sets, by adapting the Infosky hierarchy projection method (see Chapter 3.3.5) and combining it with a fast hierarchical clustering algorithm used to generate the hierarchy.
 - Exploit the hierarchical structure produced by the clustering algorithm for navigation in the visualization through the hierarchy of nested areas (similar to InfoSky, see Section 3.3.4), and as a virtual table of contents for unstructured data sets (compare to Scatter/Gather, Chapter 2.2.5).
 - Implement an interactive component for visualization of temporal developments and trends of clusters and faceted metadata, by combining and extending ideas from OnAir (see Chapter 3.4.1) and VCA (see Chapter 3.4.2.2) temporal visualizations.
 - Further support temporal analysis by visualizing changes in dynamic data sets using an animated dynamic topography information landscape (compare to Chapter 3.2.3).
 - Supported by findings of InfoSky evaluation (see Chapter 3.3.6), extend coordination mechanisms from the VCA tool (see Chapter 3.4.2.2) into a full multiple view coordination framework. Employ the CMV framework to combine various visualizations and components into a coherent user interface enabling fused, simultaneous analysis of multiple data aspects (i.e. topical, temporal and rich metadata).
 - Implement proof-of-concept applications demonstrating the capabilities of the developed technologies for addressing the above stated goals.

Executing along the strategy listed above, a set of technologies was developed which includes:

1. A scalable, incremental ordination algorithm which, based on relatedness between data elements (usually topical similarity between documents),

generates a hierarchically organized layout of large, unstructured data sets. Benefits provided by the algorithm are three-fold:

- Extracted hierarchical structure can be used for navigating the data set as a "virtual table of contents".
 - Projection of data elements and hierarchically organized clusters into the 2D visualization space is a geometrical representation which can be used for visual analysis.
 - Incremental processing capability allows for incorporating changes smoothly into an existing hierarchy and layout, aiding the analysis of dynamic data sets.
2. An advanced, scalable Landscape3D visual component provides an interactive information landscape visualization for conveying complex relationships in the data (see Section 2.4.5). In particular the component:
- Provides an overview of the whole data set including structures emerging from the data.
 - Conveys the degree of relatedness between data set elements, and provides clues on the size and cohesion of the emerging structures.
 - Offers hierarchical, level-of-detail aware navigation capability, which is supported by labels describing the essence of the data behind the structure.
 - Provides visual coding of metadata and features, enabling their correlation with the structures.

Additionally, using a high performance rendering subsystem, animation of dynamically changing landscape topography can be employed to visualize changes in the data set.

3. An advanced StreamView component conveying temporal developments of both clusters and faceted metadata categories, allowing for discovery of trends, temporal correlations between clusters, recurring events etc. The component also includes temporal selection and navigation capabilities.
4. A powerful framework for fast coordination of multiple views, which includes a shared coordinated data model, and provides visual property, logical property and navigation coordination. "Fusing" multiple visual components, where each component is specialised for analysis of a different data aspect, into a single coherent user interface enables simultaneous analysis of manifold data aspects. For example, a coordinated

interface consisting of Landscape3D and StreamView visual components enables "fused", simultaneous topical-temporal visual analysis of large document sets.

5. Several standard GUI widgets, such as a tree and table, which were extended with view coordination capabilities. Accompanying advanced visualization components with standard GUI widgets the user is familiar with is an approach which is well accepted by the users (see conclusion of Section 3.3.6.2).
6. Two prototypical demonstrator applications consisting of multiple visualizations:
 - Knowledge Discovery Visual Environment (KDVE) is a client application for analysis of "topical-temporal-faceted metadata" relationships and correlations in large, dynamic text repositories. KDVE is the main demonstrator of visual techniques and algorithms developed in this work.
 - Semantic Mediation Tool (SMT) is an application for visually supported semi-automatic ontology alignment. It uses a subset of the developed technologies (at the current development stage) and has the purpose of demonstrating the applicability of the developed techniques on data types other than text - in this case on semantic information.

The rest of this chapter presents and describes these technologies into detail, while a demonstration of how they can be applied to achieve the goals defined in Section 1.2.2 is available in Chapter 6. All examples use the Reuters Corpus Volume 1 in English language [RCV1 2000] consisting of more than 800000 news documents, except for the SMT prototype which is applied on semantic data (ontologies).

4.2 Scalable, Incremental Ordination Algorithm

The ordination algorithm introduced in this section integrates projection and clustering into a single procedure. It is built by combining and adapting following two algorithms:

- Divisive (top-down) hierarchical clustering algorithm, described in [Muhr et al. 2010], which organizes a "flat", unstructured data set into a hierarchy of clusters.

- Projection and layouting algorithm developed in InfoSky (see section 3.3.5) which generates a similarity layout of a hierarchically organized data set.

My contributions to the resulting scalable ordination algorithm are threefold:

1. The main contribution is the idea and conception of combining and adapting the above two algorithms into a scalable ordination algorithm applicable on unstructured data sets [Muhr et al. 2010], [Sabol et al. 2010b]. Also, I was the lead of the implementation team which supplied various bits and pieces.
2. I am one of the three inventors of the InfoSky visualization system as listed in EU and US patent applications (see Section 3.3), with my contribution being focused on the scalable similarity layout algorithm for hierarchical data sets.
3. Conception and initial implementation of hierarchy generation through recursive application of a k-means algorithm, which includes seeding (such as in [Bradley & Fayyad 1998] and [Arthur & Vassilvitskii 2007]) and cluster split-and-merge strategies (such as [Tou & Gonzales 1974]).

It should be noted that a member of the development team, Markus Muhr, has taken over the further development of the clustering algorithm by introducing and testing various improvements in the areas of seeding, cluster splitting and merging, incremental processing, and label computation. These improvements are not the topic of this work.

The resulting ordination algorithm has several advantages and unique features:

- **High performance and scalability** with regard to data set size and dimensionality. Data sets containing millions of data items described by thousands of features have been successfully processed. Adequate choice and combination of different methods used within the algorithm, and carefully chosen parameters, ensure that each method is employed in a segment where it performs well, and on data subsets which present no obstacle to method's scalability. Interweaving of clustering and ordination into a single algorithm reduces the number of operations and passes over the data set, further improving performance. Also, all processing is performed on optimized data structures in the main memory, which is an important difference compared to the original InfoSky algorithm. The resulting performance allows for on-the-fly application on small to medium data sets consisting of up to several ten thousands data items.

- **Generation of a hierarchical structure from unstructured data** based on similarity of data set elements is accompanied by computing a corresponding hierarchical geometry in 2D space. The algorithm includes techniques for achieving maximum separation between generated clusters at every level of the hierarchy, and for extracting descriptive and discriminative terms for labeling of hierarchy nodes. The resulting hierarchical structure, which is used as a virtual table of contents, is also used to navigate in the visualization space.
- **Dynamic cluster structure on each level** of the hierarchy through splitting and merging of clusters within the given constraints on the maximum and minimum number of elements. The strategy attempts to find the optimal number of clusters at every level of the hierarchy, while at the same time the minimum and maximum bounds ensure that the direct child count remains within a range considered usable for interactive analysis, and that the hierarchy does not degenerate. Values for the minimum C_{min} and maximum C_{max} number of direct children are usually set to 3 and 12, respectively, and are driven by usability considerations: with a typical child count being around 10, scanning of children to resolve further navigation direction appears appropriate. An additional advantage of this strategy is that it keeps the computational costs within well-defined bounds (see below).
- **Capability to deal with dynamic repositories**, meaning that once an initial layout and the corresponding cluster hierarchy have been computed, changes in the data set can be incorporated smoothly, without erratic disruptions, and without the need to recompute everything from scratch. Incremental processing modifies a previously computed configuration and incorporates changes in such a way that the amount of modifications of the result approximately corresponds to the amount of changes in the data. Incremental processing not only reduces computation time, but plays a crucial role when a sequence of landscapes visualizations is used to convey changes in the data: without recognition and smooth transitions users could not follow and understand the changes [Sabol et al. 2010b]. It should be noted that if a new result would be computed from scratch for every change in the data set, it is very likely that this results would not look similar to the previous one at all. The reason for that is that clustering and ordination algorithms, which are basically optimization problems for finding a (local) minimum, are sensitive to initial conditions. A small change in the initial configuration of data may lead to a completely different local minimum.

Assuming that feature vectors for all data set elements have been computed (see Chapter 5.1 for details), and that a root area, usually a rectangle, corresponding to the entire data set is defined, the algorithm recursively proceeds in a top-down manner executing following steps (also see Figure 4.1):

1. **Preprocessing:** Preprocessing includes preparatory steps which modify and prepare feature vectors of the processed element set in such a way that the following algorithm steps can achieve higher performance in terms of both quality and speed. Preprocessing includes:
 - (a) **Weighting:** A copy of the original feature vectors is weighted using a TF/IDF (or alternatively BM25) weighting scheme in order to assign a higher weight to features with more discrimination power (see [Nanas et al. 2003] for a survey of term weighting methods).
 - (b) **Feature selection:** An information theory-based measure was used to remove features of a vector with low information content [Yang & Pedersen 1997].
 - (c) **Vector normalization:** Scale vectors to unit length by projecting them onto a hypersphere with radius one, so that only the direction of the, not its length, plays a role.
2. **Clustering:** Partition of the element set into groups of similar elements is performed by executing the following procedure (note that clustering is performed only if the number of data elements is larger than C_{max} , otherwise the next step is executed):
 - (a) **Seeding:** k-means++ seeding method [Arthur & Vassilvitskii 2007] is used to find the initial cluster seeds with the goal of improving the performance of the following step. The first seed is taken randomly from the data set and the others are selected to maximize the distance to already selected seeds. Number of seeds is chosen as $\frac{C_{min}+C_{max}}{2}$.
 - (b) **Partitional clustering:** Data elements are clustered using the spherical k-means clusterer (see Chapter 2.2.2.1) initialized with seeds generated in the previous step. A balancing strategy is employed to prevent extreme differences in clusters sizes, as these would lead to hierarchy degeneration: a large cluster is penalized when computing similarities between its centroid and data elements, if the difference between its size and the average cluster size becomes excessively large.
 - (c) **Cluster splitting and merging:** To find a more optimal number of clusters an extended x-means [Pelleg & Moore 2000] strategy

for estimation of number of clusters is used, which allows for both splitting and merging of clusters [Muhr & Granitzer 2009]. The resulting number of clusters must remain within the limits defined by C_{min} and C_{max} . The strategy involves merging of small clusters if the cohesion of the resulting cluster remains above a predefined threshold, and splitting of large clusters if the improvement in cluster cohesion is above a threshold, where the thresholds values are based on experience and estimated through experimenting.

- (d) **Refinement:** If clusters were splitted and merged then the resulting partition, with the corresponding cluster centroids, is used as the initial configuration for the k-means clusterer, which is applied to further refine the clusters.
3. **Cluster labeling:** Labels providing a short summary of a cluster are selected from the clusters centroids as feature (terms) with highest weights. Two types of labels can be computed:
- Discriminative labels are computed from centroids built from weighted vectors which, in the context of the parent-cluster, provide a description of the cluster differentiating it from its siblings.
 - Descriptive labels are computed from centroids built from the original, unweighted vectors, providing an "absolute" description of the cluster which is independent from cluster's context.
4. **Projection:** Computation of the data set layout and geometry is performed using a modified InfoSky method (see Chapter 3.3.5).
- (a) **Force-directed placement:** Cluster centroids (or vectors of data elements, if the clustering step was skipped) are layouted in the 2D space so that topically similar objects are placed close to each other and dissimilar ones are positioned further apart. This is performed using a plain force-directed placement (FDP) algorithm, with a force model defined in equation 3.3. An important difference compared to InfoSky is that there is an upper bound on number of elements, C_{max} , which is very small making FDP perform extremely fast. Also, applying FDP on small data sets reduces the chances of getting stuck in a local minimum.
- (b) **Inscribing:** The layout computed in previous step is normalized to occupy $[-1, 1]$ 2D space, and then inscribed into the convex polygonal area of the parent cluster using the procedure described in Chapter 3.3.5.

(c) **Voronoi subdivision:** Polygonal area of the parent cluster is subdivided into sub-polygons by assigning a polygonal area to each computed cluster using a Voronoi area subdivision [Okabe et al. 2000, Aurenhammer 1991], with points inscribed in the previous step used as control points. In contrast to InfoSky the areas are not related to the size of corresponding clusters (since the size is conveyed by height in the landscape, see next section).

5. **Recursion:** Apply the above listed steps recursively to the data elements of each computed cluster, unless the number of data elements is less or equal to C_{max} in which case the recursion stops.

Cosine coefficient (see equation 3.1) is used to compute the similarity between vectors in all stages of the algorithm.

4.2.1 Incremental Computation

To compute an initial configuration, the algorithm first processes the whole data set. When the data set changes, i.e. data elements are removed or modified, and new ones are added, the algorithm is re-run using the previous configuration as the initial state, in particular:

- New documents are inserted into the existing hierarchy propagating towards the bottom by choosing the most similar cluster at each level. Removed documents are eliminated from the hierarchy.
- Centroids at all levels of the hierarchy are updated, and those subtrees which experience a shift larger than a predefined threshold are reclustered using the recursive procedure described above. Other branches of the hierarchy are left unchanged.
- Layout of the whole hierarchy is updated in a top-down fashion, where FDP is initiated using positions computed previously in the step 4.(a). As FDP is inherently incremental when applied on a previously computed stable layout, the old positions will only be slightly altered, depending on the changes of the corresponding high-dimensional vectors.

Using the previous configuration as an initial state when computing a new, modified one is necessary because both k-means and FDP are very sensitive to initial conditions. Running the algorithm from the beginning using arbitrary initial positions would most likely yield a completely different result. Nevertheless, to guarantee that the incremental changes in the layout and geometry are smooth, the algorithm should be reapplied while the data set changes are still below a certain threshold (typically less than ten percent).

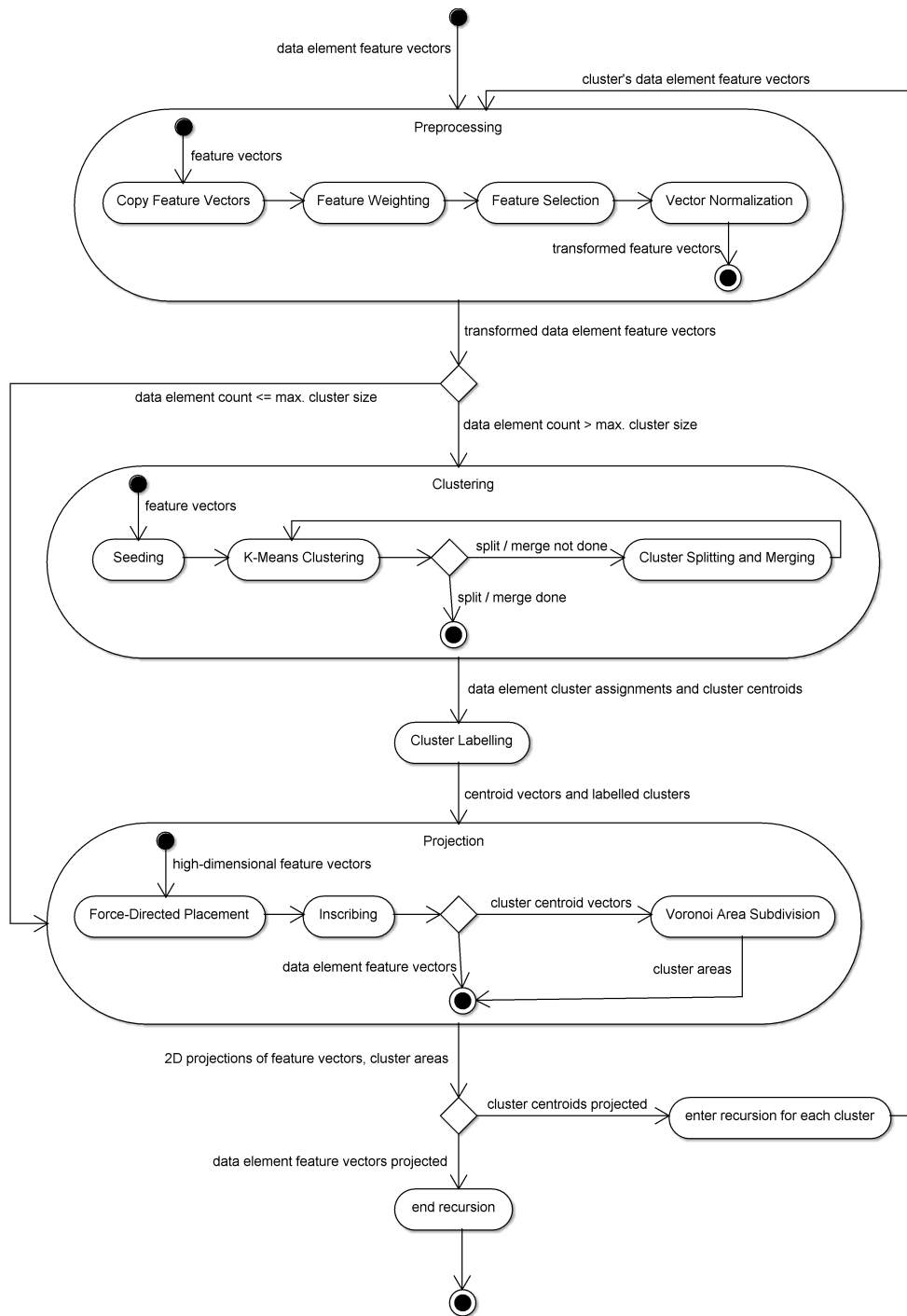


Figure 4.1: Ordination algorithm UML diagram.

4.2.2 Scalability

Given a collection of data set elements of size N , the time complexity of the k-means clusterer with splitting and merging strategy is $O(C_{max}N)$, provided there is a limit on the maximum number of performed iterations [Muhr & Granitzer 2009]. As C_{max} is constant, usually with the value of 10 or 12, the partitional clustering method can be considered to scale linearly with N (i.e. in $O(N)$ time). The same is true for preprocessing of N elements, while labeling of C_{max} clusters is performed in constant time ($O(1)$).

The hierarchy which is created by recursive application of the partitional clusterer is prevented from degenerating by the balancing strategy, which results in the depth of the created hierarchy having an upper limit of $O(\log(N))$. Balancing itself does not increase the complexity as it is implemented within the similarity computation between documents and centroids, with large clusters being penalized depending on their size. Therefore, time complexity of the whole hierarchical clustering procedure (including preprocessing and cluster labeling) is $O(N\log(N))$.

Plain FDP has a time complexity of $O(N^3)$. However, as FDP is always applied on collections having an upper limit on size equal to C_{max} , every FDP run can be considered to execute in constant time ($O(1)$). For the same reason inscribing and Voronoi area subdivision are also considered to execute in constant time. Given that for a non-degenerate hierarchy there are $O(N\log(N))$ such collections to be projected, each consisting of C_{max} or less data points which are projected in $O(1)$ time, the aggregate time required for projection is $O(N\log(N))$.

We conclude that the time complexity of the overall ordination and clustering algorithm is $O(N\log(N))$. This conclusion supported by measurements performed on a Core i7 860 2.8GHz CPU, with 8GB main memory, using Java 64-bit server JVM version 1.6.0_23. Figure 4.2 shows scaling behavior for ten document sets with sizes between 10000 and 100000 documents (in increments of 10000), execution times (y-axis) are in seconds (see Table 4.1 for exact values). The algorithm scales to millions of documents, for example the INEX09 [INEX 2009] collection consisting of 2,666,190 Wikipedia Documents was processed in about 2 hours on a 2.67GHz Xeon machine with 16GB of memory [Muhr et al. 2010].

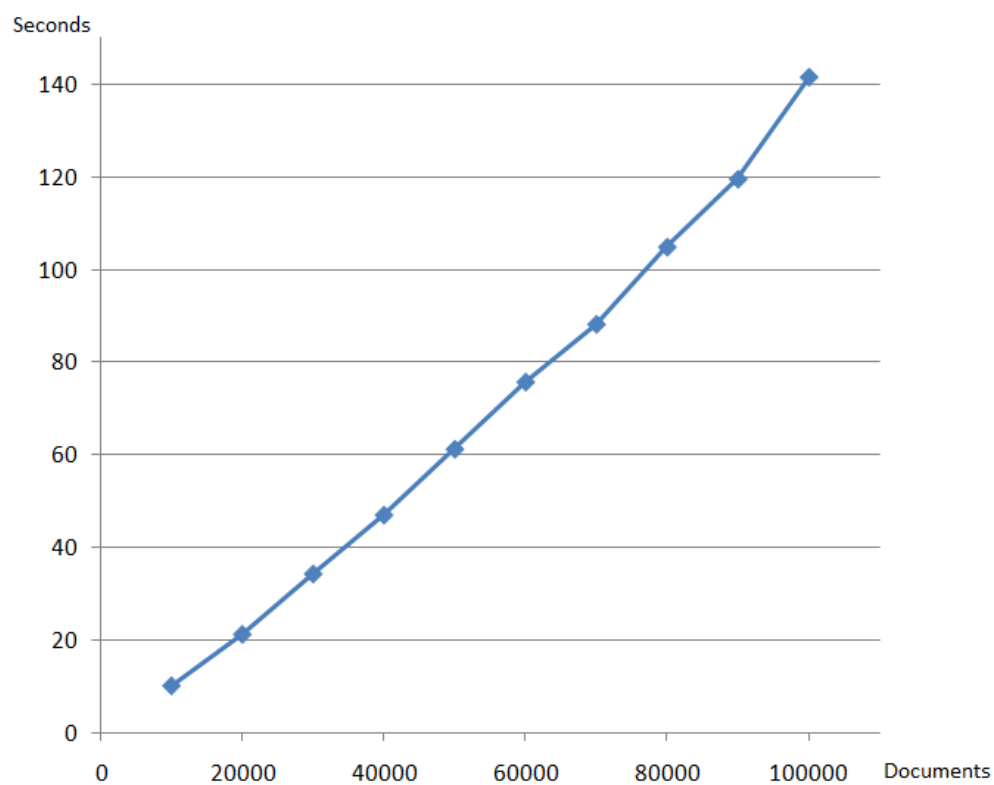


Figure 4.2: Execution times of the ordination and clustering algorithm, in seconds, for data set sizes from 10000 to 100000 documents.

Data set size	Time (seconds)
10000	10.16
20000	21.28
30000	34.32
40000	47.03
50000	61.23
60000	75.74
70000	88.22
80000	104.88
90000	119.55
100000	141.58

Table 4.1: Algorithm execution times, in seconds, for increasing data set sizes.

4.2.3 Visual Evaluation

Information loss is inherent when complex relationships from high-dimensional space are projected into a low-dimensional visualization space. Clearly, the capability of dimensionality reduction algorithms to preserve original relationships (i.e. distances or similarities) is very important. The goodness of fit is typically evaluated by computing a stress values (equation 4.1), which expresses the cumulative difference between the high-dimensional (D_{ij}) and low-dimensional (d_{ij}) distances for all pairs of data set elements.

$$S = \sum_{i \neq j} (D_{ij} - d_{ij})^2 \quad (4.1)$$

However, there are two reasons why such a globally computed stress value may not be ideal for assessing scalable ordination algorithms:

1. Scalable algorithms often attempt to reduce the amount of data element comparisons by introducing neighborhood-based optimization strategies. In such cases a good global stress value would not provide information on possible local stress peaks.
2. Distance proportions which exist in the high-dimensional space may not be ideal for visualization purposes in 2D space. Therefore, ordination algorithms need to produce 2D layouts which not only minimize stress, but also fulfill aesthetical und usability requirements.

Since the ordination algorithm introduced in this work generates the global layout by combining many locally computed layouts, local stress phenomena should be given a particular attention.

Equation 4.2 defines a local stress value between a pair of data set elements, where W_{ij} is the influence of the high-dimensional distance (i.e. to which extent high-dimensional neighbors contribute to the stress value) and w_{ij} is the influence of the low-dimensional distance (i.e. to which extent low-dimensional neighbors contribute to the stress value). D_{ij} and d_{ij} are assumed to be normalized (i.e. within the interval $[0, 1]$). The total stress for the i th data set element is given by S_i . Exponents a and b determine the size of the low-dimensional and high-dimensional neighborhood, respectively. The larger the values for a and b are, the stronger the influence of the neighborhood on the stress, yielding a local stress value which emphasizes the impact of the neighborhood. When both values are set to 0 then the standard stress value will be delivered.

$$S_{ij} = w_{ij}W_{ij}(D_{ij} - d_{ij})^2 \quad (4.2)$$

$$w_{ij} = (1 - d_{ij})^a, W_{ij} = (1 - D_{ij})^b \quad (4.3)$$

$$S_i = \sum_j S_{ij} \quad (4.4)$$

Using the StressMaps [Seifert et al. 2010a], which is a hybrid visual representation combining heat maps and information landscapes (see next Section for a detailed description), it is possible to visually identify areas with locally elevated stress values. In the following example the case should be examined when data set elements with large distances in the high-dimensional space are placed close to each other in the 2D space. For this purpose a was set to 20 and b to 0.

Figure 4.3 shows a standard information landscape consisting of 529 documents on the upper left, with hills appearing in areas with high document density. On the upper right the same landscape is shown, where color coding of data set elements is used to convey stress: blue symbolizes low stress, red symbolizes high stress, with tones of violet representing values in between. A StressMap, shown on the bottom left, reflects the stress values in three ways: data element color, landscape height and landscape texture color (heat map). In the StressMap hills will appear only in areas where stress value is high, while regions with low stress remain flat. A heat map texture is applied to the landscape geometry using a non-linear color palette representing stress values as color transitions from dark blue (low stress) over red to yellow (high stress).

The StressMap allows for effective identification of areas with high stress. Due to the fact that the original information landscape metaphor and data element positions are retained in the StressMap, visual stress assessment can be performed without any loss of context. Also, by zooming into high-stress area it is possible to identify responsible data set elements.

Two different localized stress phenomena can be identified by looking at the StressMap in the Figure 4.3:

1. Large clusters tend to have high local stress. By comparing high-dimensional cluster cohesion (i.e. the inverse average inner cluster distance) to its cohesion in the 2D space, it was found that the latter was significantly higher. The cause for elevated stress can be attributed to the Voronoi area subdivision algorithm, which does not consider high-dimensional cohesion when assigning the amount of area to a cluster. In this case the elevated stress is caused by distance scaling, which is not only acceptable but may even be desirable for visualization applications.

2. Elevated stress values are often found at the cluster boundaries. This phenomenon is a direct consequence of the neighborhood-based optimization strategies used in the ordination algorithm. The elevated stress near cluster boundaries is caused by the fact that force-directed placement algorithm only considers data elements from within one cluster, but does not consider those from the neighboring clusters. As a result, data elements which are positioned close to cluster boundaries may be placed close to a random element of another cluster which is distant in the high-dimensional space. This can be observed in Figure 4.3 on the bottom right: stress value is locally elevated for two documents at the boundary between the clusters *treaty, india, points* and *imperial, yen, corp*.

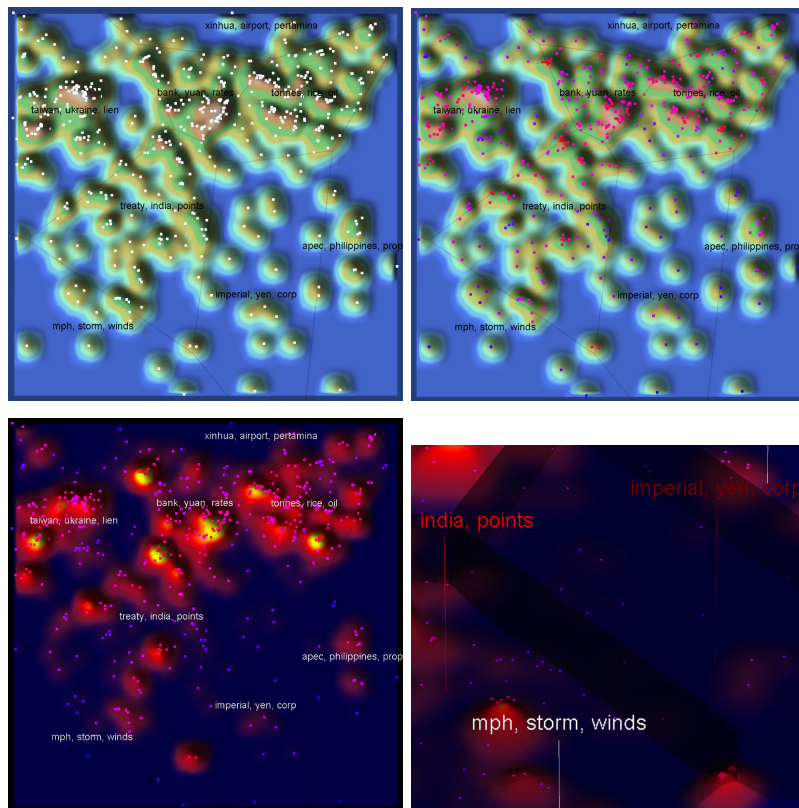


Figure 4.3: An example with 529 documents returned for query China: A standard landscape is shown on upper left, with data element stress color coding on upper right. On the bottom left is a StressMap for the same data set, with a magnification of local stress phenomenon at the border between two clusters shown on bottom right.

4.3 Landscape3D Visual Component

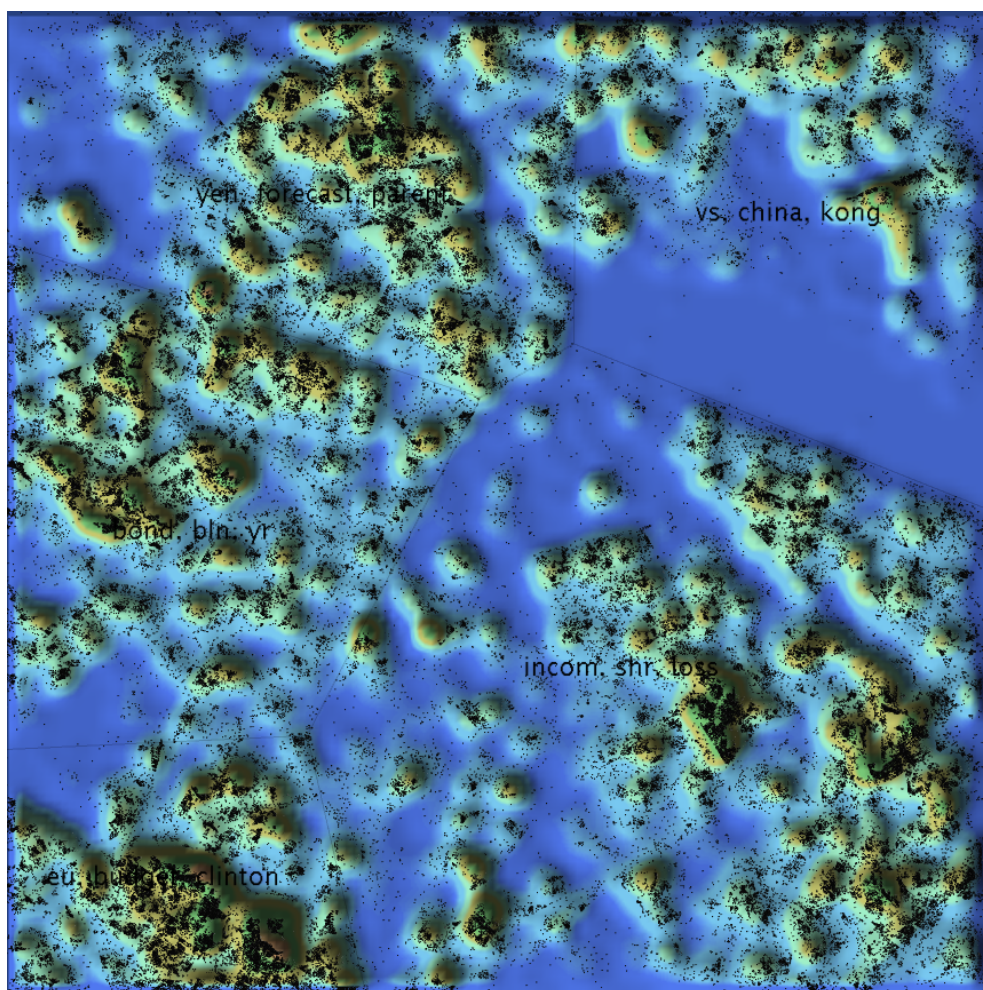


Figure 4.4: Landscape3D component showing 400000 documents.

Primary purpose of the Landscape3D component is to visualize complex relationships in large data sets, with as many as a million data items (and more) on a standard desktop machine. In 4.4 a landscape visualization of 400000 documents can be seen. Information landscape visualization 2.4.5 conveys relatedness, typically topical similarity, in the data set through spatial proximity in the visualization. Hills represent groups of related documents separated by sparsely populated areas represented as see. To facilitate orientation and navigation areas are labeled by highly relevant, descriptive terms from the underlying documents. Through hierarchical organization of the geometry Landscape3D can both provide an overview of the whole data set and

offer insight into relationships at finer levels of detail.

4.3.1 Interactivity and Navigation

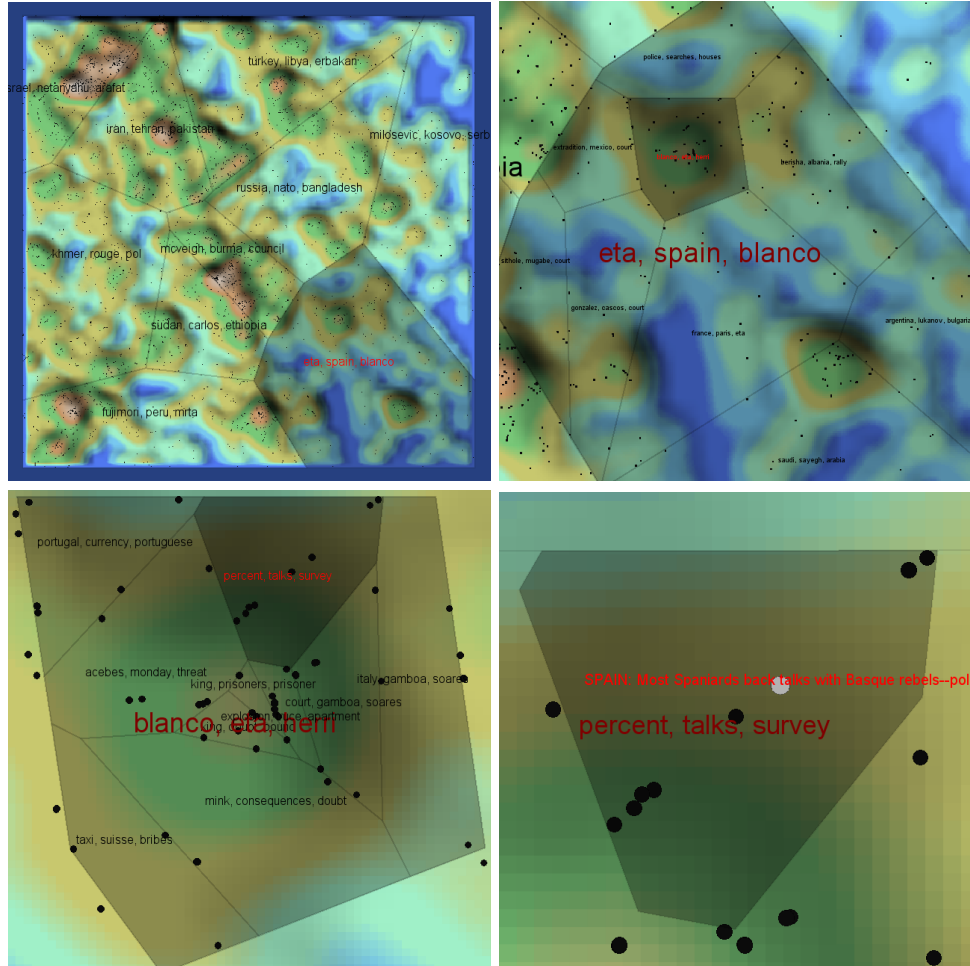


Figure 4.5: Zooming in the hierarchy of clusters. The visualization shows approximately 4000 search results on "terrorism" from the Reuters data set.

Landscape3D is fully interactive allowing zooming (mouse wheel), panning (dragging), rotating (alt+dragging) and tilting (shift+dragging) of the representation. Manipulations of visual document properties (color and icon) are supported through a context menu. Multiple document selection is possible using a lasso tool, while single selection is carried out by clicking on a document with the ctrl-key pressed. Clusters of related data set elements are represented as labeled areas which are organized hierarchically: zooming in on a cluster reveals the areas and labels of underlying sub-clusters. In this way adaptive

level of detail is provided which is adjusted to the zoom level. Labels play a central role for navigation: clicking on the label will smoothly zoom in on the selected cluster and reveal the underlying structure. Navigation guided by labels down the hierarchical structure is illustrated in Figure 4.5. The landscape also conveys the size and cohesion of the clusters. Higher hills emerge where the document count (density) is large, whereby the compactness of the cluster is an indicator of the strength of its topical cohesion: for example in the upper left screenshot in Figure 4.5 compare the clusters israel, netanyahu, arafat (upper left corner) and eta, spain, blanco (lower right corner).

4.3.2 Visual Property Coding

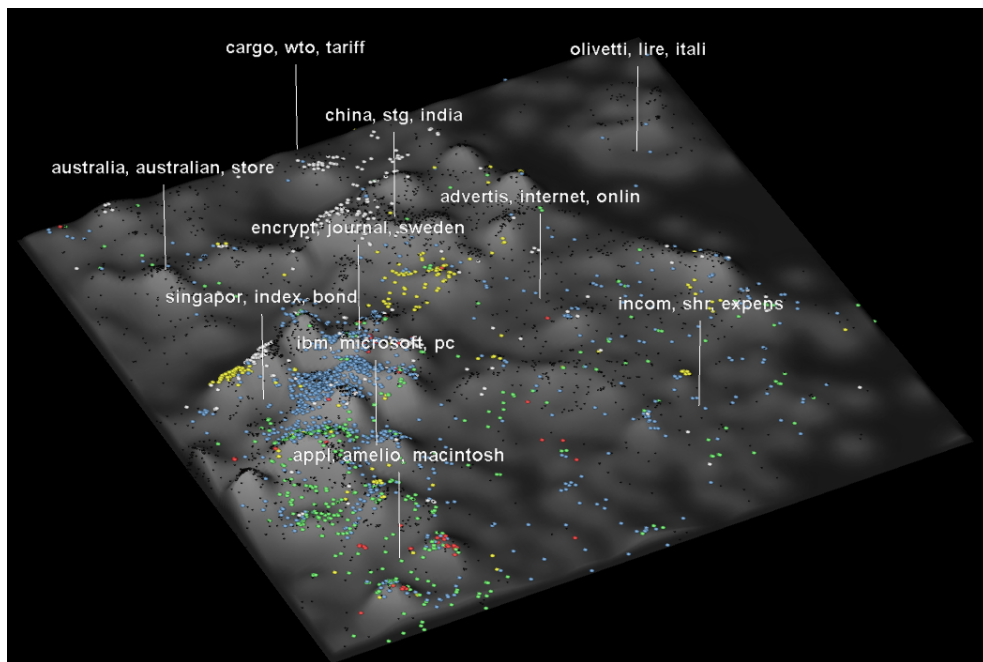


Figure 4.6: Mapping of features and metadata onto visual properties: in this case locations mentioned in the documents are mapped onto color. The query was "computer industry".

Another powerful feature is the mapping of document properties and metadata to visual properties of the corresponding visual items, such as color, icon or size. This features enables the user to discover correlations between topical clusters and selected features or metadata. For example in Figure 4.6 geographic entities mentioned in the document content are mapped to colors (note a monochromatic landscape texture for better recognition of item

colors). Visualized are about 6000 documents from 1996-97 on "computer industry", color assignments are following: New York - blue, California - green, Tokyo - yellow, Boston/Massachusetts - red, London - white. Relationships between topical clusters and selected metadata (locations in this case) can be recognized immediately: it is obvious that cluster "ibm, microsoft, pc" is connected with New York (blue) while "apple, amelio, macintosh" cluster has more to do with California (green).

4.3.3 Visualizing Change through Dynamic Topography

Large, real world data sets often have a pronounced dynamic behavior: data elements (documents) are added, removed and modified. The above described projection algorithm (Chapter 4.2) addresses this issue by the capability to incrementally incorporate changes in the data without disrupting a landscape representation the user has already familiarized with. Showing a sequence of such landscape allows the user to follow the changes through recognition of known, unchanged parts of the visualization [Sabot et al. 2009b].

An example can be seen in Figure 4.7. On the-top left is the first landscape containing 4382 documents on "computer industry" from 20.8.1996 to 19.5.1997. This landscape was augmented with documents on the same topic for the following three months. The second landscape, on top-right contains documents until 19.6.1997 (525 new document are shown in red). The third landscape, bottom-left, contains documents until 19.7.1997 (497 additional documents). The last landscape, on the bottom-right, contains documents until 19.8.1997 (for a total of 5932 documents). Notices how the global configuration remains recognizable but several areas have changed: for example the sea area in the center gradually disappears and the right cluster experiences a major change.

When the visualized data subset is modified (some documents are removed, new documents are added) the semantic map is not just filtered it is its topography that is altered. Old island and hills may disappear, change their shape or even new ones may arise from the seabed eventually remain as a permanent addition to the landscape. Other modifications of the topography, such as drifting of hills towards each other (correspond to merging of previously separate clusters) or splitting of an island (cluster breakup) may also occur.

Transitions of the landscape topography from an old to a new temporal configuration are incremental and adaptive in the sense that only those changes are introduced in the topography which are really necessary. The configurations of the parts of the topography, which are little or not at all affected by the modification of the data set, remain stable with respect to their previous

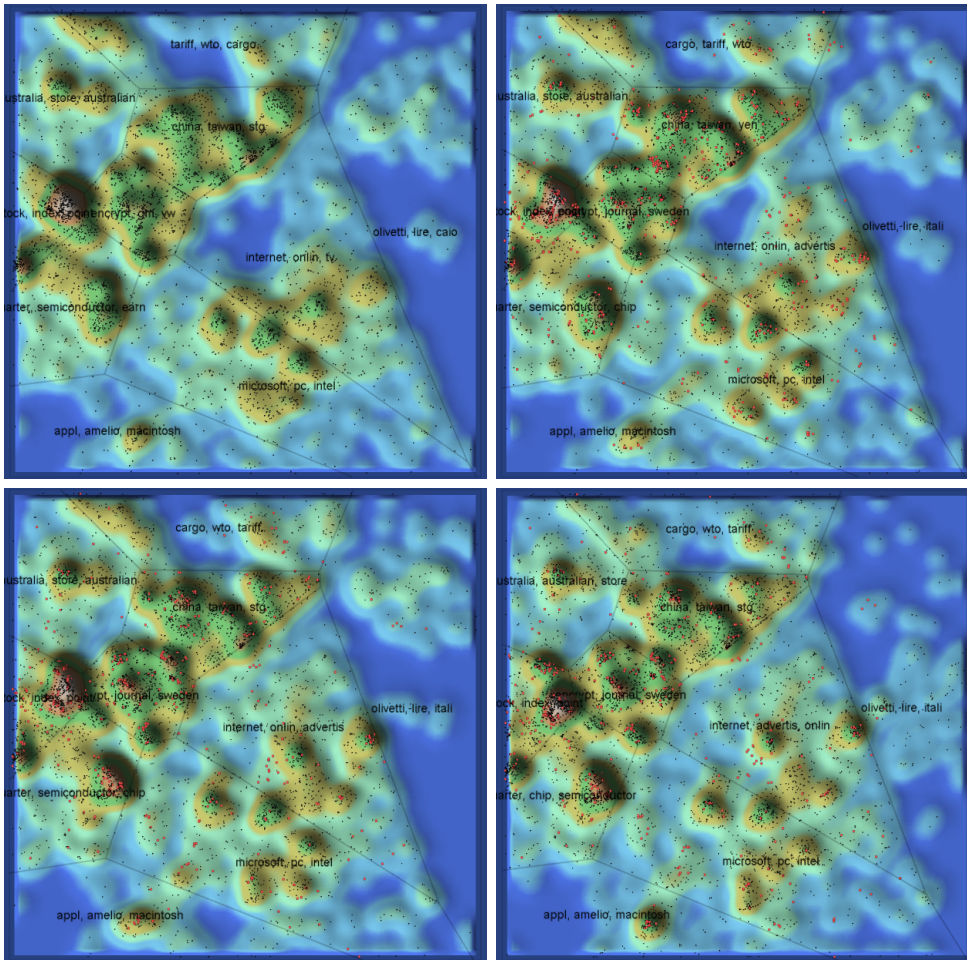


Figure 4.7: A sequence of growing, incrementally computed information landscapes. Documents added in each step are shown as red dots.

position and shape. In this way the user can understand the modified topography immediately through the recognition and orientation provided by the already known, preserved (or scarcely modified) elements of the topography. These adaptive, incremental transitions can be smoothly animated by morphing between the landscapes so that the user can follow and understand the changes.

4.4 StreamView Visual Component

StreamView is a scalable component for visualizing temporal development of several groups of data elements. Typically the groups are topically related

document clusters, but any other group of documents sharing a particular feature or metadatum, such as a particular person or organization, can be visualized. In Figure 4.8 ten topical clusters are visualized. Each cluster is assigned a color as can be seen in the legend in the bottom part of the image. Also shown in the legend is the size of each cluster and the number of elements within the selected time interval (see interval selection bar below). Above the legend there is a timeline defining the flow of time from left to right along the x-axis. Above the timeline the central part of the visualization can be seen, showing the temporal development of clusters along the time axis. The amount of documents belonging to a cluster within the corresponding time interval is represented by the thickness of the cluster, with the possibility of linear and logarithmic scaling. Interactivity includes mouse-over effects for displaying detailed cluster information, cluster selection by clicking on the corresponding cluster stream, and temporal selection of underlying documents using the interval selection bar shown at the top of the visualization. Three different alignment styles are available in the visualization: bottom-aligned, shown in Figure 4.8, centered, which closely resembles ThemeRiver metaphor (see Section 2.5.3) as shown in Figure 4.9, and top-aligned. Time granularity of the visualization can also be adjusted to show varying amount of temporal detail, for example StreamView in Figure 4.8) displays lower temporal resolution than the one shown in Figure 4.9.

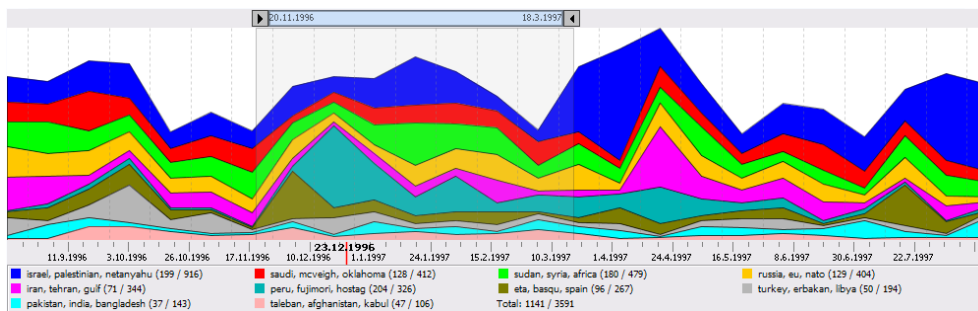


Figure 4.8: StreamView component showing temporal developments of topical clusters for news on "terrorism".

The example in Figure 4.8 shows temporal developments from August 1996 to August 1997 of topical clusters in for news returned in response to a "terrorism" query. For example it can be seen that israel, palestinian, netanyahu" and "eta, basque, spain" clusters both show continuous activity with several peaks, with the first cluster being overall significantly more intensive (i.e. more news were reported). On the other side "peru, fujimori, hostage" cluster has only one large, significant peak and then fades out to insignificance, leading

to the conclusion that it was an isolated event.

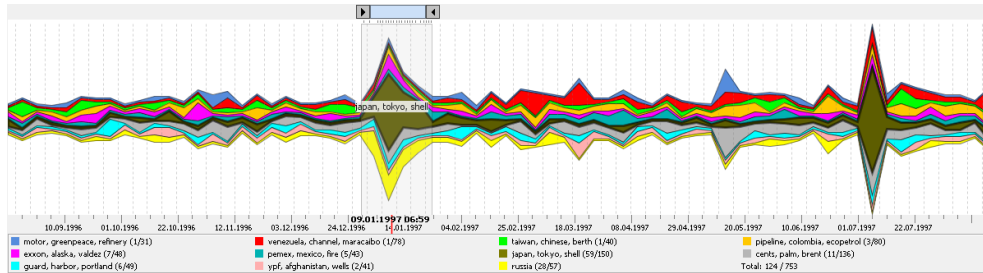


Figure 4.9: Temporal developments of topical clusters for news on "oil spill", extended by a document group containing the geographic feature "russia" (yellow stream on the bottom).

Figure 4.9 shows about 700 documents returned by the query "oil spill". This example illustrates the possibility to correlate the development of topical clusters with document groups sharing a particular metadata. Topical cluster "japan, tokyo, shell", shown in olive in the center, shows two pronounced peaks, while the metadata cluster "russia", shown in yellow at the bottom, correlates only with the first peak, but not with the second. This might be an indication that for the first event there is a connection between "japan, tokyo, shell" and "russia" in the context of "oil spill" (see Section 6.1 for a use case addressing this question).

4.5 Coordinated Multiple Views Framework

Each visualization component employs a specialized visual representation with properties and capabilities aiming at revealing relationships and patterns only for only one, or a small number of different aspects of the data (for example temporal developments, hierarchical relationships, or similarity). When the analysis of a data set necessitates considering more than just a single aspect of the data, user interfaces providing the capability for simultaneous analysis of manifold data aspects are required. One possible way to address this issue is to integrate various visualizations within a single immersive 3D virtual environment, such as in the Starlight System [Risch et al. 1998]. A more common approach is the Coordinated Multiple Views (see Section 2.7) paradigm, which addresses this challenge by combining several specialized visual components and "fusing" them together into a single, coherent user interface. Views are tightly coupled by the coordination framework so that changes caused by interactions performed in one component are immediately reflected in all

components within the user interface.

For this purpose a coordinated multiple views framework was developed. To provide for high-performance coordination even for large data sets, and to ensure coherency of visual properties and navigation without relying on external data management components, the framework is built upon a model-view-controller architecture. It includes following features:

- Shared coordinated data model including coordination of:
 - Visual properties: color, transparency, icon and size.
 - Logical properties: selection.
- Navigation coordination including:
 - Current location/focus.
 - Adjustment of the viewport and of the representation to make selected items visible on screen.
- A hierarchy provider delivers parent-child relationships needed for navigation along hierarchical structures.
- An event dispatch model and repaint rules which define the order in which coordination events are generated and dispatched, and specify when the repainting of components takes place (in order to avoid redundant repaints).
- Visual property providers ensure that different views employ a harmonized visual representation:
 - Icon provider: manages available icons and shapes.
 - Color provider: manages the color palette.
- Metadata provider delivers harmonized metadata representation for details on demand operation.

The coordination framework was used to build complex user interfaces consisting of more than ten coordinated components. The performance of the framework is sufficient to handle data sets consisting of more than a million coordinated data elements on a standard desktop PC.

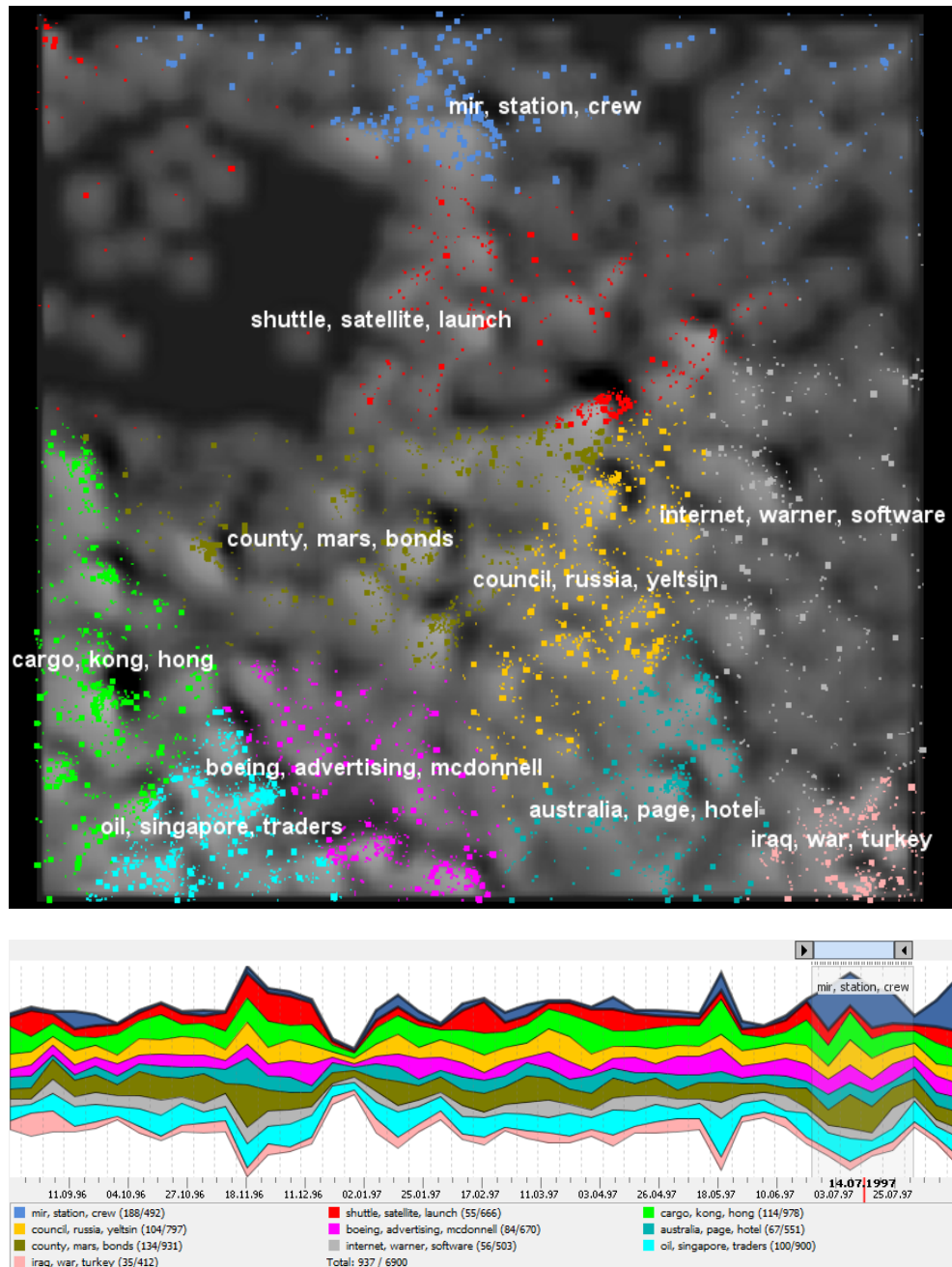


Figure 4.10: Coordination of colors and selection using a Landscape3D (top) and StreamView (bottom) components. Shown are 6900 documents for query "space".

Figure 4.10 shows an example of view coordination, where colors and selection are coordinated between a StreamView and a Landscape3D: Colors used to code clusters in the StreamView are also used to color the documents belonging to the corresponding clusters in the Landscape3D. Temporal selection of document in the StreamView (using the interval selection bar) is reflected in the Landscape3D where the selected documents are shown enlarged.

4.5.1 Additional Coordinated Components

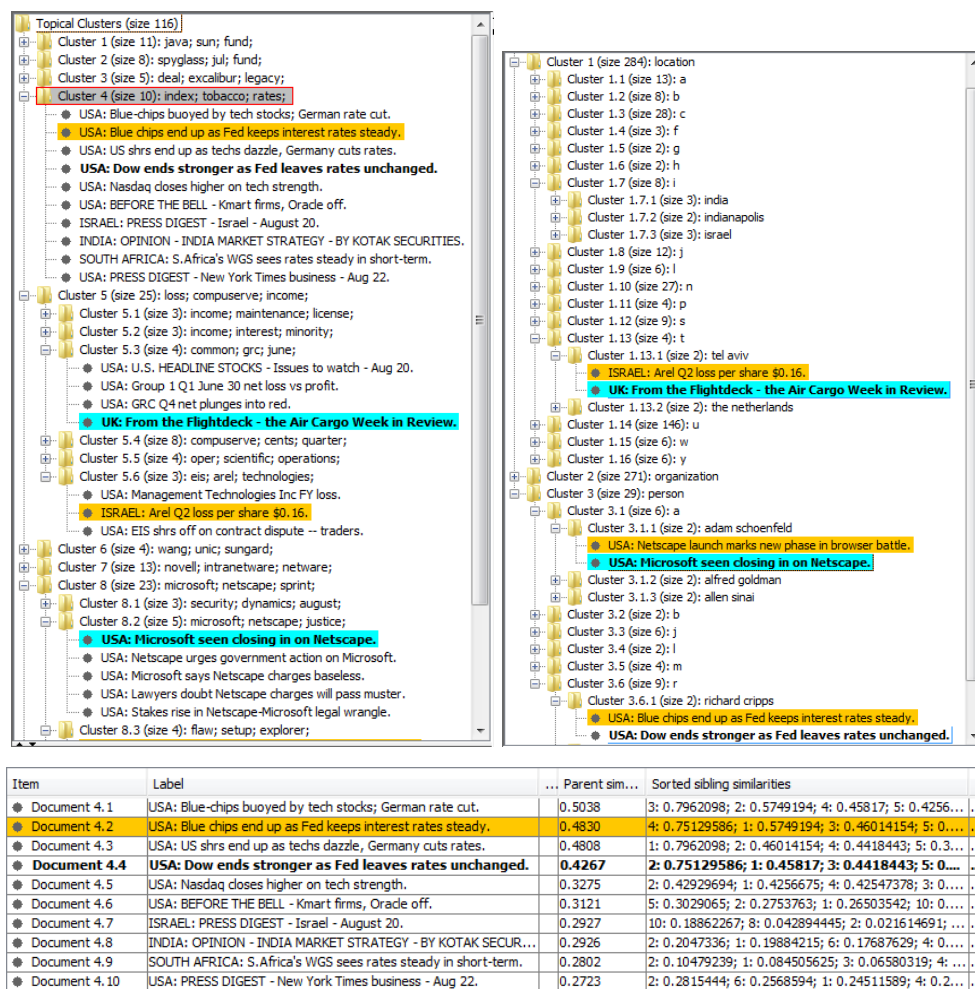


Figure 4.11: Trees showing a topical hierarchy (up-left) and faceted metadata hierarchy (up-right), and a table (bottom) showing document details.

Besides coordinated visualization components, such as Landscape3D and the StreamView, selected standard GUI widgets were extended to provide full

coordination capabilities. Figure 4.11 shows two tree components (up) and a table component (bellow), shown coordination of document selection (in bold) and color coding (orange and cyan). The left tree shown a hierarchy of topical clusters, while the right tree shows faceted metadata categories for locations, organizations and persons. The table shows detailed information for documents of the topical cluster 4 ("index, tobacco, rates"), including document's title, similarity to the parent and the similarities to its siblings.

4.6 Knowledge Discovery Visual Environment

Knowledge Discovery Visual Environment (KDVE) is a prototype user interface developed for testing and evaluating of new knowledge discovery methods and visualization components. It primarily relies on the algorithms and visual techniques described in this chapter to deliver means for visual topical-temporal analysis of large, dynamic textual data sets. It should be noted that KDVE also integrates additional algorithmic methods (see Section 5.1) supporting complex knowledge discovery workflows, as described in Chapter 6.

Application domains with data sets which are characterized by complex relationships across different aspects of the data may resist analysis unless different aspects of the data can be analyzed simultaneously. For example the StreamView representation gives a complete overview of temporal behavior of groups (clusters) of data elements, but it can not express topical relatedness between the clusters and the entities, which is the domain of the information landscape, nor it can convey hierarchical structures which are best shown as trees. Figure 4.12 shows the main KDVE visual analysis window which integrates following components into a single, "fused" [Sabol et al. 2007] visual analysis interface:

- One information landscape (on right) showing a topical relatedness within a hierarchy of topical clusters.
- One StreamView temporal visualization (bottom) showing temporal development of the topical clusters.
- Two TreeViews (on left): one showing topical clusters, the other one showing faceted metadata categories (hidden by the split pane shared by the topical cluster tree).
- Location bar (top) tracks user's location in the hierarchy during navigation, showing the topical cluster which the user is currently focusing on.

- A TableView (behind the StreamView, hidden by the TabbedPane) one showing details of the children of the currently focused topical cluster.
- A document content viewer (behind the information landscape, hidden by the TabbedPane) shows the content of a document (on mouse click).

All views are integrated by the coordination framework to provide simultaneous, "fused" topical, temporal and metadata analysis of the data set. The user can navigate or modify data element properties in any view, and the changes will be immediately reflected in all others. The coordination framework works behind the scenes to ensure that the state of visualized data set elements and the displayed subset are consistent over all views. In particular, the coordination of components includes the following:

- Navigation in the cluster hierarchy (can be triggered in any of the components) including the zoom factor, visibility in the viewport and current position in the hierarchy.
- Document selection (lasso-selection in the landscape, temporal selection in stream view, or cluster-wise selection in the trees).
- Document color (driven by the stream view color assignments).
- Document icons (user-assignable from any component).

Coordination of multiple views enables the discovery of patterns over the boundaries of individual visualizations, for example: Topical-temporal analysis can be performed by selecting documents belonging to two temporally separate events in the stream view, and then inspecting in the landscape whether those documents are topically related or not; Correlations between topical clusters and occurrences of a metadatum (e.g. persons) can be identified by assigning different icons to documents mentioning different persons, and then observing the distribution of these persons over topical clusters in the landscape. Detailed examples on how this interface is applied to achieve "fused", simultaneous topical, temporal and metadata analysis is given in the case study in Section 6.1. Results of usability studies performed using this interfaces are presented in Chapter 7.

4.7 Semantic Mediation Tool

The Semantic Mediation Tool (SMT) is a client-server system for semi-automatic, visually supported alignment of ontologies. It is being developed in a collaboration between the KnowCenter's Knowledge Relationship Discovery

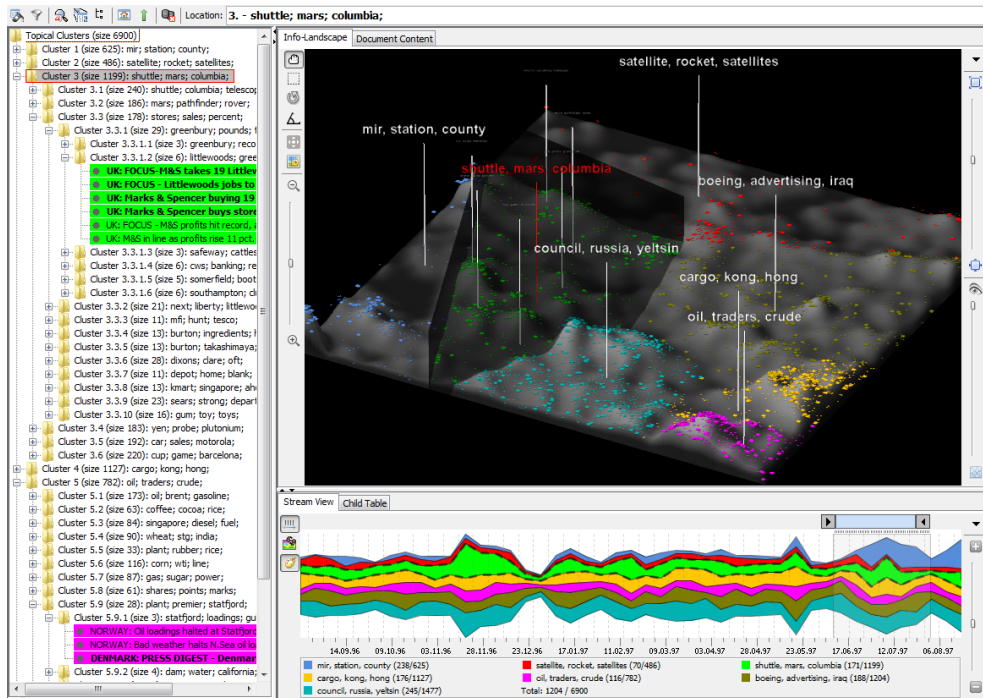


Figure 4.12: KDVE visual analysis window showing 6900 news documents on space. Document selection (by time: from June to August), document coloring (each topical cluster in different color) and navigation in the hierarchy (location: Cluster 3 shuttle, mars, columbia) are coordinated.

Areas and MIMOS [MIMOS 2011] Knowledge Technology Cluster. MIMOS is Malaysia's premier applied research center in information and communication technologies. MIMOS contributions to the project include an API for ontology loading and navigation, an alignment algorithm based purely on linguistic relationships, and an ontology visualization component. The rest of the system was developed using Know-Center technologies. My focus in the was on conception and implementation of the user interface which, besides the ontology visualization component, includes an information landscape, mediation table and view coordination technology. My other contributions include an ontology concept vectorizer, alignment algorithm based on clustering, and the client-server architecture of the system.

4.7.1 Ontology Alignment

Ontology alignment is defined as the process of bringing ontologies into mutual agreement through automatic discovery of mappings between related concepts

(see Figure 4.13). The ontologies themselves are unaffected by the alignment process. differences between ontologies with the aim of allowing their reuse. Ontology alignment is central for overcoming differences between ontologies with the goal of facilitating their reuse and interoperability. Collaboration and integration of systems using different ontologies becomes possible only when different ontologies are brought into mutual accord.

Automatic ontology alignment methods are attractive because manual creation of mappings between concepts from different ontologies is excessively time consuming for all but very small ontologies and therefore. A variety of approaches exist including methods based on string matching, linguistic methods, reasoning, machine learning techniques etc. A comprehensive overview of the field can be found in [Euzenat et al. 2004], with more recent developments introduced in [Gal & Shvaiko 2008].

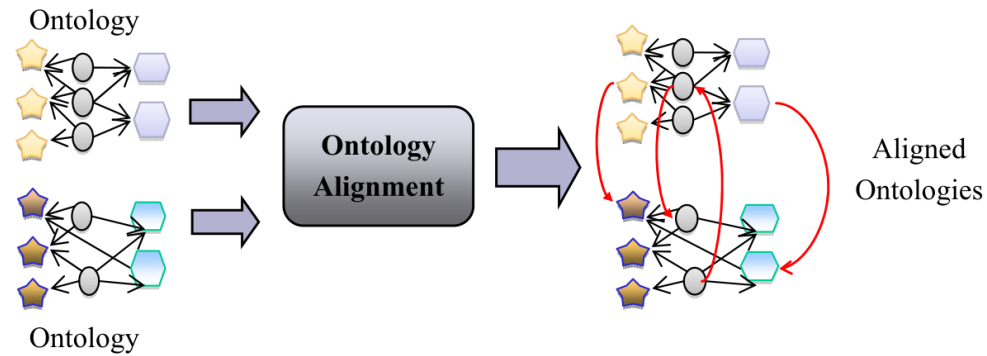


Figure 4.13: Ontology alignment.

Although fully automatic ontology alignment appear attractive as the solution of for interoperability of semantic systems, their results are rarely of sufficient quality. Challenges faced by fully automatic methods are manifold, including vocabulary differences, ontology modeling differences, different points of view on the modeled reality, etc. Semi-automatic approaches have been proposed to overcome those challenges by including human experts in the alignment process [Kotis & Lanzenberger 2008]. Obviously, design of the user interface is the crucial point for effectiveness of semi-automatic systems. In [Granitzer et al. 2010] we summarized a set of requirements for interactive ontology alignment tools from several independent studies. The requirements are:

1. Presentation of mapping candidates together with the estimated confidence and, if possible, with the inclusion of information on why the mapping was generated.

2. Navigation and exploration of ontologies providing detailed information on every element of the explored ontology.
3. Overview of the alignment results for identification of regions with promising matching candidates.
4. Capability to adjust the level of detail for the viewed data, as well as the choosing of the area of interest which shall be explored.
5. Filtering depending on features of the mappings, such as terms describing the concepts, mapping confidence, status of the mapping (confirmed, rejected, not inspected), etc.
6. Confirming and rejecting automatically generated mappings as well as adding and removing mappings manually. If possible, this should be done such that the system will learn from users interventions.
7. Collaboration via communication, commenting, tagging, and the voting on and annotating of mappings and ontology elements.
8. Ability to partition the mapping task into chunks assignable to team members and to monitor team member progress.
9. Saving and loading of users changes.

The SMT system was designed to address all of these requirements, however some of them, for example requirement 7, are not yet fully supported.

4.7.2 Visual, Semi-Automatic Ontology Alignment with SMT

Given a pair of ontologies in RDF format Semantic Mediation Tool (SMT) mediates between them by discovering mappings between pairs of concepts, where each concept is from a different ontology. SMT provides intuitive visual techniques for exploration of the computed mappings and navigation of the aligned knowledge bases. These empower the user to efficiently drill down to the area of interest and collect information required to accept or reject the automatically computed mappings.

Figure 4.14 shows the result of mediating two small medical ontologies developed at MIMOS: "Cardiovascular", shown in red, contains 414 concepts, "Occupational Health", shown in green, contains 331 concepts. On the very top of the window on the left side of the bar, a legend can be seen showing ontology names and their associated colors. A search box is shown on the right side of the bar. On the top-left there is an alignment table showing all mappings discovered by SMT algorithms. The mappings are sorted by their

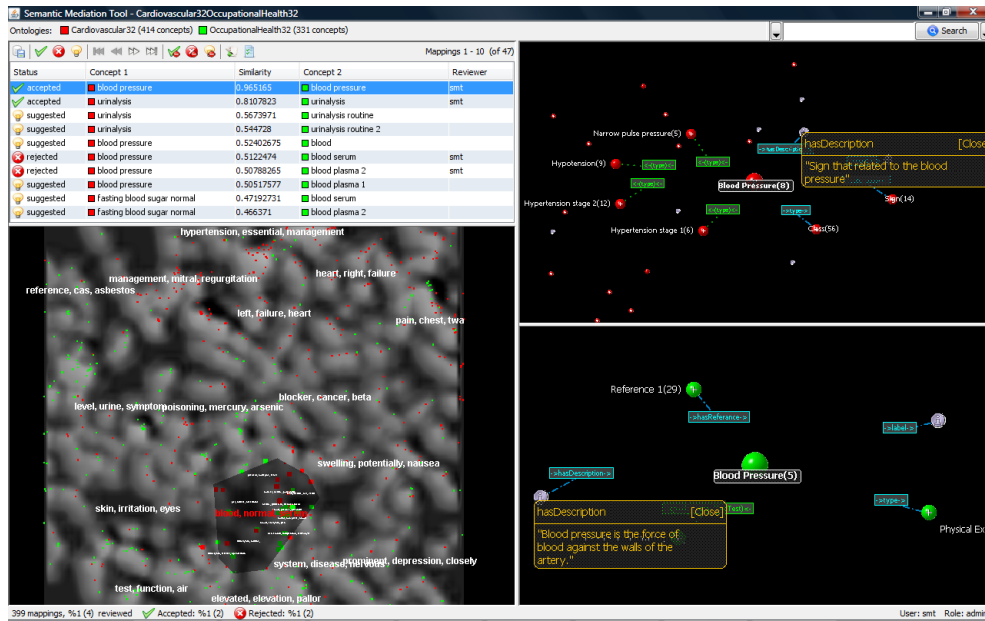


Figure 4.14: SMT Visualization Window.

estimated similarity (confidence). The table shows the name of the concepts color coded depending on the ontology, a similarity score between 0 and 1, and a status column showing users assessment of the mapping: accepted, rejected, or suggested (default value for all mappings that have not yet been reviewed). The user can review the suggested alignments, change their status to accepted or rejected, and save any changes performed on the mediation.

On the lower-left is the information landscape view which shows an overview of all concepts from the two ontologies as colored dots. Concepts are colored depending on which ontology they belong to. The distance between the dots signifies the dissimilarity between the concepts they represent. The closer the dots, the more similar the concepts are, so that areas with dots in different colors contain promising alignment candidates. The Landscape places the concepts from the two ontologies into hierarchically organized clusters, which are represented by nested areas. Each cluster is described by several descriptive keywords, which are useful when mediating between large ontologies, because the user can quickly drill deep into the hierarchy by following the keywords and in this way narrow down to his area of interest. Moving the mouse cursor over the keywords shows the area covered by the cluster. The user navigates deeper into the cluster hierarchy by clicking on clusters keywords, which reveals the keywords of its sub-clusters, and so on. Interactivity includes zooming, panning, and selection of a selected part of the

landscape. Selection in the landscape triggers a filtering action in the table, which is updated to show only mappings between concepts within the selected area. This is shown in Figure 4.14 where concepts in (and around) the cluster "blood, normal, serum" are selected (shown as enlarged rectangles). Full text searching and filtering functionality is also supported: after performing a search the hits will be highlighted in the landscape, while all other concepts and the corresponding mappings in the table will be filtered out.

Two instances of the Multimedia Semantic Browser (MMSB), which was implemented by MIMOS, are shown on the right hand side of the window. MMSB is a graph visualization providing detailed insight into the knowledge base structure. When a row is selected in the table, the triples belonging to the mapped concepts are displayed in the MMSBs to provide detailed information on the concept's neighborhood within its ontology. Concepts are shown as colored nodes where the color corresponds to the knowledge base. Literals are shown with a gray "information" icon, while predicates are displayed as links with the name of the predicate labeling the link. Concept neighborhood shown in MMSBs provides additional information on the concept pair supporting the user in assessing the suggested mapping.

Collaborative mediation is supported where an administrator selects subsets of the suggested mappings using the landscape to create tasks. Depending on the labels describing the selected region in the landscape, the administrator sends the task to an expert-user who has the appropriate knowledge to assess the mappings. The expert user can only view and modify the status of the mappings within the assigned task. When the expert assesses a mapping, the mapping is tagged with the expert's name. The administrator can view and override expert's assessments, and follow the progress of each individual task and of the whole mediation.

Detailed examples on how this interface is applied for semi-automatic, collaborative ontology alignment is given in the case study in Chapter 6.2.

4.7.3 Alignment Algorithm

SMT features a pluggable algorithm architecture capable of incorporating various ontology alignment algorithms. As of 2011 two algorithms have been implemented: the first algorithm is a scalable method based on an unsupervised machine learning approach, which also integrates linguistic information provided by WordNet [Wordnet 2011]; the second algorithm, developed by MIMOS, relies solely on linguistic relationships between two concepts. It uses WordNet lexical database to compute the similarity between a pair of terms, which is proportional to the inverse of the number of edges of the shortest

path connecting the two terms.

The machine learning based algorithm, developed by me, is based on methods implemented in the Know-Centers KnowMiner knowledge discovery framework (see Section 5.1). It operates in three stages:

1. **Concept Vectorization:** In this stage a multiple feature vector representation is computed for each concept. Every concept is described by the following information: concept label, concept description (if available), neighboring concept labels, and labels of relationship connecting to neighbors. Label information is extended with synonyms and, if desired, hypernyms and hyponyms, using the WordNet lexical database. Features obtained through WordNet are weighted to be weaker (configurable) than the original labels. In this way up to four different vector spaces are spawned, which are used to compute the similarities between any pair of concepts. Cosine similarity coefficient is used for each vector space, and a compound similarity value, computed as weighted average over all spaces, is used to compute the total similarity. Currently the weights for different vector spaces are fixed, with the importance of vector spaces declining in the order listed above. In the future version of the algorithm weights shall be automatically adjusted depending on the specifics of the ontologies to be aligned.
2. **Concept Clustering and Ordination:** Vectorized concepts from both ontologies are hierarchically clustered and projected using the scalable, hierarchical clustering and ordination algorithm introduced in /refordination-algorithm. Given that the total number of concepts from both ontologies is N , then the time and space complexity of this stage is $O(N * \log(N))$. The 2D similarity layout of concepts is used for visualization in the information landscape component. The balanced cluster hierarchy, created by recursively applying the modified k-means variant, groups together similar concepts even when they originate from different ontologies. The hierarchy is used by the next stage to efficiently find matching candidates.
3. **Match Finding:** To avoid comparing all concept pairs when finding mappings, which would lead to a quadratic matching time, mappings are found by inspecting only pairs of concepts assigned to the same cluster (or sub-cluster). By choosing sub-clusters deeper in the hierarchy in such a way that the number of concepts within the branch is always smaller than a fixed threshold V , with $V \ll N$, the number of comparisons performed for each concept is $O(V^2)$, i.e. its upper limit is constant.

This results in the last stage scaling linearly with N and performing quickly given a not excessively large value for V (for example 400).

Given that the first and last stage scale linearly, the total complexity of the algorithm is determined by the clustering step.

Chapter 5

Architecture and Implementation

This chapter provides an architectural overview of the software frameworks used to implement visual techniques described in Chapter 4. While only a fraction of the described software was directly implemented as a part of this work, a complete picture is given for readers to understand how everything fits together, how visual methods interact with the rest of the architecture, and which mining techniques they are built upon.

The implementation was mainly conducted by extending and applying two software packages, the KnowMiner knowledge discovery framework and the VisTools visualization toolkit, which were developed by the Know-Center's Knowledge Relationship Discovery Research Area [KC-KRD 2011] over the course of several years. KnowMiner [KnowMiner 2011] is a large, complex knowledge discovery framework providing a comprehensive set of techniques and algorithms targeting primarily text mining and analysis of large, high-dimensional data sets. VisTools is a visual analytics framework which complements KnowMiner with a set of coordinated visualization components. While I was involved in the conception of both KnowMiner and VisTools from their beginnings, I focused on and primarily contributed to the latter. The conception and architectural details of the KnowMiner framework, and the specifics of the wide variety of techniques and algorithms it implements, are not the contributions of this work. . However, functionality and architecture of both frameworks must be presented to understand how analytical workflows integrating visual and automatic techniques are realized (also see Chapter 6).

By applying KnowMiner and VisTools together and adequately combining human capabilities with automatic methods, one can build innovative applications and systems providing support for interactive, visually supported analyt-

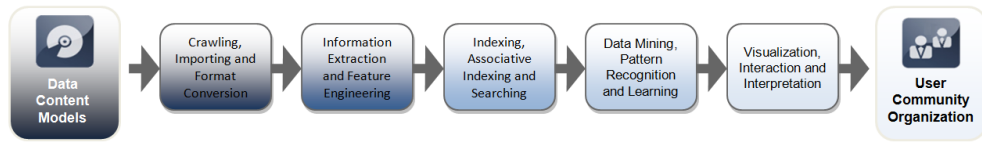


Figure 5.1: A processing chain of comprising of KnowMiner and VisTools functional blocks.

ical reasoning. Figure 5.1 shows a coarse grained view of processing pipeline comprising of KnowMiner and VisTools functional blocks, which closely resemble the well-known Knowledge Discovery Process (compare Figure 2.3). All examples given in this chapter use the Reuters Corpus, Volume 1 [RCV1 2000] as data source (unless stated differently) which includes over 800000 news articles from the 1990ties.

All software discussed in this chapter (including third party libraries) is implemented in the Java programming language and requires a Java Platform, Standard Edition, Version 6 6 [Java SE 6] compatible runtime to execute on. Visualizations are implemented as Swing components using Java2D for rendering. Some visual components can optionally make use of Java Binding for OpenGL [JOGL 1.1.1a], for hardware accelerated rendering.

5.1 KnowMiner

KnowMiner [Granitzer 2006] is a software framework offering a rich set of knowledge discovery algorithms and techniques. As discussed in Section 2.1.3, Knowledge discovery is the discipline dealing with analytical processes in which data is transformed and processed to identify relevant information and extract new, previously undiscovered, meaningful knowledge (see Section 2.1.3 for details). Thus, from the outside KnowMiner can be seen as a tool for bridging the divide between large, heterogeneous document repositories and users seeking to gain insight into knowledge hidden within very large piles of data. The primary focus of KnowMiner is on large textual data sets, where it provides a wide range of high-performance algorithms and techniques. However, due to its generic architecture it can be applied on other data types, such as semantic data (see Section 4.7.2) and multimedia repositories [Lux 2004].

The basic object KnowMiner operates on is usually a document, however, other type of objects, for example persons or abstract concepts, can also be the target of an analytical process. (Note that for the sake of simplicity we usually speak of documents.) During KnowMiner execution, documents propagate

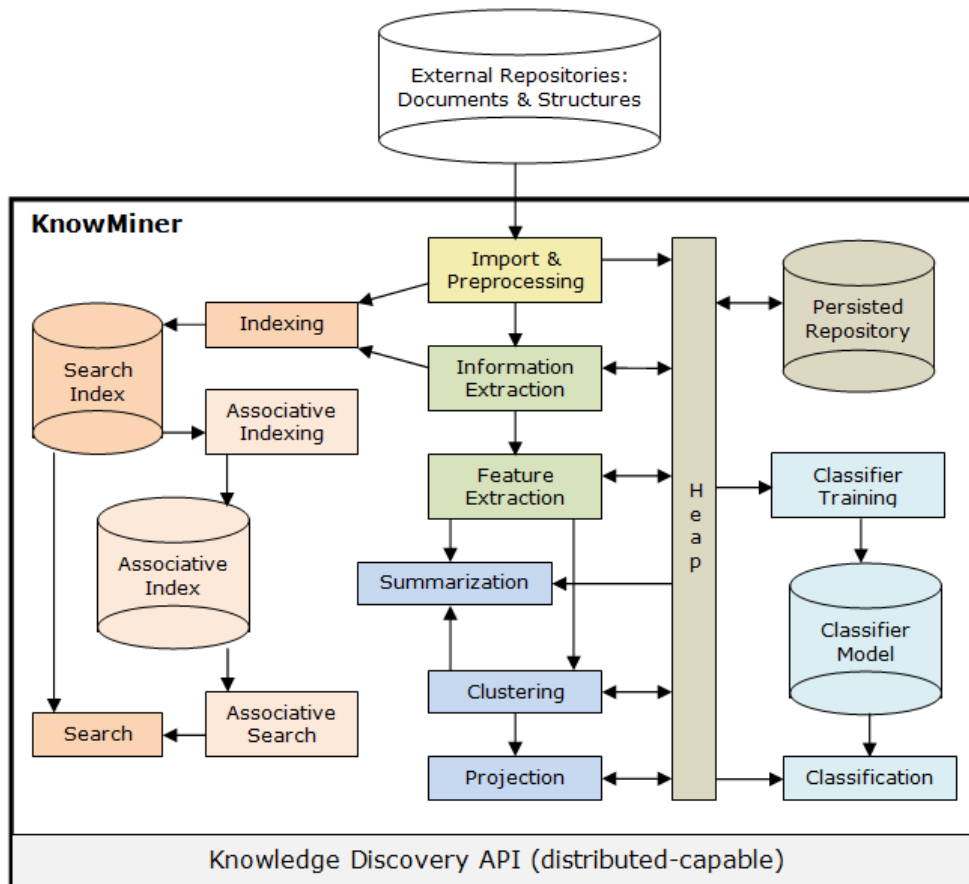


Figure 5.2: KnowMiner framework modular architecture. Color schema for modules: yellow - crawling and importing, green - semantic enrichment, red - information retrieval, blue - data mining, gray - data management.

through various functional blocks, where the algorithms analyze, manipulate and enrich them with new knowledge. In real-world applications the discovery process will, depending on the domain and application scenario, involve various combinations of available algorithms and techniques. Extensive configuration and parametrization capabilities provide means to tune the algorithms and the whole system to optimally perform on data from different application domains.

5.1.1 KnowMiner Modular Architecture

KnowMiner is a framework built around a modular software architecture [Klieber et al. 2006, Klieber et al. 2009a]. Due to its modular architecture and extensive configuration possibilities KnowMiner can be easily customized

and tuned to provide application specific solutions, realize complex knowledge discovery workflows and perform optimally on domain-specific data sets. In addition, the majority of algorithms and methods provided by the framework are accessible thorough a single, high-level, distributed-capable Knowledge Discovery API (KD-API). Knowledge Discovery API has proved powerful enough for industrial applications, which has been confirmed by successful applications of the framework in a variety of enterprise scenarios such as patent analysis, media tracking, governmental document repositories etc.

Discovery algorithms and analytical techniques offered by KnowMiner can roughly be grouped into following categories:

- **Crawling and importing** methods access remote repositories to collect data stored there. Gathered data is transformed into an a single internal format and imported into a local data back-end used by KnowMiner for further processing.
- **Information retrieval** methods provide comprehensive indexing and searching functionality in content and metadata, including statistical and semantic-based features.
- **Semantic enrichment** methods extract structured information from data, assign it a well-defined semantics and identify important features characterizing a data element. As textual information is of primary concern, information extraction and natural language processing methods are typically applied.
- **Data mining** methods include statistical methods, supported by semantic techniques, to identify patterns and new knowledge. Classification, clustering, dimensionality reduction and summarization are among the most commonly applied techniques.

To support humans in exploring and analyzing the data, identifying patterns in complex information and communicating the newly acquired knowledge, KnowMiner methods are complemented by visual analysis techniques based on information and knowledge visualization components from the VisTools framework (see Section 5.2).

The modular architecture of KnowMiner groups various algorithms into functional modules. Figure 5.2 shows KnowMiner’s high-level architecture with the main modules and the data flow between them. Detailed descriptions of the modules are as follows:

- **Import module** accesses and crawls external repositories and gathers the data. In a preprocessing step called format normalization (or for-

mat conversion), various formats (such as PDF, MS Word, MS Excel, HTML, rich text, plain text, ODF etc.) are transformed into an internal format, so that documents are accessible to other modules in a standardized way. Configurable metadata harmonization provides a possibility to map metadata from different sources onto a single (application-specific) internal schema.

- **Information extraction module** provides statistical and ruled-based methods for extracting new knowledge from text content. Information extraction (IE) [Kaiser & Miksch 2005] is a discipline dealing with extracting structured information from unstructured or weakly structured text documents using natural language processing methods. IE annotates text and assigns a well-defined semantics to the annotations, which can be used to as explicit metadata and features describing documents. IE typically includes the following techniques:
 1. Transformation and decomposition of text including tokenization, sentence extraction, stemming and part-of-speech tagging (i.e. recognizing nouns, noun phrases, verbs, adjectives etc.).
 2. Named entity recognition identifies entities such as persons, organizations, locations, time and numeric information (money amounts, scientific quantities) and others. Co-reference detection identifies various spellings or notations of a single named entity.
 3. Word sense disambiguation identifies the correct sense of a word depending on its context.

Full support for English and German is available out-of-the-box, while support for additional languages can be learned.

- **Feature extraction and vectorization module** generates statistical representation for documents from features extracted in the information extraction module. Multiple vectors per document can be constructed to capture different aspects of the document, for example: full content vector, title vector, extracted persons vector, user tags vector etc. Various feature weighting methods, such as TF/IDF or BM25, and feature selection schemes are available (see [Nanas et al. 2003] for an overview of various term weighting methods). The resulting feature vectors are used for comparing documents by numerous algorithms from other modules. Relatedness between any pair of documents is expressed as distance or similarity between the corresponding feature vectors, whereby several metrics types are supported, such as cosine, dice and Jaccard similarity coefficients, euclidean and city-block distance measures etc. (see [Cha 2007] for a survey on similarity and distance measures).

- **Information retrieval modules** provide comprehensive information retrieval [Baeza-Yates & Ribeiro-Neto 2011] functionality including:
 - **Indexer module** performs indexing of the document content, metadata and of results of other analysis methods, such as information extraction, classification etc. Supported data types include text, numerical and date/time, where by the last two can be supported with arbitrary precision. A special case is indexing of hierarchical and graph structures where the neighborhood of each node is also considered.
 - **Search module** executes search queries against the index to select relevant document sets. Provided are comprehensive document searching capabilities including: full text search, metadata search, range search, wildcard search, fuzzy search, Boolean queries, search by example (also with multiple examples and considering of structural information), concept search, relevance feedback, query expansion and suggestions, query spelling correction etc. Hits are returned as a ranked list sorted by relevance including metadata and content snippets. Faceted search capability provides filtering over aggregated metadata-fields (see Section 2.2.6), such as for examples over persons or geographic locations. Figure 5.3 shows an example of using faceted categories for search result filtering.
 - **Associative indexing and search modules** discover relationships between related concepts in the indexed data set based on co-occurrence analysis and various statistical and semantic criteria. Examples include discovery of relations between persons and organizations, between organizations and topical categories, or between any terms from the text content. Users can navigate along the computed associations by entering one or more search and then following the concepts returned by the system. An example is shown in Figure 5.4, where by entering "computer" the system returns "Microsoft, IBM, Intel,...". The user can choose any of these terms to continue exploring the associative concept graph. The concept graph in this particular case was constructed by associating concepts extracted from a 800000 documents. Associated concepts can also be used internally by algorithms, for example for search query expansion, semantic enrichment of documents, or mapping of knowledge structures onto each other.
- **Data mining modules** provide comprehensive include the following pattern recognition and knowledge extraction functionality:

- **Summarization module** computes a short summary for one document, a group of documents or any text fragment (see [Das & Martins 2007] for more information on summarization). Computed summaries are in the form of a list of weighted keywords, whereby descriptive and discriminative summaries can be generated. Descriptive summaries describe documents an isolated entity without considering the data set as a whole. Discriminative summaries describe each document in terms of what distinguishes it from other documents of a particular data set (or a subset thereof). Figure 5.5 shows cluster labels, computed from the content of the underlying documents.
- **Clustering module** makes use of on unsupervised learning techniques for computing groups of related documents [Berkhin 2002, Xu et al. 2005]. Typically clustering is performed on term vectors constructed from document content, resulting in documents being organized into a topical cluster hierarchy. As the above mentioned summarization techniques provide the possibility to generate a short summary for describing each cluster, the cluster hierarchy becomes useful for exploration of large data sets providing a virtual table of contents. Figure 5.5 (on left) shows such a cluster hierarchy for search results returned by a "computers" query: by following the labels one can quickly drill down to documents on the well-known Intel Pentium FDIV bug (see <http://www.intel.com/support/processors/pentium/sb/CS-013007.htm>). A variety of implemented clustering algorithms provides a clustering solution applicable to various domains and on heterogeneous data repositories. The choice of the algorithm and the algorithm parameterization can be performed manually or automatically, depending on the size and characteristics of the data set, and on the specification of the clustering tasks, the latter including: guessing the number of clusters (given constraints on minimum and maximum), hierarchical clustering producing several layers of sub-clusters (optionally including hierarchy balancing), performance vs. quality considerations etc. The incremental clustering capability is suitable for applications such as change monitoring in dynamic repositories, or for evolving of existing knowledge structures.
- **Classification module** includes several supervised learning approaches including various flavors of methods such as k nearest neighbors (KNN), support vector machines (SVM), Rocchio classifier etc. (see [Sebastiani 2002] for more information on classification methods). Classification tasks includes two different steps:

1. **Training:** During the training step assignments of documents to classes are learned using positive (and optionally negative) examples resulting in computation of a classification model.
2. **Classification:** During the classification step the classification model is used compute class assignments, including confidence values, for new, previously unseen documents.

Besides assigning documents to categories of a given taxonomy, classifiers are useful for tasks such as filtering of spam email (a typical binary classification problem) or sentiment detection for media analysis. Specialized features such as multi-label classification, incremental updating of classification models, or support for very large training sets is also provided by selected algorithm implementations. Figure 5.6 shows a trained classifier with twelve categories (on left), each representing a major IT company. Training examples per category include from 50 (Corel) to over 3300 (Microsoft) documents. Classified documents (on right) have confidence values assigned for different categories.

- **Projection module** performs dimensionality reduction [Fodor 2002] from the high-dimensional term vector space into a 2D visualization space, by preserving the high dimensional relationships as good as possible (see Section 2.3 for more information on ordination and dimensionality reduction methods). In the resulting 2D visualization space the spatial distance is a measure for relatedness between visualized items: related documents are placed close to each other while dissimilar ones are positioned further apart. The resulting similarity layouts can be utilized for visual exploration and analysis purposes, for example using the information landscape component (see Section 4.3). Available ordination algorithms include methods which are both scalable and incremental, thus supporting explorative visualization of large, dynamically changing data sets. Figure 5.5 (on right) shows a similarity layout computed by the ordination algorithm described in Section 4.2.
- **Persistence module** is a storage solution designed for efficiently storing and retrieving data involved in knowledge discovery processes, including externally provided data (document content and metadata, user feedback etc.) as well as knowledge generated by KnowMiner (extraction results, statistical information, cluster hierarchies etc.). Although the module does not provide knowledge discovery functionality in itself, it is optimized for common knowledge discovery scenarios and constitutes an important component of many knowledge discovery workflows.

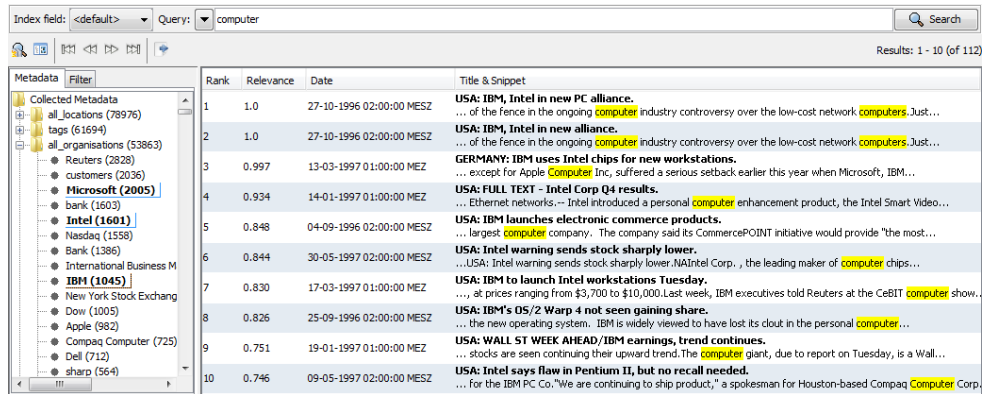


Figure 5.3: Faceted search example: Searching for "computer" returned over 20000 hits, with filtering possible using faceted categories (available are locations, organizations, persons, tags and sources). Selecting organizations "Microsoft", "Intel" and "IBM" reduces the hit count to merely over 100.

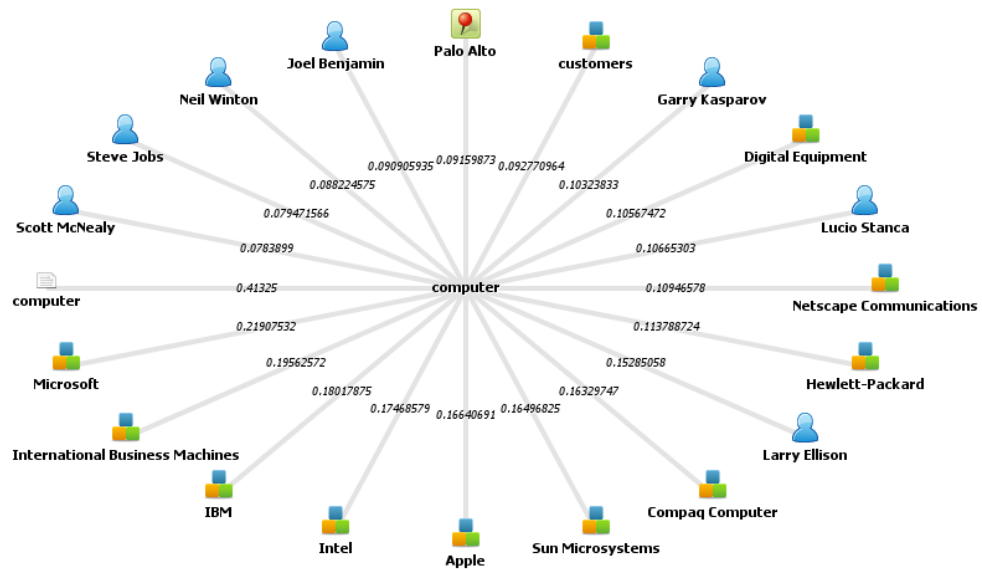


Figure 5.4: RadialView visualizing associated concept search: concepts returned for query "computer" (center) include companies, persons and locations (placed radially around the query term).

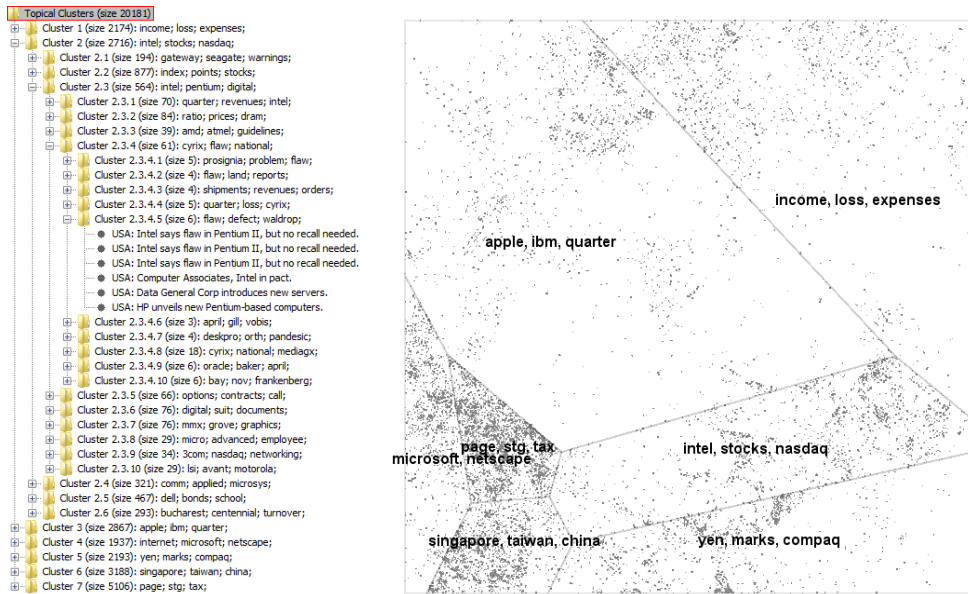


Figure 5.5: A hierarchy of topical clusters (on left) and a similarity layout (on right) for about 20000 documents returned by the query "computer".

classes	Classifier Overview		
	title	class	confidence
● sap	USA: Cisco, Intel, Microsoft back multimedia standards.	microsoft	1.0
● borland	USA: Cisco, Intel, Microsoft back multimedia standards.	apple	0.5
● corel	USA: Cisco, Intel, Microsoft back multimedia standards.	ibm	0.5
● apple	USA: Cisco, Intel, Microsoft back multimedia standards.	symantec	0.25
● ibm	USA: Computer show to highlight new network products.	microsoft	1.0
● netscape	USA: Computer show to highlight new network products.	sap	1.0
● autodesk	USA: Computer show to highlight new network products.	corel	0.5
● adobe	USA: Computer show to highlight new network products.	ibm	0.5
● symantec	SOUTH AFRICA: NEW SYSTEM COULD REVAMP CHAOTIC S.AFRICAN CUSTOMS.	microsoft	1.0
● oracle	SOUTH AFRICA: NEW SYSTEM COULD REVAMP CHAOTIC S.AFRICAN CUSTOMS.	novell	1.0
● novell	SOUTH AFRICA: NEW SYSTEM COULD REVAMP CHAOTIC S.AFRICAN CUSTOMS.	corel	0.5
● microsoft	SOUTH AFRICA: NEW SYSTEM COULD REVAMP CHAOTIC S.AFRICAN CUSTOMS.	ibm	0.5
	SOUTH AFRICA: NEW SYSTEM COULD REVAMP CHAOTIC S.AFRICAN CUSTOMS.	sap	0.5
	SOUTH AFRICA: NEW SYSTEM COULD REVAMP CHAOTIC S.AFRICAN CUSTOMS.	oracle	0.5
	SOUTH AFRICA: NEW SYSTEM COULD REVAMP CHAOTIC S.AFRICAN CUSTOMS.	apple	0.5
	USA: AirMedia releases personal portfolio software.	apple	1.0
	USA: AirMedia releases personal portfolio software.	sap	0.5
	USA: Spyglass licenses technology to Intercom.	novell	1.0

Figure 5.6: Classifying documents on "real-time software" into twelve classes representing major IT companies. Each document is assigned to more than one category, with the estimated confidence determining the winning class.

5.1.2 Knowledge Discovery API

Due to the complexity of typical knowledge discovery workflows and the involved data transformations, assembling applications directly using KnowMiner modules may cause a considerable implementation effort. This is especially the case when the number of used modules is large and the involved data flows are complex. KnowMiner offers standardized interfaces allowing one to easily assemble complex workflows based on a high-level knowledge discovery API (KD-API). KD-API provides access to the majority of KnowMiners algorithms and functions, and in most cases removes the necessity of transforming the data when it is passed from one module to another. KD-API design focuses on ease of use and hiding of complexity by internally mediating between modules, or when this is not automatically possible, returning meaningful error messages. Feedback from developers who are not experts in knowledge discovery was collected and considered during the development of several projects, which resulted in numerous refinement iterations of the API. KD-API provides the possibility to access KnowMiners functionality remotely over Java RMI. Besides a simple single-node operation a cluster-configuration is also supported. Distributed operation in clustering mode, which includes load balancing strategies, is fully supported for read-only workflows, while write operations must currently be performed on a single master instance.

5.2 VisTools

VisTools is a lightweight visual analytics framework based on the coordinated multiple views (CMV) paradigm. The framework currently includes several visualization components as well as extensions of standard GUI widgets (such as trees and tables), which can be combined using CMV to build visual user interfaces for exploration and analysis of large, dynamic, heterogeneous data sets. The framework also offers useful visualization algorithms and utilities, and provides access to a high-performance rendering engine developed by a group of specialists at the Know-Center.

5.2.1 VisTools Architecture

VisTools is built around a modular architecture shown in Figure 5.7. The four modules include:

- **Analytical component toolkit** (in blue) consists of several visualization components and extended standard GUI widgets. Majority of the

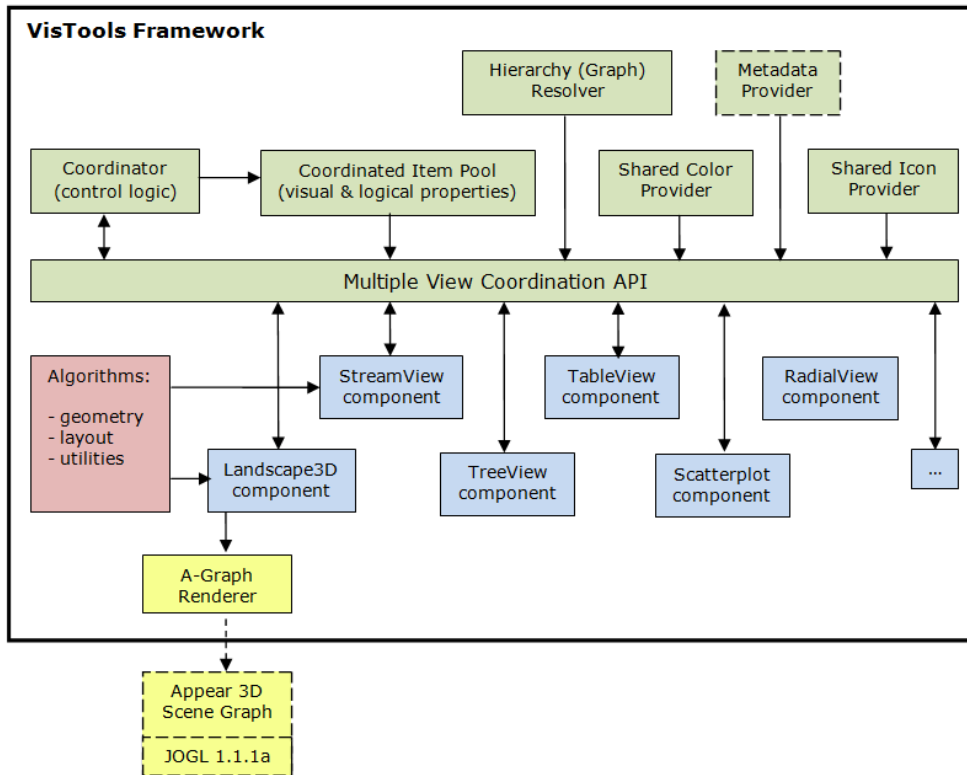


Figure 5.7: VisTools modular architecture. Color schema for modules: green - coordination of multiple views, blue - visual analysis components (incl. supporting data structures), red - algorithms, yellow - specialized renderers.

components are coordinated views, i.e. they maintain the same state of visualized data items and provide synchronized navigation behavior.

- **Coordination framework** (green) ensures that the state of visualized data and the navigation remain in sync in all views. Coordination frameworks maintains a shared data pool and implements logic for controlling the coordination process.
- **Algorithm module** (red) provides useful visualization algorithms, which are either used directly by visual components or are employed by KnowMiner ordination algorithms.
- **High-performance rendering (A-graph) engine** (yellow) provides a specialized, high-performance 2D and 3D rendering back-ends for selected visualization components (currently used by the information landscape component).

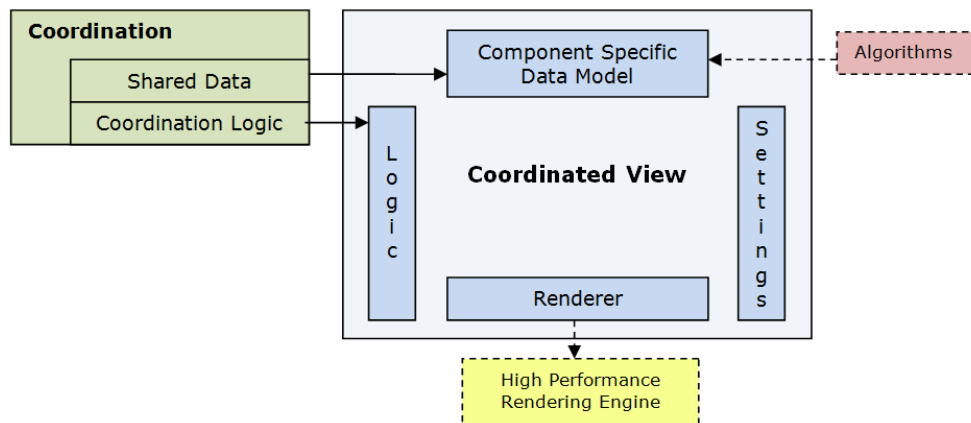


Figure 5.8: Coordinated components are based on the model-view-controller paradigm, extended with coordination-related data and logic.

VisTools architecture and implementation are entirely independent of KnowMiner and the framework can be used stand alone or in combination with other knowledge discovery software packages. However, it should be noted that all examples described in this work are based on algorithmic results provided by KnowMiner. A specialized software library, described in the Section 5.3, serves as a "glue" component between KnowMiner and Vistool providing commonly required data transformations and utilities.

5.2.2 Analytical Components

Central components of a visualization user interface are visual user interface components. VisTools offers several coordinated visual analysis components, including:

- Landscape3D - a 3D capable information landscape visualization (see Section 4.3).
- StreamView temporal visualization (see Section 4.4).
- Scatterplot visualization (see Section 2.4.1).
- RadialView visualization (see Figure 5.4).
- TreeView component (see Figure 4.11).
- TableView component (see Figure 4.11).

Components are configurable allowing the customization of colors, icons, fonts, strokes, interactivity and other properties. In this way they can be adapted to meet the requirements of different usage scenarios.

The basic architecture of each component can be seen in Figure 5.8. The components are implemented around the well-known model-view-controller design pattern, which is based on isolation between the data (model), painting and rendering (view) and the control logic (controller). Every component maintains its own control logic and data model, the latter including component-specific logical and visual information. In a coordinated multiple view architecture there will be an additional part of the data model, managed by the coordination system, which is shared by all coordinated component. Additionally, the control logic of each component must consider the coordination system and incorporate functionality for correctly handling different coordination requests. An important feature of the coordination architecture is that coordinated views may not directly change any coordinated data item properties and may not directly modify application state which is shared by several coordinated views. Instead all change requests are passed to the coordinator which notifies the views when model changes are complete and the views may repaint themselves to reflect these changes. See next Section (5.2.3) for details on coordination mechanisms.

Landscape3D component derives its name from the possibility to use 3D hardware accelerated rendering. However, this is not mandatory as rendering can also be performed by a Java2D-based rendering engine, which is limited to providing only a direct view from above (i.e. tilting and rotating are not supported). The 2D renderer is useful in avoiding rendering artifacts and errors in (rare) cases when 3D driver quality is not satisfactory. Note that because Landscape3D derives from Landscape, a 2D-based abstract super-class, these two names may be used interchangeably in the rest of this document.

In cases when scalability and performance requirements are particularly demanding, the 3D high-performance rendering engine should be employed. Using the 3D engine Landscape3D can handle up to millions of data items on a standard desktop machine (scalability subject to graphics card and memory constraints), whereas the 2D renderer will not provide smooth rendering when the data set size exceeds 100000. Landscape3D employs the A-graph rendering engine as its rendering back-end. A-Graph can switch between a Java2D-based renderer and the Appear 3D scene graph engine, the latter using Java Bindings for OpenGL [JOGL 1.1.1a] for hardware acceleration. A-Graph and Appear 3D libraries were developed at the Know-Center by colleagues specialising in high-performance rendering.

5.2.3 Coordinated Multiple View Framework

Coordination framework (see Section `refcmv-framework`) can be subdivided in three functional groups (for details on available coordination capabilities see Section 4.5):

- Coordinated data item pool maintains logical and visual properties of all data items subject to coordination. Coordinated data item properties include color, transparency, icon, size and selection of each coordinated data item, as well as the information on the currently focused data item (i.e. user's "location"). Important feature of the coordinated data item pool is that it provides read-only access for coordinated views. Only the coordinator may perform changes, subject to requests from coordinated views.
- Coordination protocols defined by the coordinator specify rules that coordinated views must obey in a coordinated environment. The purpose of these rules is to ensure that all coordinated views see the same consistent state of the coordinated data and that each view is notified about changes. Whenever a user interacts with a view and that interaction triggers a change which must be coordinated across other views of an user interface, the following coordination mechanism is triggered:
 1. Coordinated view invokes a coordinator method specifying which kind of change should be performed and which data items are involved.
 2. Depending on the request the coordinator modifies coordinated properties of the involved data items (such as color or selection) and/or changes the focus to a different data item (to perform navigation and ensure visibility of the data item).
 3. Coordinated view notifies all registered coordinated views about the change including the type (such as color or selection change, user focus change, etc.) and the data items involved. When a notification event arrives a view changes its internal model and state accordingly and repaints itself. This simple mechanism not only ensures consistency across different views, but it also enables a fast and memory efficient execution of coordinated actions which scales to millions of data items.
- Shared visual elements include icons and colors which must be displayed consistently over all coordinated views. Coordinated data items specify icons and colors by name. When a coordinated view needs to draw a data

item it must retrieve the icon and/or the color from the icon provider and color provider, respectively.

- Structure traversal is provided by a hierarchy resolver component. Methods such as clustering generate hierarchies and the hierarchy resolver provides a standardized way for all coordinated views to, given a data item (for example cluster), retrieve all its children and its parents(s). Note that currently only traversing of hierarchies is available, but the APIs would accommodate a graph based implementation too.
- Metadata provider is an optional component offering a standardized way for on-demand retrieving of metadata for one or more data items. A metadata provider can also inform a coordinated view which metadata is available.

5.2.4 Algorithms

The algorithms module provides several useful visualization algorithms which may be used directly by the visual component or are employed by KnowMiner as building blocks of ordination algorithms. These include:

- A force-directed placement (FDP) algorithm for computing similarity layouts of high-dimensional data along the lines of [Chalmers 1996] (see Section 2.3.4 for more information on FDP). This FDP implementation primarily used as a building block of the KnowMiner's scalable, incremental ordination algorithm described in Section 4.2. FDP can also be used directly in visual components for on-the-fly layouting of data sets containing up to several hundreds items.
- A Voronoi area subdivision method [Aurenhammer 1991] is employed in the same ordination algorithm as the above mentioned FDP. Voronoi subdivision can also be used directly in visualization components, such as for example in the tag cloud shown in Figure 2.6.1.
- A stream shape generator takes documents' cluster membership and their time-stamps (usually creation date) as input, and computes the shapes of cluster stream for the StreamView visualization component. It is a simple and fast algorithm which subdivides the time interval covered by the document set into subintervals, and for each subinterval computes the width of the cluster's stream to reflect the number of documents in that subinterval. The fact that the routine executes in sub-second time for millions of documents makes it suitable for on-the-fly computation of the StreamView geometry.

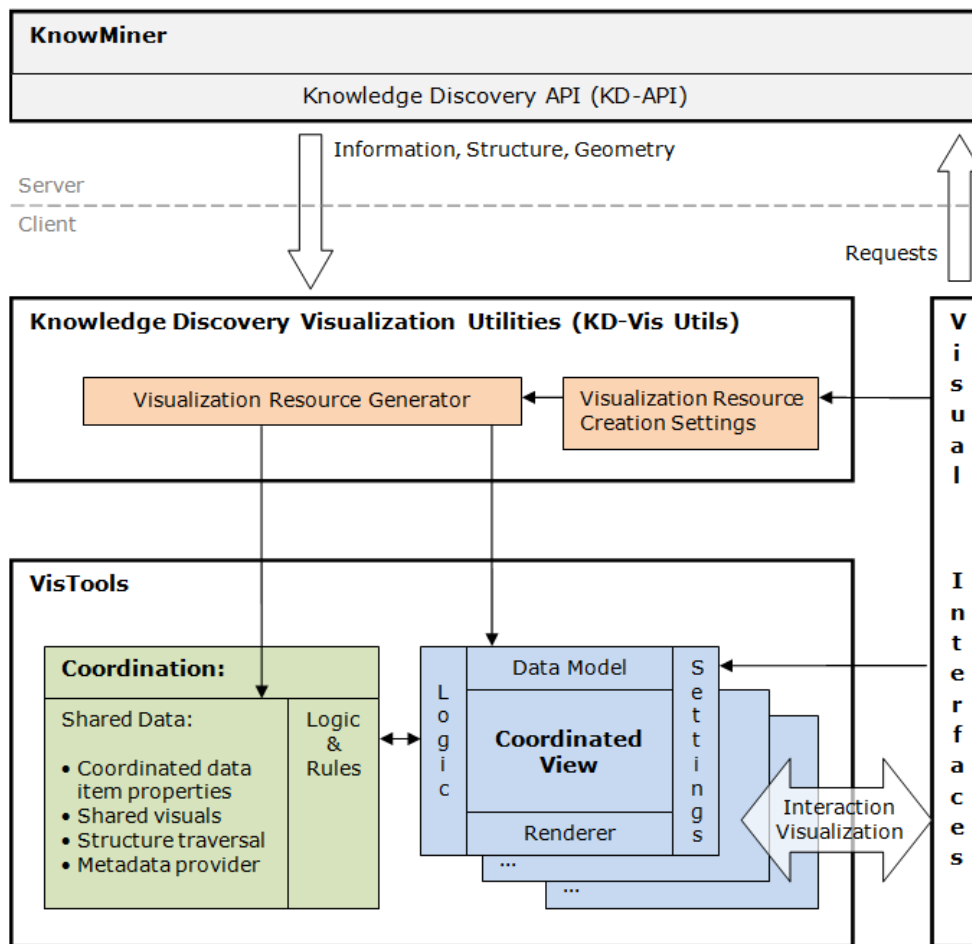


Figure 5.9: Integrating KnowMiner and VisTools to create visual analytics applications.

5.3 Integration of VisTools and KnowMiner

Each VisTools component implements a specialized data model suitable for the visual representation it provides. The coordination module maintains its own data structures which fit the needs of view coordination. On the other side, KnowMiner defines its own data structures in the knowledge discovery API (KD-API) which are suitable for returning results of various mining or retrieval operations, such as clustering, ordination, associative search etc.

To enable interoperability between KnowMiner and VisTools and facilitate creation of analytical visual applications which integrate automatic and visual methods, a data transformation library was developed, the Knowledge Discovery Visualization Utils (KD-Vis-Utils). As shown in Figure 5.9 the library acts

as a "glue" layer between the two frameworks and automatically creates visual component-specific data models from the KD-API results. It also creates two central data structures for coordination: the coordinated data item pool and the hierarchy resolver. Due to remote and clustering capabilities of the KD-API, visual client-server applications talking to distributed KnowMiner instances can be created easily.

Data structures employed across VisTools and KD-API make extensive use of arrays and Java primitive data types (as apposed to object collections and objects). Also, in cases when lists and maps are required and usage of primitive types appears adequate, TROVE library providing primitive collections and maps was employed [TROVE v2.1.0]. These measures led to significant memory footprint reductions and noticeable performance improvements, especially on the client side.

5.3.1 Prototype Applications

Two prototype visualization systems based on the described server-client architecture were implemented using VisTools, KnowMiner and accompanying libraries: the Knowledge Discovery Visual Environment (KDVE, see Section 4.6) and the Semantic Mediation Tool (SMT, see Section 4.7).

KDVE is a demonstrator for VisTools and KnowMiner functionality built on the architecture shown in Figure 5.9. It offers four distinct functional groups:

1. Import pipeline includes document import and format conversion, information extraction and document vectorization, text and metadata indexing, associative indexing, and persisting of imported and extracted information. Once the importing process is completed for a documents set, various analytical and retrieval techniques can be applied on them.
2. Information retrieval functions offer extensive document search functionality and provides visual navigation in concept networks using the associative search.
3. Visual analysis window provides several coordinated views, including a Landscape3D and a StreamView, for topical-temporal analysis of a document set (see Figure 4.12).
4. Classification functions offer the possibility to define and train new classes and classify previously unseen documents. and

These function groups are typically used in combination with each other to realize various knowledge discovery workflows. Selected workflows, focusing

on the the third function group - the visual analysis, are demonstrated in the case study discussed in the next Chapter (6).

Semantic Mediation Tool (SMT), a visually supported ontology alignment application, uses a smaller part of available techniques than KDVE, but applies them on semantic information instead of text. SMT's client-server architecture uses components shown in Figure 5.9 but also introduces new application-specific components, some of which were developed by MIMOS, which demonstrates the flexibility and extendibility of VisTools and KnowMiner architectures. As the KD-API functionality goes far beyond of what is needed by SMT, and at the same time would have to be extended with ontology alignment methods, SMT introduces its own, simple RMI-based client-server interfaces. The resulting application consist of the following main components:

- Server:
 - Ontology access component accesses the knowledge bases and delivers concept information to the alignment algorithms.
 - WordNet-based ontology alignment algorithm (MIMOS).
 - Clustering-based ontology alignment algorithm, with integrated similarity layout computation of ontological concepts used for visualization. This component also performs the indexing of the ontological concepts which is used for client-side filtering (KnowMiner, Know-Center).
- Client:
 - Table of concept mappings suggested by the alignment algorithms.
 - A landscape visualization providing an overview of all ontological concepts involved in the alignment process (VisTools, Know-Center).
 - Two graph visualizations for ontology navigation and exploration.
 - Coordination framework.

For an example on how SMT is used to align concepts from a pair of ontologies see Section 6.2.

It should be noted that besides the two prototype applications, a productive system was realized within a 5-year applied research project, with me being the project leader at the Know-Center. The system was built by integrating KnowMiner and VisTools technologies in the m2n Intelligence Management Framework [m2n IMF 2011]. The architecture of the resulting system

is beyond the scope of this work, however, integration of VisTools visual techniques and KnowMiner algorithms into the productive system was achieved using the same building blocks and technologies as described here.

Chapter 6

Case Study

This Chapter demonstrates the use of prototype visual applications on real world data. Applications of visual analysis methods are described application in the context of textual and semantic data, with examples outlining the relationships between visualization and machine processing. The first example demonstrates how the Knowledge Discovery Visual Environment (KDVE) (see Section 4.6) can be applied to perform a topical-temporal visual analysis on a set of documents. In the second example the Semantic Mediation Tool (SMT) (see Section 4.7) is used for semi-automatic matching of concepts from a pair of ontologies using visual techniques. To follow the presented workflows it is assumed that the reader is familiar with the visual and algorithmic components introduced in Chapters 4 and 5.

6.1 Fused Topical-Temporal Analysis

In the following example visual analysis methods from the Knowledge Discovery Visual Environment (KDVE) are applied to analyze a document set gain insight into topical and temporal aspects of the data. Demonstrated workflow makes primarily use of visual techniques which are backed by the results of automatic analysis. The Reuters Corpus, Volume 1, in English language [RCV1 2000], which is a collection of news articles from the 1990ties is used as data set. The corpus is frequently used in the text mining community for testing, evaluation and demonstration purposes.

In the provided example the user is seeking to gain knowledge on oil spills (which unfortunately are recurring incidents nowadays, as they were in the 1990ties). After selecting relevant documents using search our hypothetical user applies visual methods to identify a particular area of interest and recognize potential major events. Subsequently, the user focuses on discovering

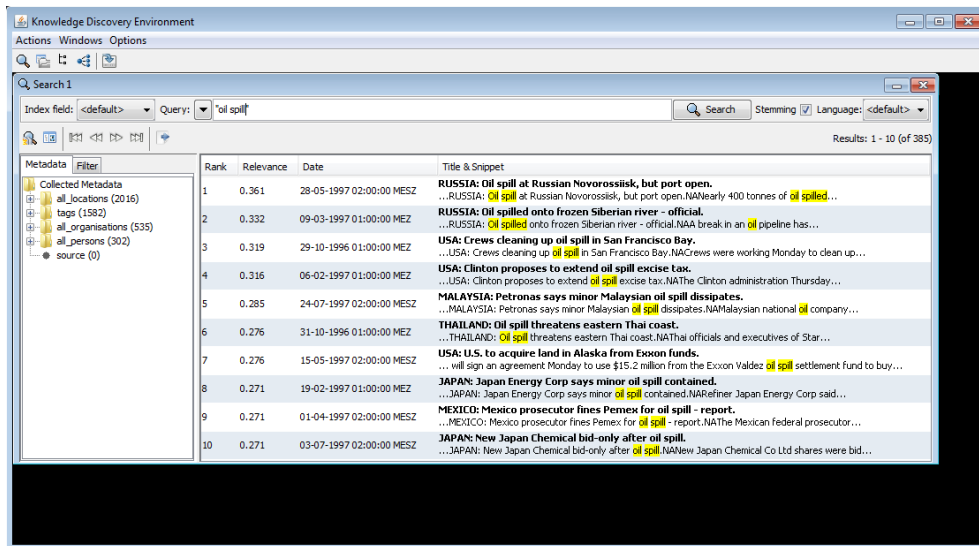


Figure 6.1: Search for "oil spill" returns 385 news articles, which is too many for manual analysis of the whole data set.

whether different events, which occurred in the area of interest, are independent and isolated from each other or related and causally connected. Using explorative analysis techniques user postulates a hypothesis and then attempts to validate it by applying different analytical strategies. The process of recognizing patterns, generating a hypothesis and then validating it through on-demand analysis, generates new insights and facts, even when the hypothesis is rejected (also see Section 2.1.3). As we will see, important for the success of this analytical process is a suitable combination between the interactive visual methods and the result of automated analysis.

6.1.1 Step 1: Getting an Overview

To select news articles on oil spill the user opens a search window, types "oil spill" in the search query field and executes the query (Figure 6.1). Information retrieval tools excel at finding a single or several relevant pieces of relevant information, but when a when a holistic view on a data set is needed, techniques providing an aggregated overview of the whole data set are required. For all documents returned by the query Figure 6.2 shows:

- A similarity layout of documents, computed by an ordination method (see Section 4.2), is shown in the Landscape visualization component (see Section landscape-component).

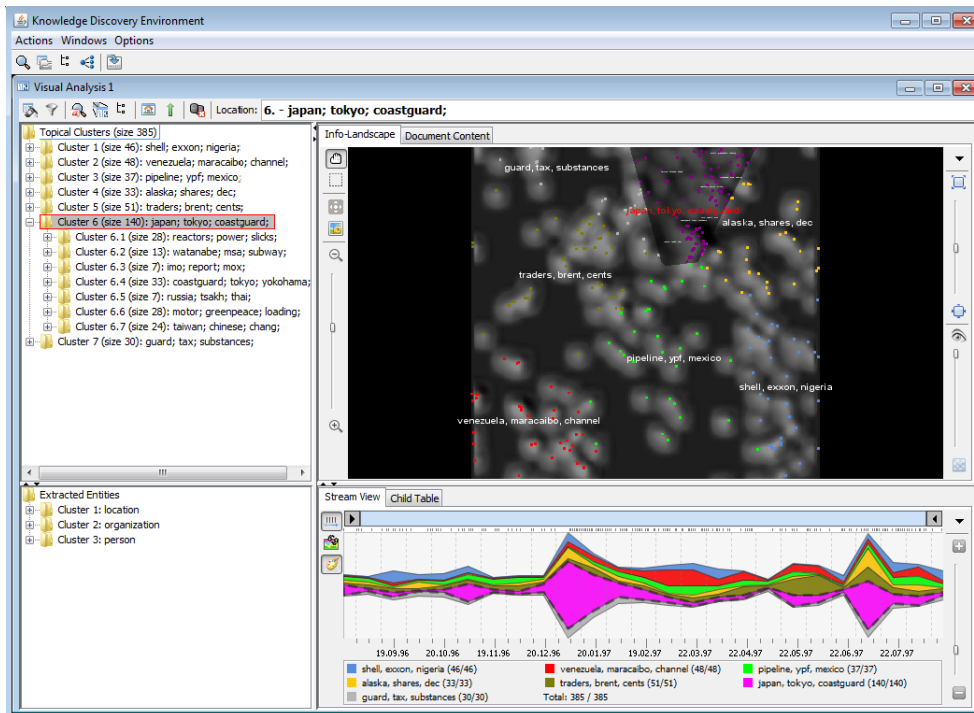


Figure 6.2: Topical and temporal visualization of all results returned for the query "oil spill".

- A topical hierarchy, computed by a clustering algorithm (see Section 4.2), is shown in the upper TreeView and in the Landscape3D.
- Faceted metadata categories for locations, organizations and persons, identified by information extraction methods (see Section 5.1.1), are shown in the lower TreeView.
- Temporal development of the topical clusters and metadata categories are shown in the StreamView (see Section 4.4).

Note that the the color assignments to clusters defined in the StreamView are coordinated with document colors in the Landscape.

As seen from the cluster labels, clustering partitions the data set along geographical regions (such as Japan, Venezuela and Alaska), oil companies (Shell, Exxon, YPF), technical (pipeline, substances, Brent) and commercial (traders, tax, shares) concepts, etc. Being interested in oil spills in Asia and Japan the user immediately spots that "Cluster 6: japan, tokyo, coastguard" (shown in magenta) exhibits two sharp temporal peaks in the StreamView. The first peak occurred in the period from end of December 1996 to end of

January 1997, the second one during the second half of June and the first half of July 1997. Two questions arise concerning the peaks:

1. Which events caused those two temporal peaks?
2. Are the events independent of each other or are they related or even causally connected?

In the following, the user will work towards finding answers to those questions using visual analysis methods.

6.1.2 Step 2: Topical Relatedness of Temporal Peaks

To understand whether the peaks have a topical relationship or not, one can make use of the information landscape combined with time-based selection of documents using StreamView. This is demonstrated in Figure 6.3. In the upper screenshot the user selects documents from the time interval corresponding to the first peak, using the interval selection component, which is the blue bar on top of the StreamView visualization. Here it is important to know, that selected documents are shown enlarged in the Landscape, while all other documents (i.e. those outside the chosen time interval) are tiny. In the lower screenshot the same can be seen for the second temporal peak.

By looking at "Cluster 6: japan, tokyo, coastguard" in the Landscape, and comparing the document selection in the upper and lower screenshot, the user can see that the documents from the first peak are predominantly placed towards the upper part of the cluster's area, while documents belonging to the second peak are concentrated towards the bottom. Knowing that similar objects are placed next to each other in the Landscape, the fact that documents from the two peaks occupy rather different areas indicate that the two peaks are not closely related in the topical sense. If they were, they would appear intermixed within the same area.

To conclude, a summary of what the user knows so far:

- All documents are relevant to "oil spill", as they were retrieved by that search query.
- Documents in magenta are related to "japan, tokyo", because of the cluster membership.
- Within the context "oil spill"/"japan, tokyo" the documents form two temporal peaks which appear to be topically rather unrelated.

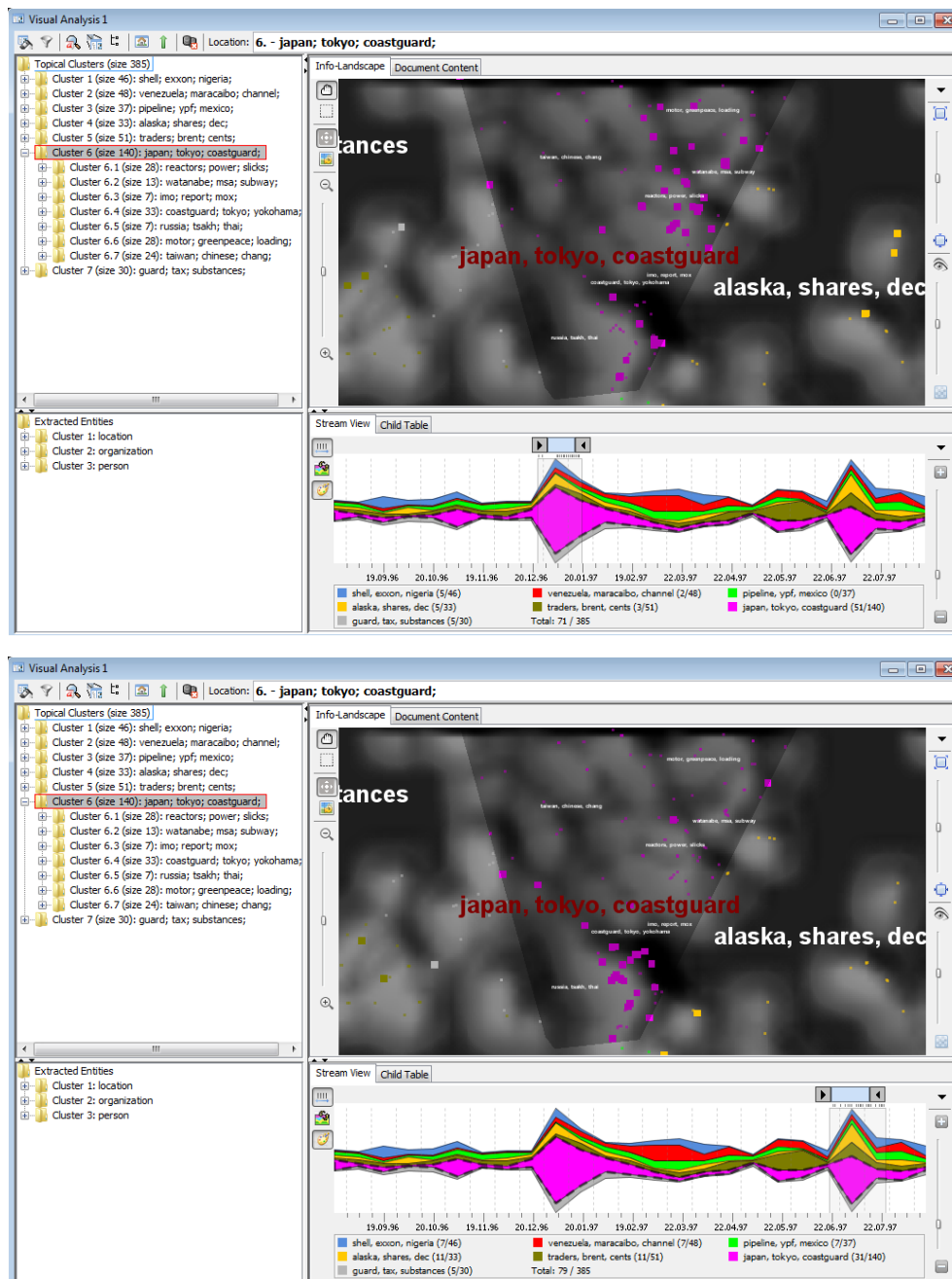
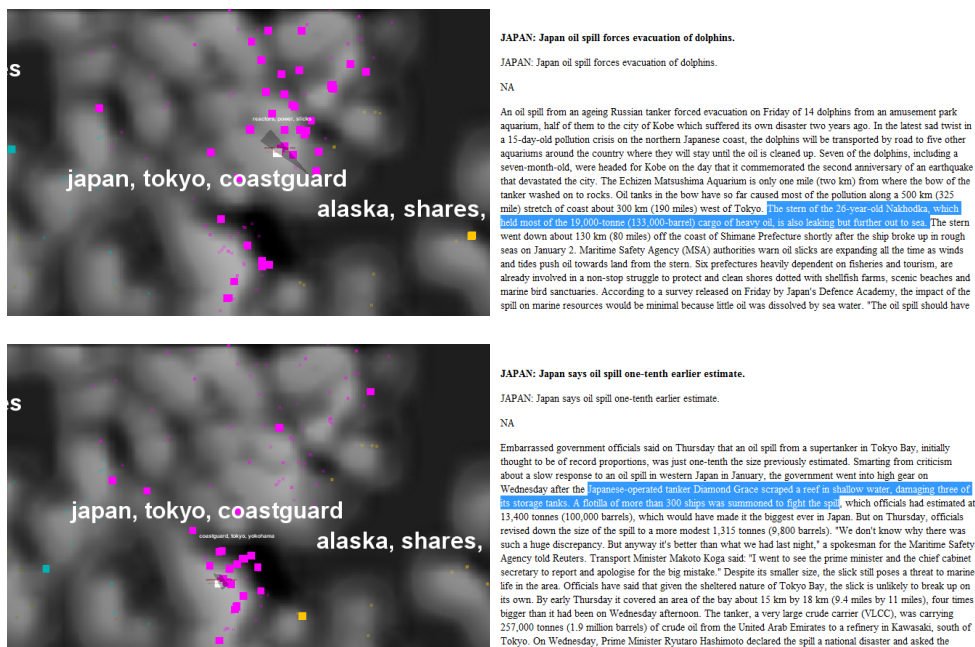


Figure 6.3: Selection of two temporal peaks with the time interval selection bar (selected documents are larger in the Landscape). In the Landscape documents from the first peak are predominantly placed at the top of the cluster, while those from the second peak are at the bottom, indicating that the peaks are topically unrelated.

The last point, if proved correct, indicates that the events corresponding to the temporal peaks are different in the sense of differing in their causes, involved participants, course of actions etc.

6.1.3 Step 3: Finding Possible Causes of Temporal Peaks

To find out what could have caused the two peaks, the user chooses one document from the first peak and one from the second, and takes a look on what is inside. In Figure 6.4 a selected document for the first peak is shown in white in the upper Landscape, and the document's content is shown on the right next to the Landscape. The same is shown for the second peak in the lower Landscape and document text. Note that in the user interface the text pane showing document content is placed in the tabbed pane shared with the Landscape.



JAPAN: Japan oil spill forces evacuation of dolphins.
 JAPAN: Japan oil spill forces evacuation of dolphins.
 NA
 An oil spill from an ageing Russian tanker forced evacuation on Friday of 14 dolphins from an amusement park aquarium, half of them to the city of Kobe which suffered its own disaster two years ago. In the latest said twist in a 15-day-old pollution crisis on the northern Japanese coast, the dolphins will be transported by road to five other aquariums around the country where they will stay until the oil is cleaned up. Seven of the dolphins, including a seven-month-old, were headed for Kobe on the day that commemorated the second anniversary of an earthquake that devastated the city. The Echizen Matsushima Aquarium is only one mile (two km) from where the bow of the tanker washed on to rocks. Oil tanks in the bow have so far caused most of the pollution along a 500 km (325 mile) stretch of coast about 300 km (190 miles) west of Tokyo. The stern of the 26-year-old Nakhodka, which had broken in two 13,000 tonnes (13,000 tonnes) super tanker, is also leaking but further out to sea. The stern went down about 130 km (80 miles) off the coast of Shimane Prefecture shortly after the ship broke up in rough seas on January 2. Maritime Safety Agency (MSA) authorities warn oil slicks are expanding all the time as winds and tides push oil towards land from the stern. Six prefectures heavily dependent on fisheries and tourism, are already involved in a non-stop struggle to protect and clean shores dotted with shellfish farms, scenic beaches and marine bird sanctuaries. According to a survey released on Friday by Japan's Defence Academy, the impact of the spill on marine resources would be minimal because little oil was dissolved by sea water. "The oil spill should have

JAPAN: Japan says oil spill one-tenth earlier estimate.
 JAPAN: Japan says oil spill one-tenth earlier estimate.
 NA
 Embarrassed government officials said on Thursday that an oil spill from a supertanker in Tokyo Bay, initially thought to be of record proportions, was just one-tenth the size previously estimated. Smearing from criticism about a slow response to an oil spill in western Japan in January, the government went into high gear on Wednesday after the Japanese-operated tanker Diamond Grace scraped a reef in shallow water, damaging three of its storage tanks. A flotilla of more than 300 ships was summoned to fight the spill, which officials had estimated at 13,400 tonnes (100,000 barrels), which would have made it the biggest ever in Japan. But on Thursday, officials revised down the size of the spill to a more modest 1,315 tonnes (9,800 barrels). "We don't know why there was such a huge discrepancy. But anyway it's better than what we had last night," a spokesman for the Maritime Safety Agency told Reuters. Transport Minister Makoto Koga said, "I went to see the prime minister and the chief cabinet secretary to report and apologise for the big mistake." Despite its smaller size, the slick still poses a threat to marine life in the area. Officials have said that given the sheltered nature of Tokyo Bay, the slick is unlikely to break up on its own. By early Thursday it covered an area of the bay about 15 km by 18 km (9.4 miles by 11 miles), four times bigger than it had been on Wednesday afternoon. The tanker, a very large crude carrier (VLCC), was carrying 257,000 tonnes (1.9 million barrels) of crude oil from the United Arab Emirates to a refinery in Kawasaki, south of Tokyo. On Wednesday, Prime Minister Ryutaro Hashimoto declared the spill a national disaster and asked the

Figure 6.4: Choosing and reading one document (shown in white) from the first temporal peak (upper row) and one from the second peak (lower row). The first document mentions an oil spill caused by the Russian ship "Nakhodka", the second one an oil spill caused by the Japanese-operated tanker "Diamond Grace".

The content of the document from the first peak reveals that the Russian ship "Nakhodka" caused an oil spill about 300km west from Tokyo. The document from the second peak states that the Japanese-operated tanker "Di-

among Grace” caused an oil spill in the Tokyo Bay. These could be the possible causes of for two peaks, but to know for sure one needs to check whether all (or a significant majority) of the documents from the first peak mention the Russian ship ”Nakhodka”, but are unrelated to the Japanese-operated tanker ”Diamond Grace”. The opposite should be shown for the second peak. As reading every document to confirm or reject this would be very time consuming (especially when involved data sets are large) the user will attempt to answer this question using visual means.

To conclude, a summary of what the user knows at this point:

- Documents on oil spill in the area Tokyo, Japan have two distinct temporal peaks which appear topically unrelated.
- Sample documents from each peak mention different possible causes for the peaks: Russian ship ”Nakhodka” and Japanese-operated tanker ”Diamond Grace”.

The second point is a completely unproven hypothesis at this point, which needs to be validated by further analytical steps.

6.1.4 Step 4: Validating the Hypothesis using Faceted Metadata

To validate the above postulated hypothesis the user resorts to faceted metadata categories extracted by the information extraction techniques (see Section km-modules). The idea behind using faceted metadata categories is to correlate documents containing a particular extracted metadata instance with the temporal peaks. In this case the user will look for correlations between the temporal peaks and the operator countries of the ships. This is performed by selecting the locations ”Russia” (the origin of the first ship) and ”Japan” (country operating the second ship) and checking whether documents containing these two metadata instances correlate with the respective temporal peaks.

Faceted metadata categories are available in the TreeView on the lower-left, below the topical cluster hierarchy. The two screenshots in Figure 6.5 show the location ”Russia” highlighted in the tree. This has two effects on the visualizations:

1. The StreamView shows the faceted metadata category ”Russia” along the topical clusters. The stream for ”Russia” is shown in cyan.

2. Landscape shows documents mentioning "Russia" also colored in cyan. All other documents are now shown in white (using topical cluster colors is disabled here for clarity).

It is important to remember that documents selected by the time interval selection bar are shown enlarged, while documents from outside of the selected time interval are small. The upper screenshot shows documents from the first peak selected, the lower one from the second.

Looking at the StreamView in Figure 6.5 (same in both screenshots), shows that Russia correlates strongly with the first peak, but not at all with the second one. However, an experienced user will notice that during the first peak the stream for "Russia" is thinner than the stream for "Cluster 6: japan, tokyo, coastguard". This means that only a subset of documents from the first peak mentions Russia. Nevertheless, this adds strength to the hypothesis that the oil spill caused by the Russian ship Nakhodka is responsible for the the first peak, but is not related to the second one.

Similar can be learned from the Landscape. For the first peak selected (upper screenshot) only a part of the enlarged documents is in cyan. For documents from the second peak (lower screenshot) no documents are in cyan. This fits the conclusion derived from the StreamView: "Russia" correlates strongly, but not decisively, with the first peak, while it shown no correlation with the second.

Unfortunately, the attempt to perform the same kind of analysis with "Japan" as the highlighted faceted metadata category does not provide any new insights. As can be seen in Figure 6.6 Japan correlates strongly with both peaks, in the StreamView and in the Landscape - which was to be expected. The reason being, that all documents in "Cluster 6: japan, tokyo, coastguard" are obviously strongly related to Japan, so that Japan cannot be used to differentiate between the peaks.

A summary of what the user has discovered in this step is as follows:

- Temporal distribution of documents mentioning "Russia" correlates strongly with the first temporal peak, but not at all with the second.
- "Russia" is mentioned only in subset of documents from the first temporal peak (approx. 40-50

This information provides some insight, but is definitely insufficient for conclusions. While the chosen validation strategy is valid and often useful, this examples demonstrates that the features chosen to perform the correlation analysis have to be chosen carefully. The user will have to readjust the strategy.

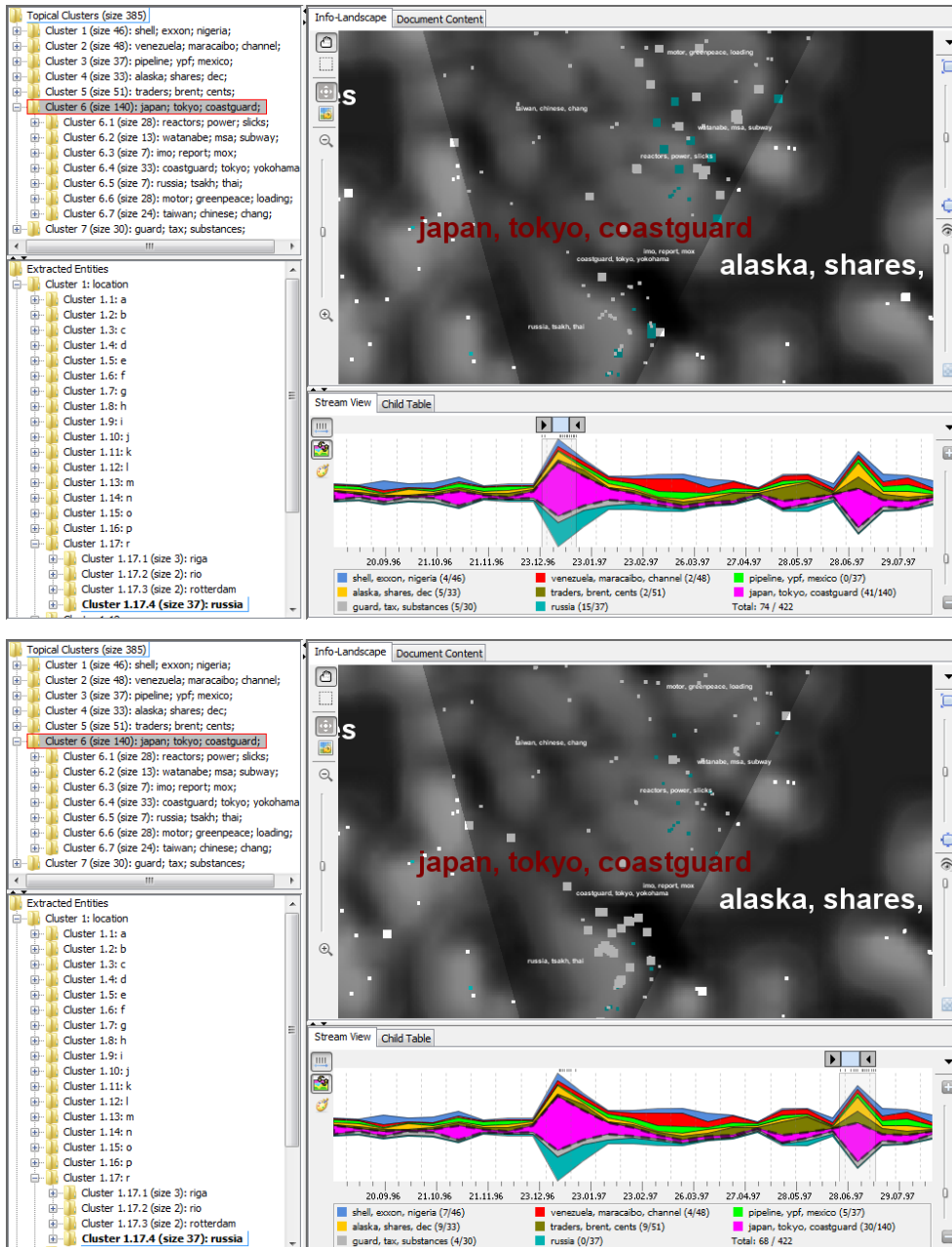


Figure 6.5: Displaying faceted metadata category for location "Russia" (in cyan): In the StreamView it correlates strongly with the first temporal peak, not with the second. In the landscape the same can be seen. However, only a subset of documents from the first peak mention "Russia".

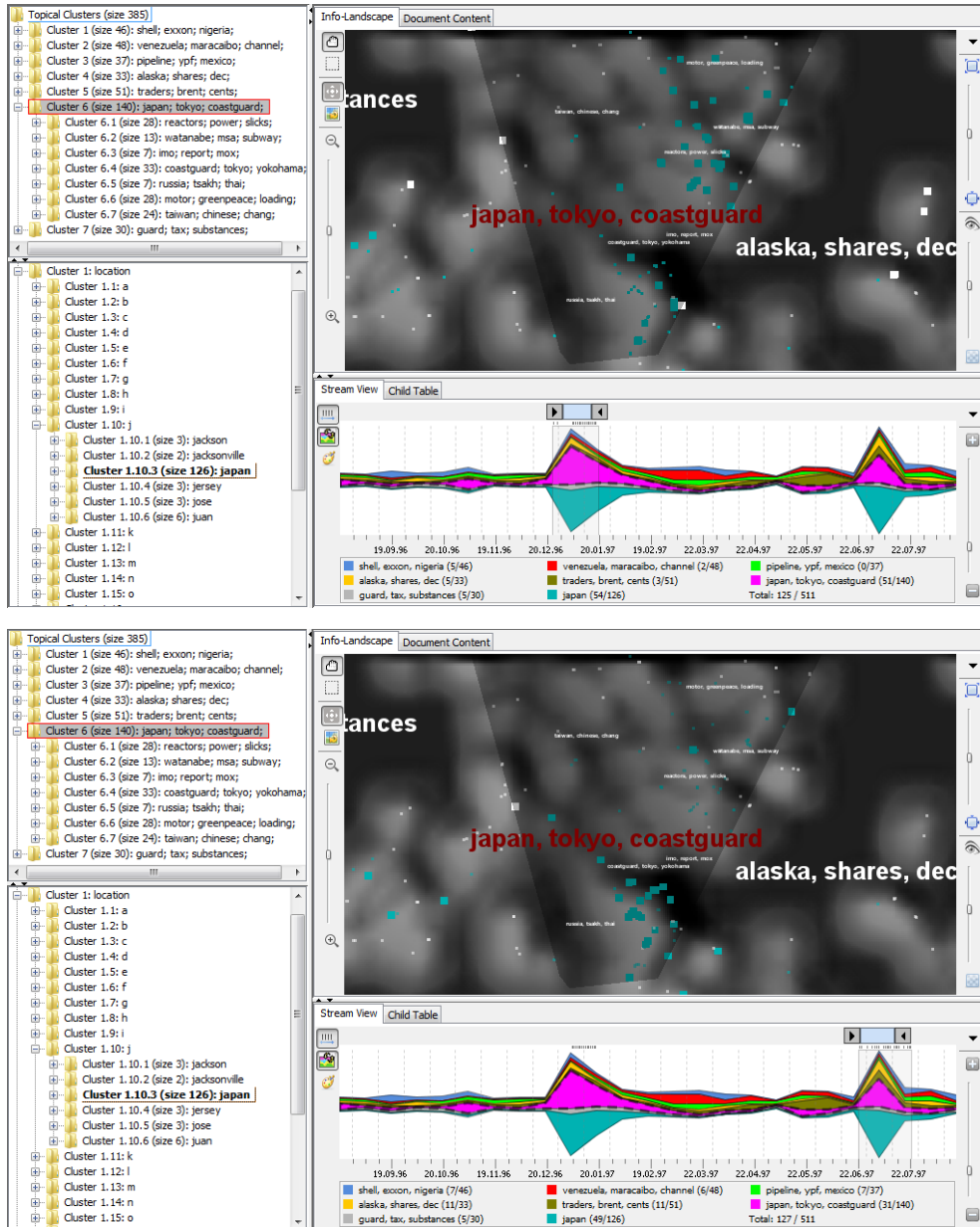


Figure 6.6: Displaying faceted metadata category for location "Japan" (in cyan): Correlates with both temporal peaks, in the StreamView and in the Landscape, making no conclusions possible. The reason being, that "Japan" is the main feature of the topical cluster, which is mentioned in the majority of the cluster's documents.

6.1.5 Step 5: Validating the Hypothesis using Retrieval

To validate the hypothesis generated in Step 3 (Subsection 6.1.3) the user can try applying a strategy very similar to the one applied previous step, however with the use of retrieval instead of faceted metadata. To do this the user searches for the ship names, "Nakhodka" and "Diamond Grace", and observes how found documents correlate with documents from the respective temporal peaks.

The results can be seen in Figure 6.7 for "Nakhodka" and in Figure 6.8 for "Diamond Grace". Note that the search hits are shown in red in the Landscape, while the StremView is used only for temporal selection of documents using the time interval selection bar. As in the above examples, documents from the selected time interval are shown enlarged in the Landscape, those outside the interval are small.

What the user can see by looking at the Figure 6.7 it that "Nakhodka" is mentioned in almost all documents from the first peak (upper screenshot), but in only very few from the second (lower screenshot). Figure 6.8 shows that the opposite is true for "Diamond Grace", which is not at all mentioned in the documents from the first peak (upper screenshot), but is present on all documents (except one) from the second peak (lower screenshot). An uncertainty which remains is why "Nakhodka" is also mentioned in some documents from the second event. By inspecting one of those documents the user will see that the main topic is actually about the "Diamond Grace" oil spill, but "Nakhodka" is mentioned since it caused a similar problem at the similar location just six months earlier.

Considering these results the user can now conclude that following statements hold with high probability:

- "Nakhodka" is responsible for the first temporal peak and "Diamond Grace" for the second.
- The two temporally separated peaks are two separate oil spill events in the vicinity of Tokyo, Japan, caused by different ships (from different countries).

This conclusions confirm the hypothesis which was generated by explorative navigation actions performed in steps 2 and 3.

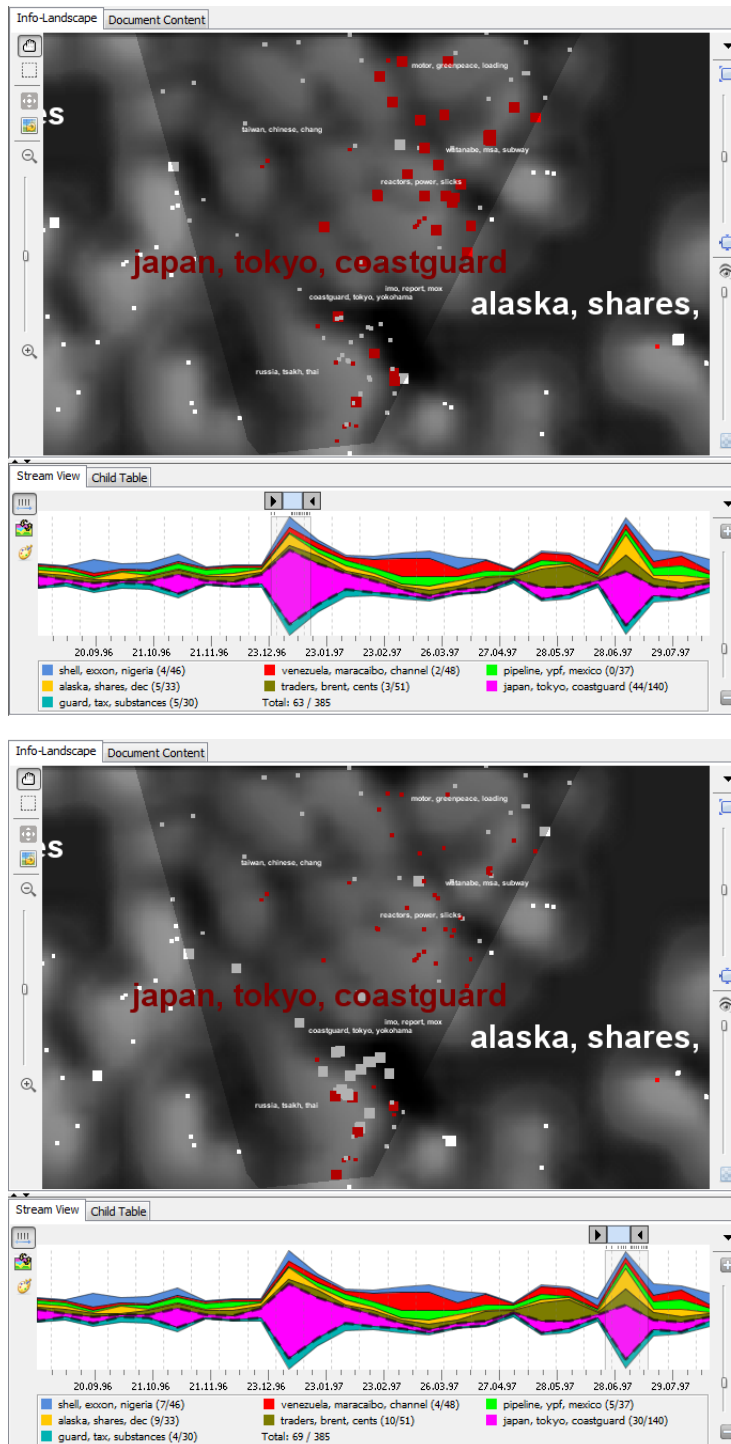


Figure 6.7: Searching for "Nakhodka" and showing hits in red. In the landscape correlates predominantly with the first event (top), and mostly not with the second one (bottom).

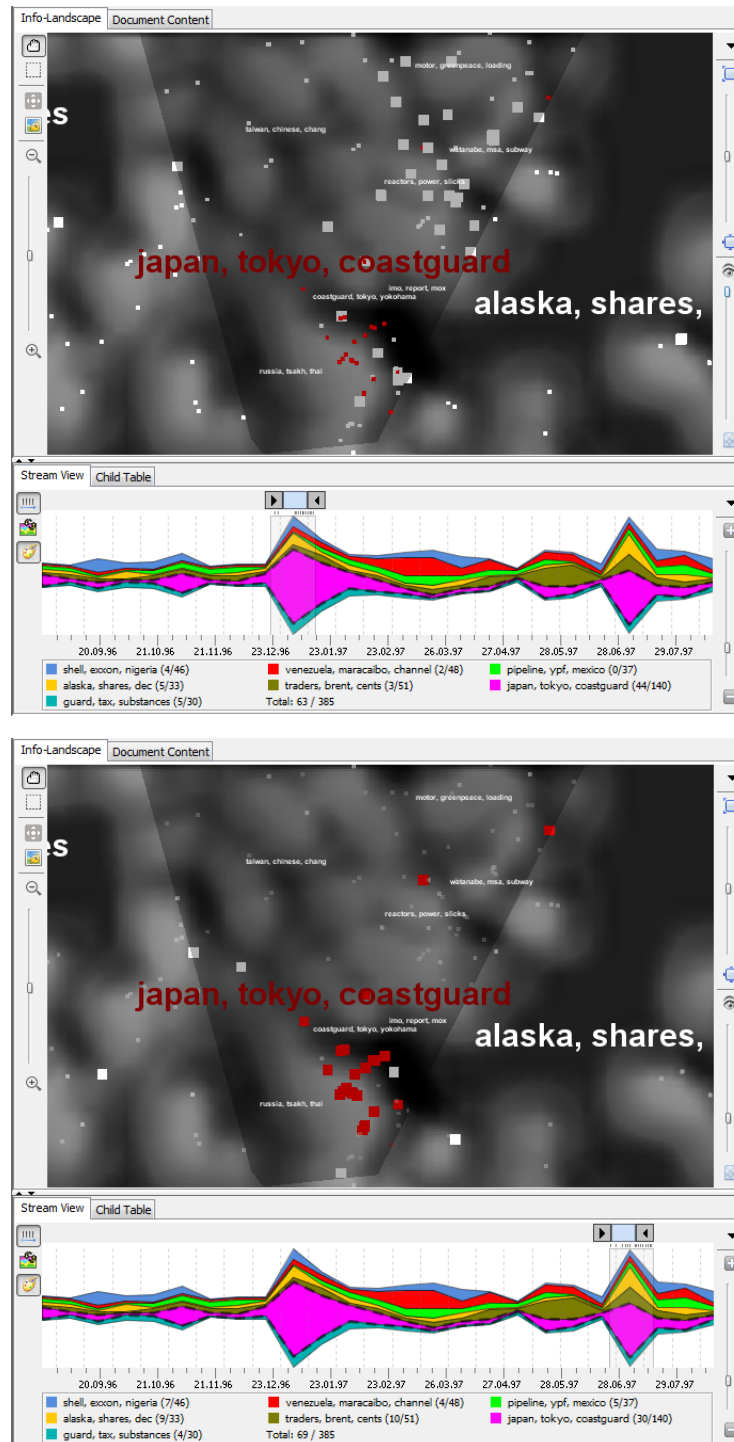


Figure 6.8: Searching for "Diamond Grace" and showing hits in red. In the landscape clearly correlates only with the second event (bottom), definitely not with the first one (top).

6.2 Semantic Mediation

Semantic Mediation Tool (SMT) applies visual methods to facilitate collaborative, semi-automatic mediation between a pair of knowledge bases (ontologies). The mediation process consist of two major phases:

1. Automatic computation of mappings between concepts from different ontologies.
2. Reviewing (i.e. accepting or rejecting) of mapping suggestions created in the first step by the users.

In the following three use cases are demonstrated: creating a new mediation, collaborative reviewing, and drill down to the area of interest.

6.2.1 Starting a Mediation Process

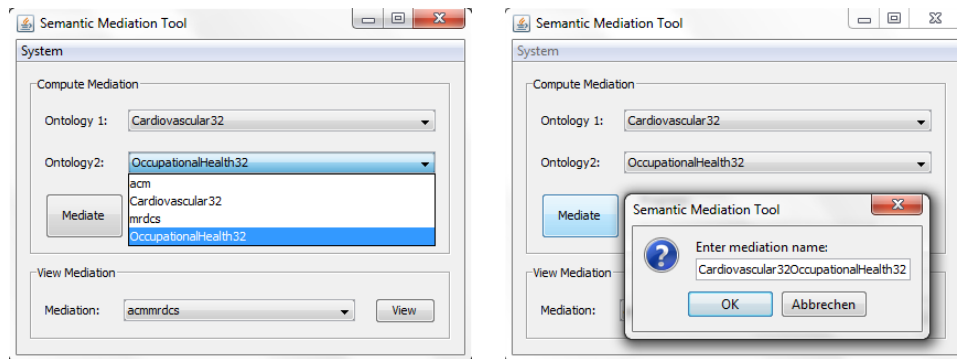


Figure 6.9: After logging in the user can choose a pair of ontologies (left) and start the mediation process (right).

Creating a new mediation is a simple process shown in Figure 6.9. It begins by choosing a pair of ontologies to be aligned, assigning a name to the new mediation, and starting the automatic part of the mediation process. Once the algorithms are done computing mapping suggestions between concepts, the mediation is saved. At this point the users can begin with the reviewing process in which the computed mappings are accepted or rejected.

Results of the automatic alignment are shown in the reviewing window which can be seen in Figure 6.10. The list of mapping suggestions sorted by estimated confidence is shown in the mapping table (on the upper-left). At the beginning all mappings have the state "suggested", which can be changed by the reviewer to "accepted" or "rejected". Two different visual representations

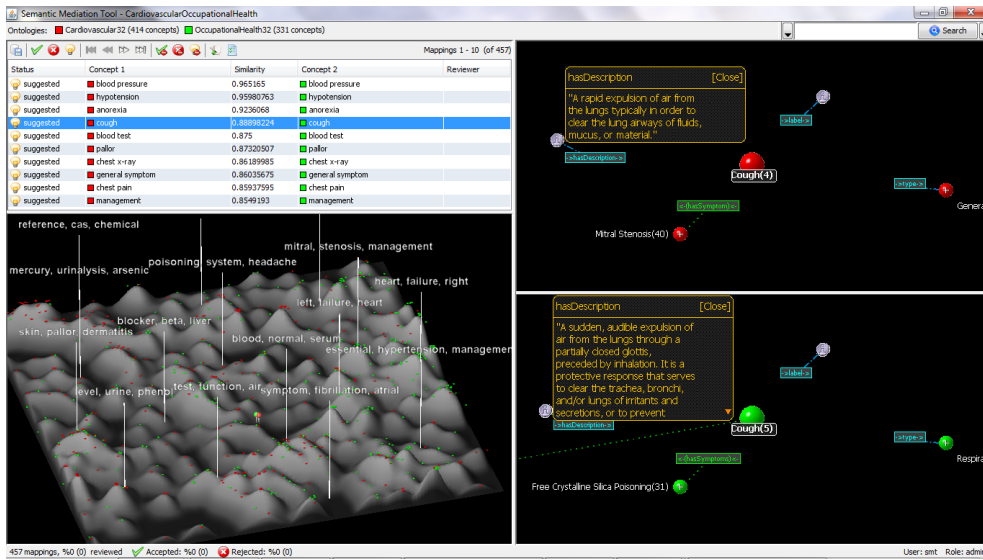


Figure 6.10: Result of mediating two medical ontologies: table of mapping suggestions is on top-left, an information landscape visualizing all concepts from both ontologies is on bottom-left, and two graph visualizations for browsing the ontologies are on right.

are available to support the user during the reviewing task: the Landscape visualization (on bottom-left) provides an overview of all concepts involved in the mediation process, while the two Multimedia Semantic Browsers (MMSB, on right) provide a graph visualization for ontology navigation. The example shown in Figure 6.10 shows the result of mediating two small medical ontologies: Cardiovascular with 414 concepts and OccupationalHealth with 331 concepts. The user can now begin with the process of reviewing the suggested mappings, which is described in the next example.

6.2.2 Collaborative Reviewing

SMT facilitates collaborative reviewing of suggested mappings by selecting a subset of the mappings and assigning them to a user for reviewing. Partitioning of the suggested mappings into tasks is primarily useful for splitting a large data set between users to reduce individual workload. However, it also makes sense to assign mapping suggestions to tasks in such a way, that tasks correspond to a particular domain or field. Such a task should be assigned to an expert user who is knowledgeable in the particular domain or field.

A collaborative reviewing scenario involves the following steps:

1. **Area selection:** Information landscape is useful at providing an overview and identifying major groups of related concepts. The user (administrator) selects a chosen group of concepts, where labels provide guidance revealing what topics a particular area of the Landscape is covering. In Figure 6.11 selection of concepts using the lasso selection tool (white curve) in the area "left/right, heart, failure" can be seen.
2. **Task assignment:** In Figure 6.12 concepts selected in the previous step are shown enlarged. The effect of selection is that the mapping table will show only mappings between the selected concepts, resulting in 10 mappings (down from 457 total mapping). These 10 mappings are assigned as a task to the user "jim" and saved under the name "heart failure" (see Figure 6.13). The administrator can create and assign more tasks by repeating the procedure of selecting areas of related concepts in the Landscape, and assigning the corresponding mapping suggestions to users for reviewing.
3. **Reviewing:** When the user "jim" logs into the system next time, the only mappings suggestions available in the table will be those from the "heart failure" task, which can be seen in Figure 6.14. User "jim" will only be able to review these mapping suggestions, access to all others will be blocked. Reviewing a mapping includes selecting the mapping in the table, which shows the two concepts in the semantic browsers (on right). A semantic browser shows detailed information on a concept within the ontology it originates from, to support the user in making decisions. The user accepts (green check mark icon) or rejects (red cross icon) the mapping depending on the presented information.
4. **Progress monitoring:** The administrator can follow the reviewing progress for each task and for the whole data set using the task overview table, shown in Figure 6.15. It shows reviewing progress information for five tasks of different sizes assigned to different users. The administrator can follow the progress for single tasks (table entries), for all tasks (status bar, on left), and can see the percentage of mappings suggestions already assigned to different tasks (status bar, on right).

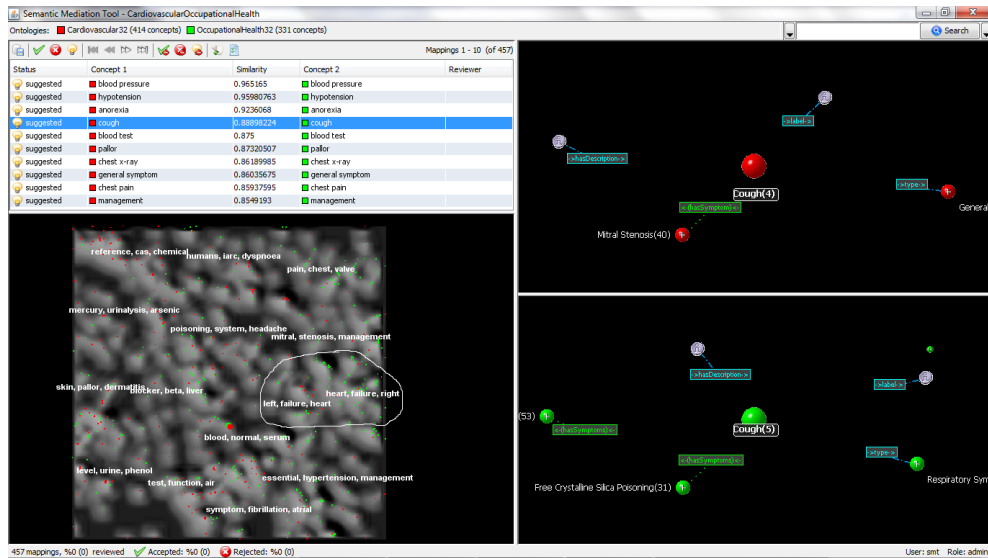


Figure 6.11: Selecting the area "left/right, heart, failure" with the lasso selection tool.

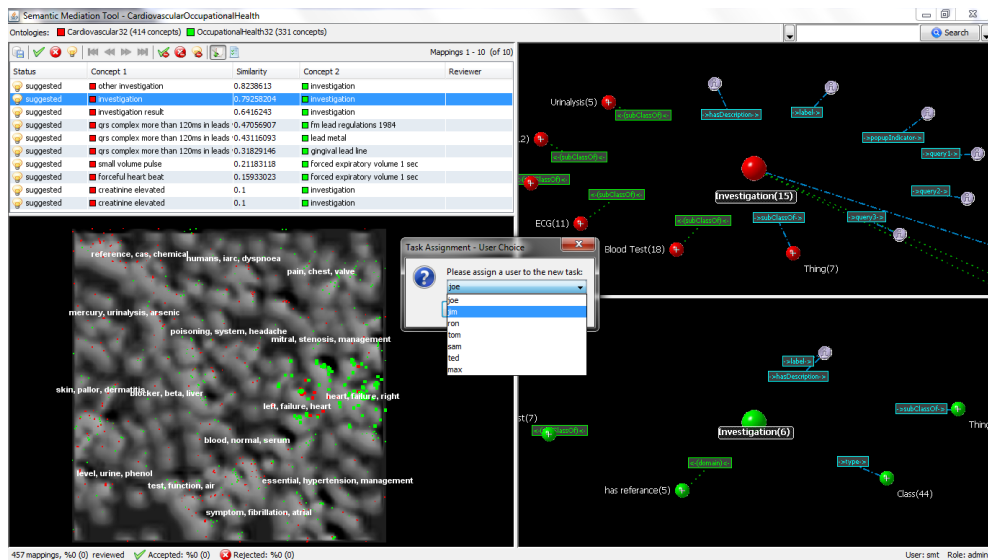


Figure 6.12: Assign the 10 mappings between selected concepts (enlarged in the information landscape) to user "jim".

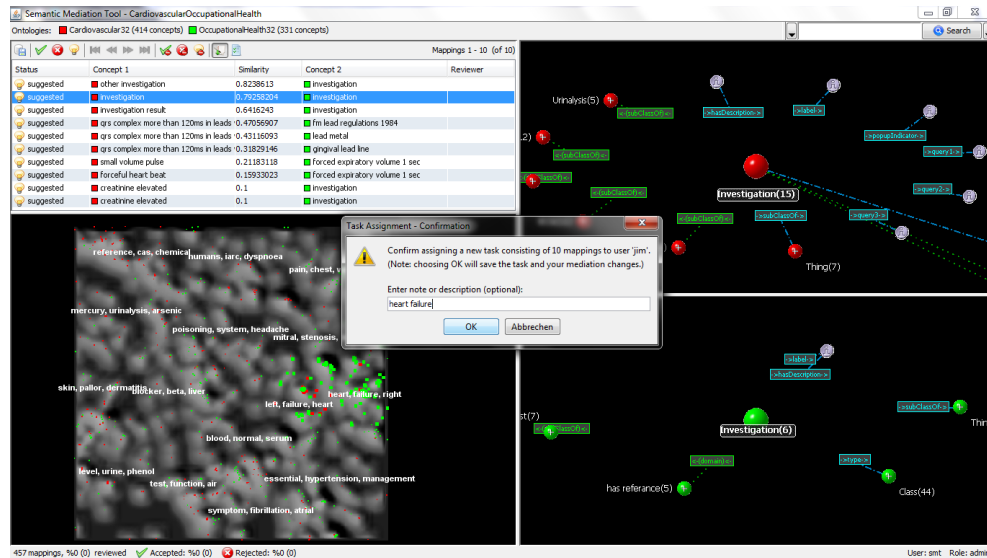


Figure 6.13: Save the newly created task under the name "heart failure".

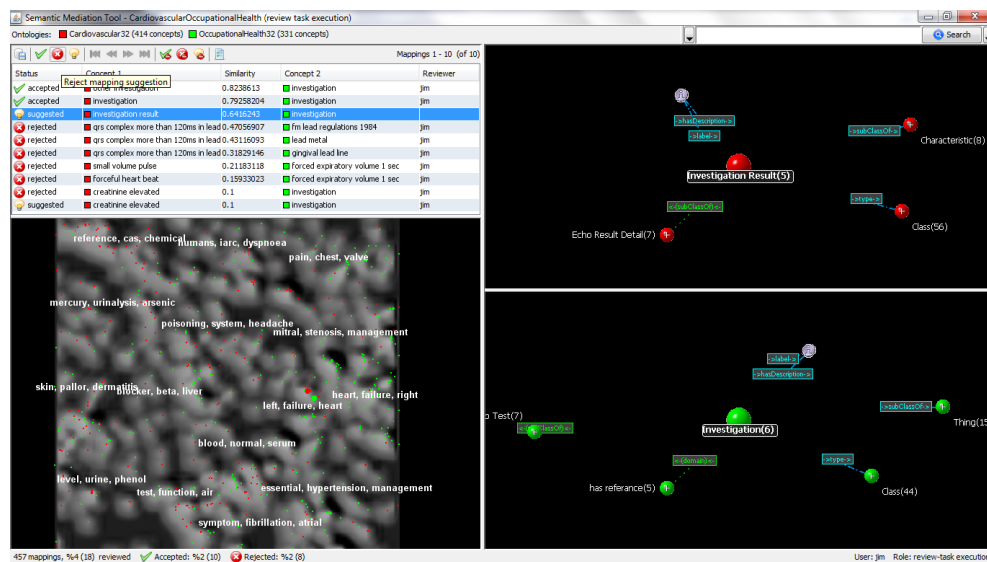
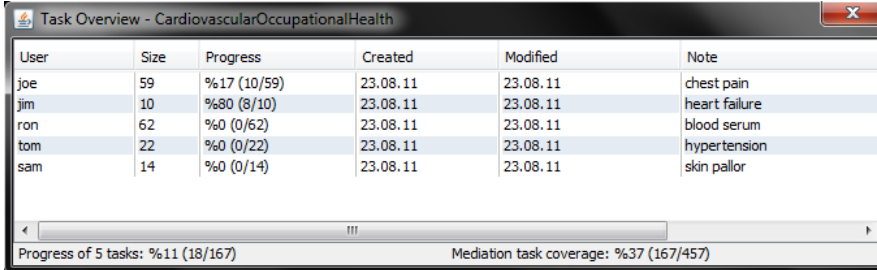


Figure 6.14: After logging in as user "jim", only mappings assigned to his task ("heart failure") can be reviewed. A few mapping suggestions have been accepted (green check mark), several others rejected (red cross) in this example.



User	Size	Progress	Created	Modified	Note
joe	59	%17 (10/59)	23.08.11	23.08.11	chest pain
jim	10	%80 (8/10)	23.08.11	23.08.11	heart failure
ron	62	%0 (0/62)	23.08.11	23.08.11	blood serum
tom	22	%0 (0/22)	23.08.11	23.08.11	hypertension
sam	14	%0 (0/14)	23.08.11	23.08.11	skin pallor

Progress of 5 tasks: %11 (18/167) Mediation task coverage: %37 (167/457)

Figure 6.15: The administrator can view the progress of tasks assigned to different users. Here, two tasks out of five show progress.

6.2.3 Drill Down to Area of Interest

This example demonstrates the usefulness of the Landscape visualization when dealing with large amount of concepts. The landscape provides overview and orientation in the different topical areas covered by the aligned concepts, and offers means for explorative navigation toward areas matching the user's needs and interests.

In this example two technical classification systems were aligned:

- The ACM Computing Classification System [ACM 1998].
- Malaysian Research and Development Classification [MRDCS 2011], which is a set of classifications designed for use in the measurement and analysis of research and development activities in Malaysia.

Since the total number of concepts is larger than in the previous example (745 to 5110), the role of the information landscape gains importance, both as an overview tool and as an orientation and navigation help. These features become more important when larger ontologies should be mediated, for example when user's focus is on one or more specific areas. Landscape empowers the user to drill down directly to the area of interest guided by labels provided by the cluster hierarchy.

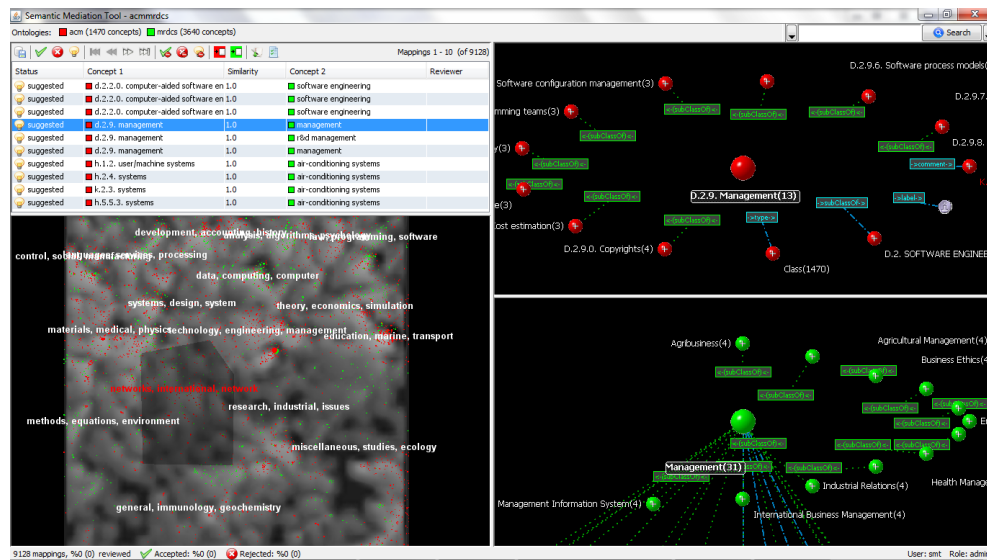


Figure 6.16: Mediating the ACM and MRDCS classification systems yields 9128 mapping suggestions. A user with focus on computer networks can immediately identify the area "networks, international, network" in the mid-left part of the concept landscape.

Figure 6.16 shows the result of automated aligning of 5110 concepts from the ACM and MRDCS classification systems, which produces a large number (9128) of mapping suggestions. To narrow down the mapping suggestions to those corresponding to user's particular interest - computer networks - the user inspects the labels describing different areas of the Landscape. The area "networks, international, network" (label in red) appears as the appropriate starting point.

An example of a drill down is shown in Figure 6.17. Beginning from the "networks, international, network" area (top image), the user propagates toward an area containing specific concepts by following the labels. By zooming in on area "networks, international, network" more detailed labels appear describing smaller areas (middle image), where the sub-area "networks, network, nets" is highlighted. Finally, by zooming in on "networks, network, nets" (bottom image), the user gets the possibility to choose between specific network types, with "computer, information, protocols" coming closest to the desired topic (computer networks).

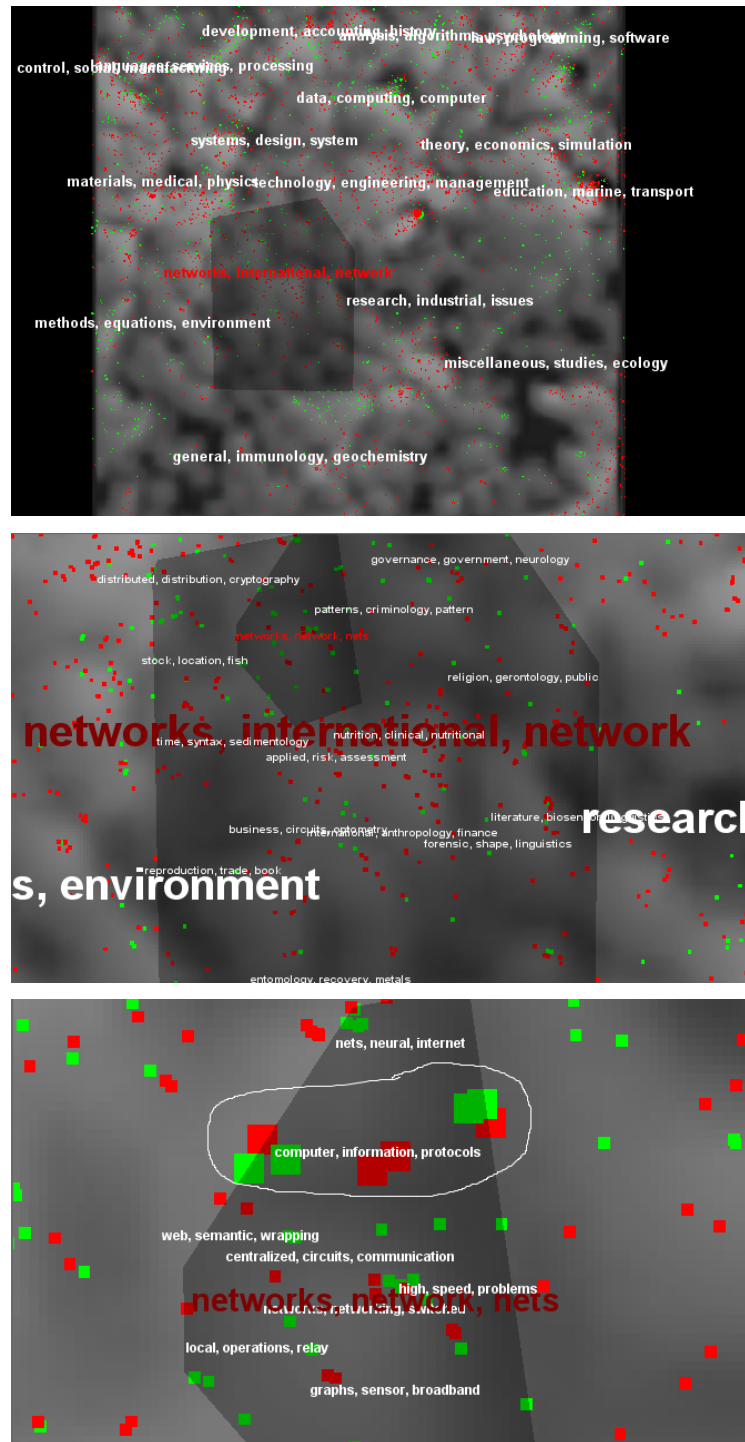


Figure 6.17: Drill-down to the area of interest - computer networks - by following the labels of the topical cluster hierarchy. Concepts are selected using lasso-selection (bottom).

Ontologies: ■ acm (1470 concepts) ■ mrdcs (3640 concepts)

Mappings 1 - 10 (of 9128)

Status	Concept 1	Similarity	Concept 2	Reviewer
suggested	d.2.2.0. computer-aided software en 1.0		software engineering	
suggested	d.2.2.0. computer-aided software en 1.0		software engineering	
suggested	d.2.2.0. computer-aided software en 1.0		software engineering	
suggested	d.2.9. management	1.0	management	
suggested	d.2.9. management	1.0	r&d management	
suggested	d.2.9. management	1.0	management	
suggested	h.1.2. user/machine systems	1.0	air-conditioning systems	
suggested	h.2.4. systems	1.0	air-conditioning systems	
suggested	k.2.3. systems	1.0	air-conditioning systems	
suggested	h.5.5.3. systems	1.0	air-conditioning systems	

Ontologies: ■ acm (1470 concepts) ■ mrdcs (3640 concepts)

Mappings 1 - 6 (of 6)

Status	Concept 1	Similarity	Concept 2	Reviewer
suggested	i.2.4.6. semantic networks	0.5087408	neural network computer	
suggested	i.2.4.6. semantic networks	0.4838852	computer communications networks	
suggested	c.2.2. network protocols	0.48316997	neural network computer	
suggested	c.2.2. network protocols	0.4793088	computer communications networks	
suggested	h.3.4.2. information networks	0.4789947	neural network computer	
suggested	h.3.4.2. information networks	0.45745507	computer communications networks	

Figure 6.18: After the drill-down and selection of topic of interest the amount of suggested mappings was narrowed down from 9128 (top) to merely 6 (bottom).

Selecting concepts of the area "computer, information, protocols" will update the table to show only mapping suggestions where these concepts occur. As seen in Figure 6.18, this reduces the number of mappings in the table from 9128 to just 6, allowing the user to focus on concepts of interest.

Explorative navigation capabilities of the information landscape provide a lot of flexibility to the user in scenarios such as the described one. Guided by the labels, the user can try selecting different areas in the Landscape and see immediately in the mapping table whether the selected concepts fit the topic of interest or not. For example, the user might want to extend the selected area to include potentially interesting neighboring concepts, or dig deeper in the hierarchy to provide more focus. Guidance provided by the labels makes the described method particularly effective when user's goals are not precisely defined or when the user is unfamiliar with the data and with the used vocabulary.

6.3 Production Scenarios

Visual and algorithmic methods described in this work were integrated with the m2n Intelligence Management Framework [m2n IMF 2011] resulting in a productive system covering a variety of knowledge discovery, information retrieval and visual analysis scenarios. Development of new technologies and their application in real-world scenarios was performed in a 5-year applied research project, me being the project leader at the Know-Center. m2n, the industrial partner in the project, is an Austrian company offering a semantic technology-based Intelligence Management Framework. The resulting system was installed, tested, and finally productively applied in the application domains of business intelligence and governmental document management, with installations in other application domains to follow.

6.3.1 Business Intelligence

The business intelligence system was designed for analysis of large, growing patent databases and repositories of related scientific publication. Use cases addressed by the system [Atzmüller & Sabol 2007] include the following:

- Advanced retrieval using concept search, search by example, collaborative searching, associative searching, etc. (see Section 5.1.1).
- Explorative analysis of topics, relationships and metadata in large data sets using the Landscape component (see Sections 4.3.1 and 4.3.2).
- Analysis of topical trends using the StreamView component and the dynamic topography information landscape (see Sections 4.4 and 4.3.3).
- Definition and maintaining of document distribution profiles using classification and dynamic topography information landscape.

While the first three use cases have already been discussed in one form or another, the last use case has not, but deserves some attention. The idea behind it is to use a classifier for automatically dispatching new patents and publications to persons and groups with specific interests. The recipients of the dispatched documents need to define the training set of the classifier and they do that in cooperation with the patent expert who maintains the classifier. Note that the technological enablers for the use case are the incremental ordination algorithm described in Section 4.2.1 and the information landscape with dynamic topography presented in Section 4.3.3.

The workflow of the use case involves the following steps [Sabol et al. 2009a]:

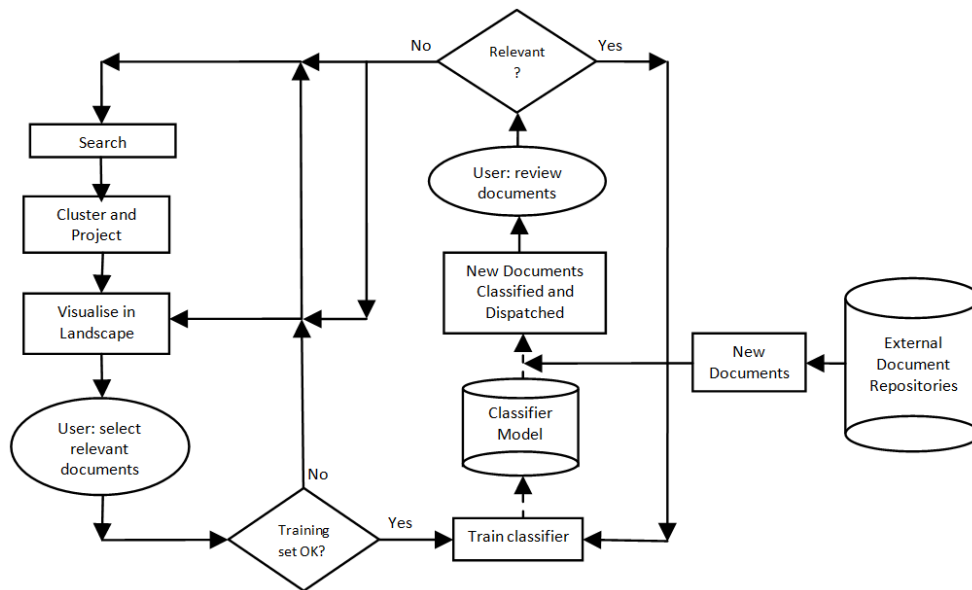


Figure 6.19: Classifier refinement workflow using a dynamic topography information landscape.

1. Using search to find an initial set of documents describing the area relevant to the recipient.
2. Using the Landscape visualization to perform:
 - topical disambiguation by eliminating non-relevant clusters,
 - elimination of outlier documents.

This step is necessary because the search is likely to return some non-relevant and ambiguous results. Visual analysis is applied because the amount of documents defining a class may be far too large (10000 documents and more) for manual inspection.

3. Creating a new classification category and training the classifier with the documents remaining from the previous step.
4. Daily classification and dispatching of new documents from external sources to the recipients.
5. Periodical refinement of the training set through repetition of steps 1 to 3.

The sketch of the workflow can be seen in Figure 6.19.

The motivation behind the last step is that every day new documents from external sources are classified and dispatched to different recipients. However, due to the fact that new technologies are being patented and published, and that vocabulary gradually changes with time, it is necessary to readjust and extend the original training set periodically. To achieve this, new documents which are potentially good candidates for the training set are added to the information landscape previously created in the step 2, and analyzed within the context of the training set.

As the users performing the analysis are already acquainted with the topical and spatial configuration of the information landscape created in the step 2, it is important to integrate the new documents in the existing visual and topical structures. Through the recognition of old, familiar structures the user can quickly judge whether the new documents integrate within exiting topical structure or not. When new topical structures arise the relatedness to the existing topics is an indicator of whether these are relevant to the user or not.

The described workflow demonstrates, albeit only in a basic form, another interesting opportunity which arises through the use visualization: the utilization of user feedback for improvement of algorithmic methods, in this case classification. A more advanced approach to this interesting field of research, which uses dedicated classification visualization techniques, is described in [Seifert et al. 2010b].

Chapter 7

Evaluation

To test the performance and improve the usability of developed visual components and applications, usability evaluation (see Section 2.1.2) was performed in several forms. During the design phase, heuristic evaluation of the components and of the planned interactivity model were performed by usability experts to identify design flaws and shortcomings as early as possible. Formal experiments and thinking aloud tests were performed with test users in the later development phases, and the components were improved and tuned according to the findings. Finally, during pilot installations of the systems using the developed visual techniques, user feedback was collected resulting in further improvements.

The backbone of the efforts toward achieving good usability were formal experiments, designed to consider and extend on the results of previously performed usability tests [Granitzer et al. 2004]. Results of the previous formal experiment, and especially the collected user feedback, strongly suggest that combining an information landscape visualization with tree and table components into a coordinated user interface provides tangible advantages for explorative analysis scenarios (see Section 3.3.6). Such a combined user interface provided higher rates of successful test tasks completion than an interface without the visualization, or an interface which included only the visualization. In interviews users stated their preference for an interface including the information landscape, a tree and a table. These results determined the direction for user interface design of the KDVE and SMT applications. Both of them use the information landscape and employ the coordinated multiple views paradigm to integrate further visualization components, such as trees, tables, StreamView, graph visualization-based ontology browsers, relevant to their respective application domain. These include trees, a StreamView and a table for KDVE, and for SMT a table and graph visualization-based ontology

browsers.

More recently, additional experiments were performed which can be subdivided into two groups:

1. Evaluation of a user interface for topical-temporal analysis composed of a Landscape and a StreamView components.
2. Evaluation of selected aspects of explorative analysis using the information landscape.

Performed usability tests were conducted by two students within their bachelor thesis, [Weitlaner 2009] and [Krnjic 2008], under my tutorship and supervision. It should be noted that the choice of features and functionality which were tested, was also driven by the practical need of users involved in the pilot installations of the productive system outlined in Section 6.3. This chapter describes these two groups of usability experiments, and presents their results. To conclude, a brief discussion of lessons learned is given and an outlook for possible further usability improvements is outlined.

7.1 Testing Methodology and Environment

Design of the tests, including the detailed description of the tasks to be performed, were developed by me in cooperation with the students performing the tests. Experiments were performed to compare the performance of two different user interfaces, U1 and U2, and determine which of the two interfaces performs better for a particular tasks or group of tasks. For example, use of visualization vs. no visualization or automatic navigation support vs. manual navigation, were tested.

Each experiment was performed on a group of 10 test users which, with each user performing 2-3 (depending on experiment) simple tasks on each user interface. For each task a reasonable time limit was defined which, when exceeded, resulted in a timeout meaning that the user could not complete the task successfully. Within-groups experiment configuration was chosen to allow the test users to provide direct comparative statements on which user interface they preferred. In withing-groups configuration the users are subdivided into two groups of equal size. Each user performs the tasks on both user interfaces, however each group begins with the different interface.

As each user performs the same tasks on both user interfaces, the effects of learning affect the outcome of the experiment. To limit the effects of learning, each task is defined on two different, but similar data subsets, D1 and D2. Performing the task on different data sets yields different outcomes, although

the task is the same. For example, a user who has performed a task "estimate size" on some "Cluster 1" (D1) using interface U1, will execute the same task on "Cluster 2" (D2) using interface U2, yielding different estimated cluster sizes. If the data sets were not different, the user would likely be faster with the second interface, as the exact outcome of the test would be known from performing it with the first user interface. As with the order of the user interfaces (U1 and U2), the order of D1 and D2 is also switched. Order of user interfaces and data subsets for each user is given by the following table:

User	Interface/Dataset	Interface/Dataset
Person 1	U1/D1	U2/D2
Person 2	U2/D2	U1/D1
Person 3	U1/D1	U2/D2
Person 4	U2/D2	U1/D1
Person 5	U1/D1	U2/D2
Person 6	U1/D2	U2/D1
Person 7	U2/D1	U1/D2
Person 8	U1/D2	U2/D1
Person 9	U2/D1	U1/D2
Person 10	U1/D2	U2/D1

Table 7.1: Each user executes the test tasks twice, with the order of user interfaces (U1, U2) and data subsets (D1, D2) given here.

Experiments were conducted with 10 test users in a controlled environment with the setup as shown in Figure 7.1, the only differences between the two experiments being computer speed and screen size. The user interface was operated with a wireless mouse. User's actions are recorded on audio and video, with the camera capturing the computer screen and user's facial expressions in the mirror. The person conducting the test, who observes and takes notes, sits behind the user to minimize distraction and prevent possible communication attempts.

A day before the actual testing, a pilot test was performed with two additional pilot users trying out the test setup. Pilot users executed the tasks to identify glitches in the test configuration and in the tasks. After the pilot test the test tasks were polished up and the timeout limits were tuned accordingly to the findings, to ensure that the real testing procedure runs as smoothly as possible.

Before testing each user was given a short orientation script providing information about the goals of the test, and was asked to fill out a form providing basic personal information, such as:



Figure 7.1: Usability testing environment.

- Personal information: sex, age and profession
- Vision: wearing glasses, color blindness
- Education and specialization
- Computer usage experience
- Familiarity with visualization tools
- Taking part in usability experiments

After that each test user was given a crash course on the tested KDVE user interface and the main concepts of the employed visualizations, restricted to the functionality relevant to the test. Users were only instructed how to use the relevant functions of the user interface, but not how to execute a particular task or workflow, as this would have tainted the test results. To minimize the possibility of users getting lost in the functionality not relevant to the test, many functions and options were temporarily disabled or completely removed from the user interface.

Once the the introduction to the user interface was completed, users were given several minutes to try it out and familiarize with the interactivity and functionality. The testing would begin when users were reasonably confident that they have understood the visualizations and the user interface, and were able to perform basic operations needed for performing the tests, such as navigation or selection. At this point users were requested to ask all remaining questions they had about the interface and about the test. During the actual testing users were not allowed to ask questions or otherwise communicate with the person conducting the test as this would affect the duration of the tasks. However, users were permitted to make comments while performing tasks but, as the person conducting the test was sitting behind them, direct communication could not be established.

Testing procedure begins by the person conducting the test handing over task descriptions to the test user. During the test, all actions performed by the users are recorded, on video and audio. Time required to execute each task was recorded by the person conducting the test, who was also observing the test users during test execution and was taking additional notes. For each task a time limit If the time required to complete a task was exceeded, a timeout was recorded and the user was instructed to proceed to the next task.

After the test each user was given a feedback form with statements describing selected functions of the visualizations and of the user interface. Users expressed the degree of agreement with the statements on a Likert scale [Likert 1932] with values ranging from 0 to 6. The statements and values were formulated such that, in the given context, a higher value would mean "better" while a lower value would mean "worse", except stated differently. Finally, each user was given the opportunity to orally provide personal impressions, asses the performance of the user interface, and point out its advantages and disadvantages.

7.2 Topical-Temporal Analysis Experiment

In this experiment two different user interfaces for simultaneous, "fused" topical and temporal analysis were compared [Weitlaner 2009]. The user interfaces can be seen in Figure 7.2. The first interfaces (up in the Figure), which includes two visualization components, consists of three coordinated views: a tree (for hierarchy navigation), a Landscape (for analysis of topical similarity) and a StreamView (for temporal analysis). The second interface (down in the figure), uses only one visualization and consist of the following coordinated views: a tree (for hierarchy navigation), a table (for analysis of topical similarity) and a StreamView (for temporal analysis).

The main difference between the two interfaces is that one interface uses the Landscape visualization for analysis of topical similarity (which is conveyed by spatial proximity), while the other one employs a well-known table component. The table displays information on the children of the currently focused cluster and, most importantly, for each shown sub-cluster or document provides a list of siblings sorted by topical similarity. In this way the user can quickly find the most similar siblings and read the exact topical similarities between them, which are given in numerical form. For example, in the Figure 7.2 (lower interface), one can see in the table that for "Cluster 1" the list of sorted siblings looks like "3: 0.14317748; 2: 0.140653; 4: 0.12397176; 7: 0.08564...", which means that "Cluster 3" is its most similar sibling, followed by "Cluster 2", "Cluster 7" and so on.

The idea behind comparing these two interfaces is to discover whether a combination of two visualization, each addressing a different aspect of the data, performs better than a single visualization paired with standard components, such as a table and a tree. While trees and tables are familiar to the user, visualizations pose an additional cognitive load on the user which may lead to reducing performance of a user interface instead of improving it.

Experiments were performed on a group of 10 users, 2 female and 8 male, between 20 and 42 years of age. All test users had a technical background with majority of the being students of technical courses such as mathematics, computer science or telematics. All user had at least a decade of experience using computers, with the typical weekly computer use from 25 to almost 100 hours. Five persons had experience with visual software tools, however only to a small degree. Since the evaluated tools are targeted toward expert users, the technical inclination of the test users appears adequate. Seven persons used glasses for working, but nobody was color blind or had any other visual impairment. Experiments were performed on a Desktop PC using an Intel Pentium D dual core 3.4Ghz CPU, 3GB main memory and Windows XP Professional operating system. The PC was connected to a Fujitsu-Siemens 17" LCD monitor with a resolution of 1280x1024 pixels.

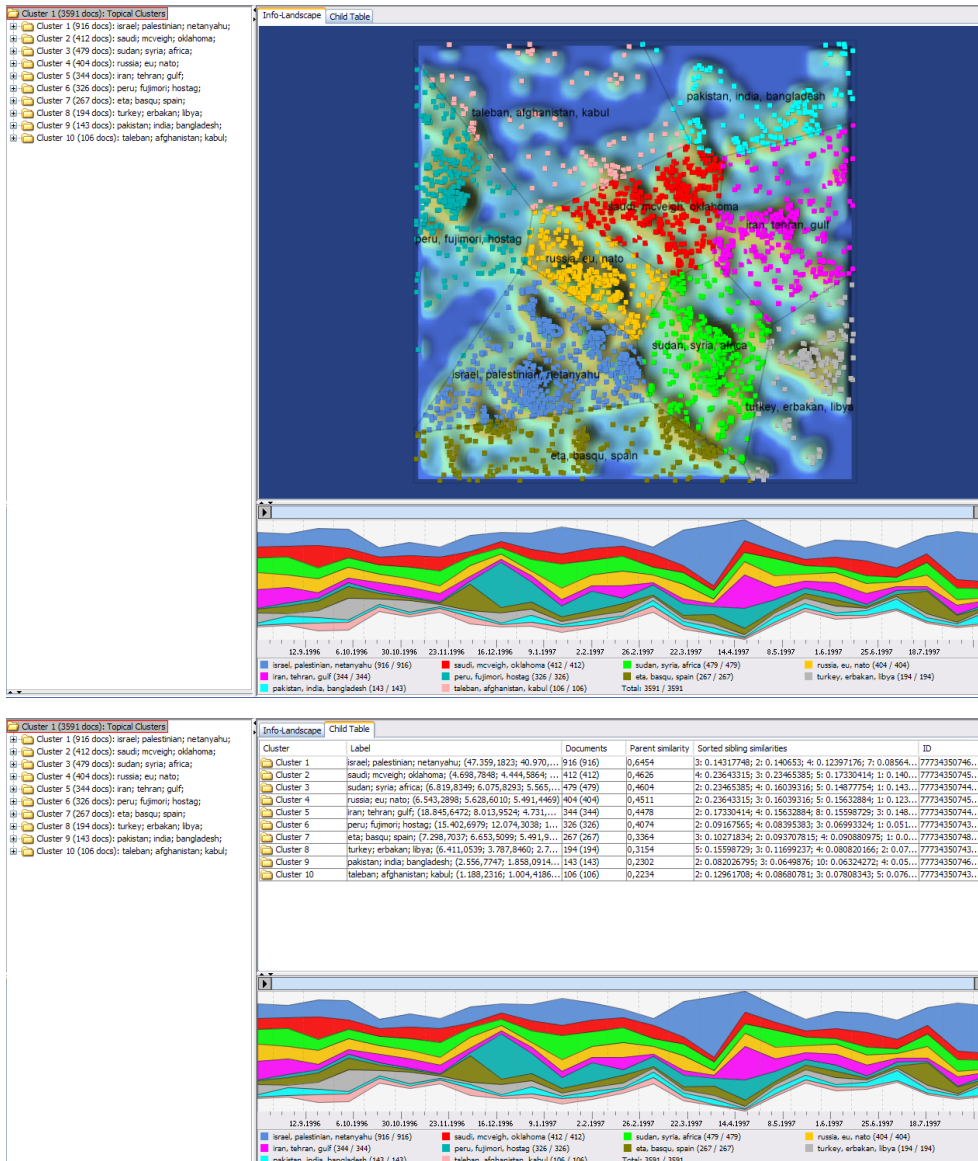


Figure 7.2: Child table showing 10 sub-clusters, with sibling-similarities being shown for each sub-cluster sorted in descending order.

7.2.1 Tasks

Users performed three tasks, beginning with discovering simple temporal patterns and then proceeding to more complex topical-temporal analysis. Tasks were performed twice, using the two user interfaces described above, according to the within-groups methodology. The performed tasks, in ascending level of difficulty, are:

Compare tree-table-StreamView configuration vs. tree-Landscape-StreamView. In the no-landscape configuration, sibling similarities were listed in the siblings-table sorted in descending order. Recursive StreamView was on per default.

- Task 1 (difficulty level - easy, timeout 90s): Identify two top-level topical clusters with temporal peaks and find the time interval in which they occurred, and find one cluster with mostly uniform development
- Task 2 (difficulty level - medium, timeout 180s): Starting from a temporal peak identified in the previous step, identify a sub-cluster which is mostly responsible for the peak, and find one sibling that temporally correlates with the peak and one that does not.
- Task 3 (difficulty level - hard, timeout 240s): Identify the main temporal peak for a given topical cluster, find out when it occurs and identify a sub-cluster which is mostly responsible for the peak (i.e. just as in the previous task). Now find a second sub-cluster which is topically related AND temporally correlates with the first sub-cluster, and find a third sub-cluster which is topically related but temporally does NOT correlate with the first sub-cluster.

	Task 1	Task 2	Task 3
Person 1	122	62	190
Person 2	56	117	113
Person 3	89	103	134
Person 4	I	85	70
Person 5	T	50	I
Person 6	45	84	I
Person 7	131	93	131
Person 8	101	T	125
Person 9	47	60	66
Person 10	T	50	136
Average	84.4	78.2	120.6

Table 7.2: Task execution times, in seconds, with a user interface employing and information landscape for analysis of topical relatedness.

	Task 1	Task 2	Task 3
Person 1	94	75	149
Person 2	126	114	T
Person 3	43	90	111
Person 4	T	105	I
Person 5	118	64	155
Person 6	101	45	65
Person 7	76	77	103
Person 8	96	85	217
Person 9	89	42	71
Person 10	101	130	177
Average	93.8	82.7	131

Table 7.3: Task execution times, in seconds, with a user interface employing a child table with sorted sibling lists for analysis of topical relatedness.

7.2.2 Task Execution Times

Table 7.2 shows task execution times for the user interface employing an information landscape and a StreamView for topical-temporal analysis. Table 7.3 shows task execution times for the user interface employing a table with sorted sibling lists instead of the Landscape. When a user could not complete a task within the maximum intended time, the task was interrupted and a timeout (T) was recorded. If a user completed a task in time but provided a wrong result the outcome is marked as incorrect (I). Difference between the average task execution times using the two interfaces is shown in Table 7.4.

	Task 1	Task 2	Task 3
User interface with the Landscape	84.4	78.2	120.6
User interface with the table	93.8	82.7	131
Improvement using Landscape over table	9.4	4.5	10.4
Improvement as percentage	10.02%	5.44%	7.94%

Table 7.4: Improvements in average task execution times achieved by using an interface with a Landscape and StreamView compared to an interface using a table and a StreamView.

7.2.3 User Feedback

After completing all tasks users filled out a feedback form providing subjective assessments on:

1. Usefulness and intuitivity of the StreamView temporal visualization.
2. How well each of the two different user interfaces performed.
3. General impressions on the user interface which employs both visualizations (Landscape and StreamView).

Answers were delivered on a Likert scale from 0 to 6.

7.2.3.1 Feedback on StreamView

Statements used for collecting feedback on the StreamView temporal visualization component are shown in Table 7.5, answers delivered by users are available in Table 7.6.

1.	Identification of major events is	easy	6 .. 0	hard
2.	Selection of a time interval is	easy	6 .. 0	hard
3.	Intuitivity of the temporal visualization is	good	6 .. 0	bad
4.	Intuitivity of recursive temporal visualization is	good	6 .. 0	bad

Table 7.5: Statements used to collect user feedback on the StremView temporal visualization.

	1.	2.	3.	4.
Person 1	6	5	6	5
Person 2	5	6	5	4
Person 3	5	4	6	4
Person 4	6	6	5	4
Person 5	5	5	3	4
Person 6	5	4	6	5
Person 7	5	6	6	4
Person 8	6	5	5	4
Person 9	5	6	6	4
Person 10	6	6	5	6
Average	5.4	5.3	5.3	4.4

Table 7.6: Results of the user feedback on the StremView temporal visualization.

7.2.3.2 Feedback on the User Interfaces

Subjective feedback on the Landscape and its combination with the StreamView for performing topical-temporal analysis was collected using a feedback form as seen in Table 7.7, with results shown in Table 7.8.

7.2.3.3 General Impressions

General statements on the visual application for topical-temporal analysis were collected using a feedback form as seen in Table 7.9, with results available in Table 7.10.

1a.	Finding a cluster with interface using the table	easy	6 .. 0	hard
1b.	Finding a cluster with interface using the Landscape	easy	6 .. 0	hard
2a.	Topical-temporal analysis using StreamView and table	easy	6 .. 0	hard
2b.	Topical-temporal analysis using StreamView and Landscape	easy	6 .. 0	hard

Table 7.7: Statements used to collect user feedback about the two different user interfaces for topical-temporal analysis.

	1a.	1b.	2a.	2b.
Person 1	4	5	6	6
Person 2	5	4	5	4
Person 3	4	5	5	5
Person 4	5	5	5	5
Person 5	4	5	3	4
Person 6	6	6	5	6
Person 7	3	6	1	5
Person 8	2	4	2	5
Person 9	4	5	4	5
Person 10	6	6	6	6
Average	4.3	5.1	4.2	5.1

Table 7.8: Results of user feedback collected on the two different user interfaces for topical-temporal analysis.

7.2.4 Results Discussion

This experiment, which is of particular importance to this work, yielded very satisfactory results. By comparing execution times of the two compared user interfaces, it is clear that the interface using both visualization (Landscape and StreamView) achieved slightly better results than the interfaces which uses table instead of the Landscape. Although the difference is small (5-10%) the result demonstrates that the increased cognitive load introduced by the two visualizations is not excessively high, and that users can very well cope with such a complex visual interface. Considering that users were familiar with the table, but used an information landscape for the first time, provides additional weight to this result. These findings are also supported by the user feedback, where users clearly stated that the topical-temporal analysis was easier to perform using the interface employing a Landscape and a StreamView.

1.	Graphical design of the application	good	6 .. 0	bad
2.	General impression of the application	good	6 .. 0	bad
3.	Would you use the application professionally	yes	6 .. 0	no

Table 7.9: Statements used to collect general user feedback about the visual application for topical-temporal analysis using both visualizations (Landscape and StreamView).

	1.	2.	3.
Person 1	6	6	4
Person 2	5	5	4
Person 3	5	3	4
Person 4	5	5	4
Person 5	3	4	4
Person 6	6	6	0
Person 7	6	5	3
Person 8	6	1	0
Person 9	5	5	4
Person 10	5	5	3
Average	5.2	4.5	3

Table 7.10: Results of general user feedback on the visual application for topical-temporal analysis using both visualizations (Landscape and StreamView).

Users gave very positive feedback on the usefulness and intuitivity of the StreamView temporal visualization. One aspect of how the StreamView is used in the coordinated user interface did cause some confusion: the recursive StreamView recomputation. When the user starts using the interface, StreamViews displays temporal developments of the top-level clusters. When navigating deeper in the hierarchy, the StreamView will recompute the visualization to show temporal developments of the currently chosen cluster’s children (sub-clusters). This is not something that each user immediately understood and a few users were confused by the behavior. Another problem was, that in situations when users applied temporal selection in the StreamView, but then navigated to a neighboring or a child cluster in the Landscape, the temporal selection would suddenly pick the documents from that other cluster, which was unintended by the user. To ameliorate the situation, a button for

”freezing” the StreamView on a selected set of clusters, independently of the navigation in the hierarchy, was introduced in the following version of the user interface.

Finally, users provided very favorable feedback on the graphic design of the application and also had a solid general impression about it. However, users were in average undecided on whether they would use the application professionally. When asked about this, users declared that several small but annoying bugs and interactivity glitches reduced the satisfaction of use and made some operations tedious. The application did not appear polished enough for productive use. As a consequence, both reported and observed glitches were fixed in the next version of the application.

7.3 Explorative Analysis with the Information Landscape

For evaluating the explorative analysis performance of the information landscape in the context of a multiple coordinated view user interface, two separate sub-experiments with separate set of test tasks were conceived [Krnjic 2008]. The first one addresses navigation in the hierarchically organized cluster hierarchy visualized as nested areas (see Section 4.3.1), while the second one focuses on visual property coding of metadata and features (see Section 4.3.2). The tests were performed on a data set containing approximately 10000 news articles from the Reuters corpus [RCV1 2000] using a GUI configuration consisting of a tree, information landscape and a table, as seen in Figure 7.3. Experiments were performed on a HP Pavilion dv2000 Widescreen 14.1” (1280x800) laptop with an AMD Turion dual core 2GHz processor, 4GB main memory and Windows Vista Home Premium operating system.

Experiments were performed on a group of 10 users, 2 female and 8 male, between 22 and 35 years of age. All test users had a technical background with typical computer use from 30 to 70 hours a week, or 47,5 hours per week in average. Four persons had some experience with visual software tools. Since the evaluated tools are targeted toward expert users, the technical inclination of the test users appears adequate. Two persons used glasses for work and no user was color blind or had any other visual impairment.

7.3.1 Navigation Experiment

This experiment evaluated explorative navigation and browsing of the topical hierarchy using the information landscape. The goal was to discover whether

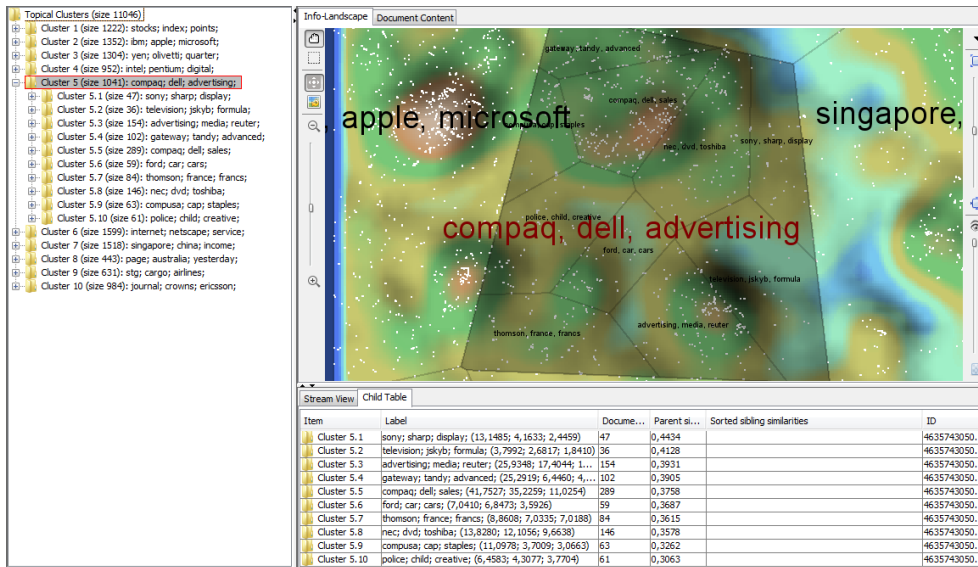


Figure 7.3: User interface configuration used for the two explorative analysis experiments.

the automatic cluster area focusing provides advantages compared to completely manual zooming and panning. When automatic cluster area focusing is disabled the user must set the focus on the chosen cluster by manually zooming (using mouse wheel) and panning (dragging the mouse). With the automatic focusing enabled, a user clicking on a cluster label will trigger a smooth animated transition, lasting about a second, in which the landscape is scrolled and zoomed in such a way that the cluster area appears centered and occupies the screen area available to the Landscape (see Figure 4.5). Note that manual zooming and scrolling are still available, allowing users to adjust the view should they find it necessary.

7.3.1.1 Tasks

Each user performed three different task on two different data subsets, whereby the order of data subsets and the order of automatic vs. manual navigation was interchanged for different users (as described in Section 7.1). The three tasks performed by the test users, in ascending level of difficulty, are:

- Task 1 (difficulty level - motivational, timeout 120s): Given is a highlighted document and its parent cluster. Two other visible cluster are specified by name (labels). Decide which of the two other clusters is more similar to the document.

	Task 1	Task 2	Task 3
Person 1	44	100	124
Person 2	102	153	94
Person 3	T	233	60
Person 4	61	125	224
Person 5	32	T	140
Person 6	73	119	45
Person 7	98	T	77
Person 8	36	103	75
Person 9	83	222	145
Person 10	114	T	T
Average	71.44	150.71	109.33

Table 7.11: Task execution times, in seconds, with the automatic cluster focusing.

- Task 2 (difficulty level - easy, timeout 240s): Given a highlighted document of interest, identify 5 similar documents and find out which direct parent cluster these documents belong to.
- Task 3 (difficulty level - medium, timeout 180s): Given is the same document as in the previous task. Find all neighboring clusters to this document's parent cluster.

7.3.1.2 Task Execution Times

Table 7.11 shows execution times for automatic cluster area focusing turned on, Table 7.12 shows execution times for manual navigation. When a user could not complete a task within the maximum intended time, the task was interrupted and a timeout (T) was recorded. Improvement or deterioration in average task execution times achieved by using automatic cluster area focusing compared to manual navigation is shown in Table 7.13.

	Task 1	Task 2	Task 3
Person 1	67	166	44
Person 2	45	149	53
Person 3	T	117	37
Person 4	T	186	108
Person 5	80	107	45
Person 6	T	203	131
Person 7	T	195	137
Person 8	95	205	T
Person 9	68	104	90
Person 10	122	T	90
Average	79.5	159.11	81.67

Table 7.12: Task execution times, in seconds, using manual zooming and panning.

	Task 1	Task 2	Task 3
Automatic focusing average	71.44	150.71	109.33
Manual navigation average	79.5	159.11	81.67
Improvement automatic over manual	8.06	8.4	-27.66
Improvement percentage	10.13%	5.28%	-33.87%

Table 7.13: Improvements and deterioration in average task execution times achieved by automatic focusing compared to manual navigation.

7.3.1.3 User Feedback

After completing all tasks the users were given a feedback form, to provide subjective assessments on how easy or difficult it was to perform operations, and how useful a particular function is. Answers were delivered on a Likert scale from 0 to 6. Statements used for collecting feedback are shown in Table 7.14, answers delivered by test users can be found in Table 7.15.

1a.	Finding a document using auto-focus is	easy	6 .. 0	hard
1b.	Finding a document with manual navigation is	easy	6 .. 0	hard
2a.	Finding a cluster using auto-focus is	easy	6 .. 0	hard
2b.	Finding a cluster with manual navigation is	easy	6 .. 0	hard
3.	The automatic focusing is helpful	yes	6 .. 0	no

Table 7.14: Statements used to collect user feedback after performing the navigation experiment.

	1a.	1b.	2a.	2b.	3.
Person 1	5	5	5	3	5
Person 2	5	5	1	6	1
Person 3	6	6	5	3	5
Person 4	5	5	4	5	5
Person 5	5	5	4	5	3
Person 6	4	4	5	5	4
Person 7	3	1	3	1	0
Person 8	6	5	6	5	6
Person 9	5	2	5	1	5
Person 10	6	1	6	4	3
Average	5	3.9	4.4	3.8	3.7

Table 7.15: Results of user feedback collected after performing the navigation experiment.

7.3.1.4 Results Discussion

Time measurements show that for the first two tasks automatic focusing was slightly better than manual navigation, but in the third, most complex task automatic focusing performed much worse (Table 7.13). Number of timeouts was 5 for automatic focusing and only slightly larger - 6 - for manual navigation. These results are inconclusive and do not indicate a clear winner between

the two navigation strategies. User feedback results (Table 7.15) indicate that users still preferred the automatic focusing over manual navigation.

From observations and from users comments during the test, the performance hit in the third task was due to users losing orientation and getting confused. When zoom factor is large, by clicking on a neighboring label or document the automatic focusing will navigate "too far away" if that label or document belong to the neighboring cluster (and not to the one the user is focusing on). This sudden, large change of focus is unexpected for users and they would need time to orientate themselves again. Users expressed the need for turning the automatic focusing off, for example in situations when they wanted to make smaller adjustments to their current position in the landscape. Therefore, a button for switching automatic focusing on and off was built into the next version of the user interface.

7.3.2 Visual Properties Coding Experiment

Mapping of document properties and metadata to visual properties such as colors and shapes enables users to discover correlations between topical clusters and selected features and metadata (see Section 4.3.2 and Figure 4.6). This goal of this experiment was to discover which type of icons is more suitable for this purpose: overlaid colored shapes or a single shape - a disk - composed of different colors. Icons used for testing can be seen in Figure 7.4, where on the left side colored disks can be seen and on the right overlaid symbols (plus, circle and cross) in different colors are shown.



Figure 7.4: Icons used for evaluation: disk icons in different colors (on left), icons with different shapes and different colors (on right).

In the experiment location metadata was correlated with topical clusters. Three cities were each assigned a different color (London - red, New York - blue and Tokyo - yellow). When using a disk icons, documents mentioning more than one city are shown as disks composed of more than one color. When using colored symbols, documents mentioning multiple cities are shown as

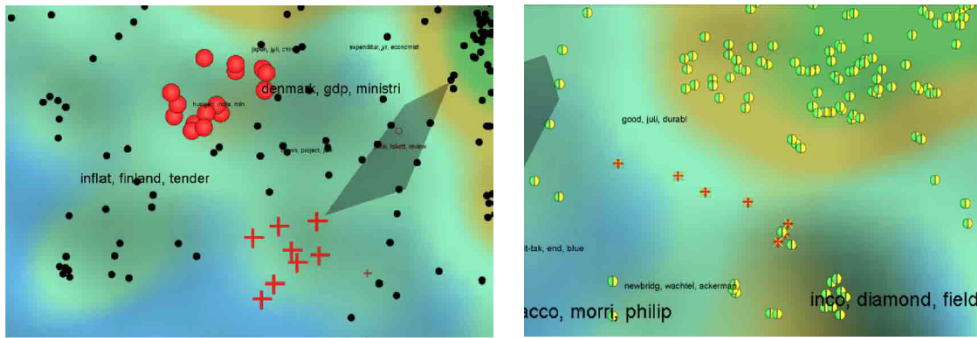


Figure 7.5: Examples of using colored disks and overlaid colored symbols to convey properties, with the left image showing a single property and the right showing two properties mapped [Krnjic 2008].

icons with different symbols overlaid over each other. Documents which did not mention any of the three cities were shown as a small black dot. Examples of how icons appear in the Landscape can be seen in Figure 7.5

7.3.2.1 Tasks

Users performed two simple tasks in which they estimated the amounts of items with specific properties within different clusters. Task were performed twice, using colored disks and then using overlaid colored symbols, according to the within-groups methodology. The performed tasks, in ascending level of difficulty, are:

- Task 1, difficulty level - easy: Find the topical cluster and a sub-cluster within that cluster, where a specified city is mentioned most often, while other two cities are mentioned as rarely as possible.
- Task 2, difficulty level - medium: Find two topical clusters where all three specified cities are mentioned often, and identify one topical cluster which contains documents mentioning two or more cities.

	Task 1	Task 2
Person 1	56	88
Person 2	52	77
Person 3	93	91
Person 4	70	138
Person 5	T	115
Person 6	77	149
Person 7	187	80
Person 8	69	78
Person 9	72	84
Person 10	54	33
Average	81.1	93.3

Table 7.16: Task execution times, in seconds, using colored disk icons.

	Task 1	Task 2
Person 1	44	134
Person 2	61	142
Person 3	114	93
Person 4	81	197
Person 5	81	224
Person 6	114	241
Person 7	157	125
Person 8	81	103
Person 9	52	75
Person 10	67	58
Average	85.2	139.2

Table 7.17: Task execution times, in seconds, using overlaid colored symbols icons.

	Task 1	Task 2
Colored disk icons	81.1	93.3
Overlaid colored symbol icons	85.2	139.2
Difference colored disks vs. overlaid symbols	4.1	45.9
Improvement disks vs. symbols as percentage	4.81%	32.97%

Table 7.18: Differences in average execution time between colored disks and overlaid colored symbols.

7.3.2.2 Task Execution Times

Table 7.16 shows execution times using colored disk icons, Table 7.17 shows execution times for icons composed of overlaid colored symbols. When a user could not complete a task within the maximum time intended time, executing of the task was interrupted and a timeout (T) is recorded. Difference between average task execution times between colored disks and overlaid colored symbols are shown in Table 7.18.

7.3.2.3 User Feedback

After completing all tasks users filled out a feedback form, to provide subjective assessments on how well colored disk icons and overlaid colored symbol icons performed. Answers were delivered on a Likert scale from 0 to 6. Statements used for collecting feedback are shown in Table 7.14, answers delivered by test users can be found in Table 7.15.

1a.	Property coding using colored disks	good	6 .. 0	bad
1b.	Property coding with overlaid colored symbols	good	6 .. 0	bad

Table 7.19: Statements used to collect user feedback after performing the the visual property coding experiment.

	1a.	1b.
Person 1	5	2
Person 2	6	5
Person 3	4	4
Person 4	6	4
Person 5	4	5
Person 6	5	2
Person 7	5	2
Person 8	5	4
Person 9	6	0
Person 10	5	2
Average	5.1	3

Table 7.20: Results of user feedback collected after performing the visual property coding experiment.

7.3.2.4 Results Discussion

The result of this experiment is clear: For visualizing document metadata and properties, using a colored disk icons is superior to using overlaid colored shapes. Objective time measurements as well as user feedback confirm this result. Although one might expect that using both colors and symbols to differentiate properties would produce better results than using just one shape in different colors, this was not the case. The reason stated by the user was, that colored disks had a stronger, clearer visibility. This was especially the case in overview of the whole data set when individual icons are small. Also, when using overlaid colored shapes some users got confused, stating that they would expect colors and shapes, which are different visual channels, to encode different types of properties (which was not the case in the experiment).

1.	Graphical design of the application	good	6 .. 0	bad
2.	The navigation is in general intuitive	yes	6 .. 0	no
3.	General impression of the application	good	6 .. 0	bad
4.	Would you use the application professionally	yes	6 .. 0	no

Table 7.21: Statements used to collect general user feedback about the visual application after performing both explorative analysis experiments.

	1.	2.	3.	4.
Person 1	5	6	5	4
Person 2	6	6	5	5
Person 3	5	5	5	4
Person 4	6	5	6	5
Person 5	4	5	4	5
Person 6	6	5	5	4
Person 7	4	2	3	3
Person 8	5	4	5	6
Person 9	2	1	2	3
Person 10	2	5	3	1
Average	4.5	4.4	4.3	4

Table 7.22: Results of general user feedback on the visual application used for explorative analysis experiments.

7.3.3 General Impressions

Finally, some general feedback about the graphical design and interactivity was collected from the users. Again, Answers were delivered on a Likert scale from 0 to 6. Statements are shown in Table 7.21, answers delivered by the users are given in Table 7.15. There results are in the positive part of the scale, where no particular highlights or weaknesses could be identified. Users were reasonably satisfied with the graphical design and navigation possibilities, and had a solid, but not excellent, general impression of the application, stating that they would use it professionally.

7.4 Summary

From the results of the performed experiments it can be said that the developed technologies proved useful in fulfilling their purposes. Objective measurements and subjective user feedback, presented in this Chapter and in Section 3.3.6, confirm that using multiple-visualization user interfaces for explorative topical and temporal analysis, provided tangible advantages compared to interfaces using a single visualization or no visualization at all. Valuable information was obtained on explorative navigation in hierarchically organized information landscapes and on how document features and metadata should be represented. These results, together with many smaller findings collected during the evaluations, such as for example preferred label size and coloring, animation duration, or detection of several minor (but annoying) interactivity glitches, were used to improve the visual components and the KDVE user interface. Graphical and interactivity improvements implemented after the tests resulted in visual components and a user interface which, with high probability, perform better and provide a better overall experience than what was used for the experiments. Successful application of the developed visual technologies in productive environments provides additional strength to this claim.

Chapter 8

Conclusion

This work addressed the problem of analysing and understanding large, heterogeneous, dynamically changing repositories through application of visual analytics methods. Visual methods were combined and integrated with automatic techniques to develop usable methods and tools for explorative analysis of such complex data sets.

After providing motivation for the use of visual analytics methods in general, a survey of the scientific fields relevant to this work was given. Presentation of my results begins with earlier research I have conducted on the use of visualization for analysis of large, dynamic data sets. Building upon and combining these results, and extending them with scalable, incremental algorithms and new interactive visual components, culminated in the design of two prototype applications. The first (and main) prototype targets the analysis of dynamically changing, metadata-enriched text document repositories, while the second one addresses semantic knowledge bases. After describing implementation details of the algorithmic and visualization components, applications of the developed technologies and prototypes were demonstrated and discussed. The document concludes with the evaluation of selected technologies and components.

8.1 Result Summary

The main contribution of this work lies in development, evaluation and application of visual methods which, tightly integrated with automatic techniques, provide analytical means for exploring, analyzing and understanding large, dynamically changing, metadata-rich data sets. Developed visual techniques were applied primarily on text documents to reveal patterns and correlations in and between topical information, temporal information and metadata. Ap-

plication of the developed methods on semantic data demonstrates the wider applicability of the developed methods. The target audience of all resulting methods, tools and applications are expert users, typically analysts or knowledge engineers.

In particular, to achieve the goals listed in Section 1.2.2 this work delivers following important results:

- Incremental, scalable ordination algorithm, which is suitable for visualization of relatedness in large, dynamically changing data sets. The algorithm computes a hierarchical structure suitable for navigation and a corresponding geometry used by interactive visual components.
- Scalable visualization components for analysis of relatedness and dynamics in large, dynamic data sets, in particular:
 - Information landscape provides an overview of the data and visually conveys information on relatedness, size and cohesion of structures arising from the data. Explorative navigation is supported through a hierarchy of nested, labeled polygonal areas providing means for level of detail-sensitive orientation. Discovery of correlations between relatedness-based structures and selected metadata and features is supported through color and icon coding of data elements. Additionally, a dynamic topography landscape supports the understanding of changes in relatedness-based structures in dynamic data sets.
 - A StreamView visualization component provides support for discovery and understanding of temporal patterns in the data.
- Scalable view coordination mechanisms address coordination of visual and logical data item properties over multiple visualization components, and provide synchronized navigation in the data set. The coordination framework enables creation of complex user interfaces composed of multiple visual components for simultaneous analysis of different data aspects (for example topical-temporal analysis using Landscape and StreamView components).
- Prototypical user interfaces, based on multiple coordinated views, are available for testing and evaluating visual techniques. The applications address:
 - Fused analysis of topical relatedness, temporal developments and metadata distribution in large, dynamically changing text repositories.

- Visually supported, semi-automatic alignment of ontologies.

The first application is the primary demonstrator showing the usefulness of the developed technologies for achieving the defined goals. The second application demonstrates the flexibility and applicability of selected techniques on an alternative domain and data type.

- Tight integration of visual methods and automated processing as an enabler for several analytical techniques, tasks and workflows, in particular:
 - Combining clustering and ordination techniques to provide hierarchical navigation capability in similarity layout-based visualizations.
 - Using information extraction to produce faceted metadata categories, which are used to visualize distribution of metadata over topical clusters and over time.
 - Using high-performance retrieval techniques for filtering and highlighting in large scale visualization.
 - Supporting creation of and periodical tuning of classifiers using incremental visualization techniques.
- Usability evaluation providing information on the performance of coordinated visual user interfaces:
 - Measurements and user feedback confirm that a multiple-visualization, coordinated user interface for explorative topical-temporal analysis, consisting of Landscape and StreamView components, provides advantages compared to an interface where one visualization is replaced by a standard GUI widget (such as a table).
 - Valuable information was obtained on user preferences for explorative navigation in hierarchically organized information landscapes, and on visual representation of document features and metadata for discovery of correlations between topical clusters and metadata distribution.
- Successful integration of developed technologies in productive systems and their application in real-world scenarios.

To conclude, the research questions defined in Section 1.2.2 can be answered as follows:

1. How can visual analytics techniques, i.e. visual methods combined with automatic processing, be used to achieve the defined goals?

Answer: An incremental, scalable ordination algorithm is used to compute a hierarchically organized geometry of the data set which organizes the data depending on relatedness. A StreamView component provides visual analysis of temporal developments, while an information Landscape is used for conveying relatedness in the data set and visualizing metadata distribution. For supporting analytical tasks integration of additional automatic methods is provided, such as information extraction for extracting metadata from text or retrieval techniques for fast filtering in large data sets.

2. Does the integration of multiple visualization components into a single interactive user interface provide an effective instrument for simultaneously addressing all goals?

Answer: A coordinated multiple view framework is used for fusing the visualization components into one complex, coherent user interface for simultaneous analysis of multiple data aspects. Usability evaluation results show that a user interface for temporal-topical analysis consisting of two visualizations (Landscape and StreamView) did not cause cognitive overload of the user, and that it performed better than a user interface where one visualization (Landscape) was replaced by a table providing data in numerical form.

3. Can the developed techniques be extended and applied on more than one data type and more than one type of relatedness?

Answer: To demonstrate the flexibility and applicability of selected algorithms and visual techniques (i.e. the ordination algorithm, Landscape visualization and coordination framework) were applied on semantic knowledge bases, resulting in a tool for semi-automatic, visually supported ontology alignment. Instead of addressing topical relatedness in text data, analytical methods are applied on semantic relatedness between concepts from different ontologies.

8.2 Future Work

Although many questions have been answered and many problems were successfully solved along the way, new challenges never cease to appear and ideas do not run dry. My future mission includes practical problems which definitely should be addressed in the next future, as well as some more visionary, adventurous ideas which are also more tempting.

The first category is mainly limited to the following two points:

- Direct comparison of the ordination algorithm with other methods. This task is facing several practical problems, including but not limited to: To my knowledge no other freely available technique is capable to handle extreme high-dimensionality (as present in text data), scale to large data sets, provide incrementality and deliver visually appealing layouts. Also, for visualization purposes a development of a quality measure which, in addition to goodness of fit, also considers visual and aesthetical aspects would have to be developed. The standard stress measure, which only considers goodness of fit, may not be ideal from the usability point of view, because the extreme differences in distances occurring in very high-dimensional data sets, if reproduced faithfully in the visualization space, would not yield usable and visually pleasing visual representations.
- Usability evaluation of the Semantic Evaluation Tool. Although the information landscape representation has been the subject of numerous usability evaluations, its combination with graph visualization methods and application in ontology alignment scenarios should be the subject of a more into depth examination.

The list of things which I would personally like to address in the future includes the following:

- Adding the capability to explicitly visualize relationships in the information landscape, effectively making it a scalable, dynamic-topology graph visualization. This could be achieved along the lines of ideas presented in [Kandlhofer 2009], [Sabol et al. 2010a], [Sabol et al. 2010b] and [Kienreich & Seifert 2010].
- Consideration of dynamic, evolving ontologies in the Semantic Mediation Tool using StreamView visualization and the above mentioned scalable, dynamic-topology graph visualization.
- Adding infinite zoom-like capability to the information landscape, which would allow the user to zoom in into the visualized items displaying their internal structure and content.
- Add a geovisualization component to the KDVE coordinated user interface to provide a fused topical-temporal-geospatial analysis capability. As adding more different visual components to the user interface increases the cognitive load on the user, performing usability evaluation would be necessary to assess whether such an interface is useful or overloaded.

- Exploration of visual user feedback mechanisms and how these should affect the data model backing the visual representation.
- Real-time collaboration using multitouch tables, projection walls and mobile devices.

Appendix A

Authored and Co-authored Publications

This appendix briefly introduces the author and lists all authored and co-authored publications (as of August 2011).

A.1 About the Author



I have completed the study of Telematik in 2001 at the Technical University of Graz, Institute for Information Systems and Computer media (IICM), receiving the degree of Dipl.-Ing. I received the prize for the best thesis completed at IICM in 2001. As of August 2011 I am a senior researcher at the Know-Center and a deputy division manager Knowledge Relationship Discovery division, where I have been employed since 2001.

My interests are in the fields of visual analytics, information and knowledge visualization, human-computer interaction, and knowledge discovery. My research is focused on visual analysis of dynamics and relationships in large, heterogeneous data sets, which is also the topic of this work. I was involved, either as a collaborator or as a project leader, in various research projects addressing the aforementioned areas. These include, but are not limited to:

- InfoSky - Visualization of Large Hierarchical Document Spaces [Andrews et al. 2002], 2001 - 2004.
- WebRat - Web-Based Retrieval, Clustering, and Visualization Framework [Sabol et al. 2002a], 2002.
- KnowMiner - Know-center's Knowledge Discovery Framework [KnowMiner 2011], 2003 - ongoing, project lead 2003 - 2005.
- OnAir - APA Intelligent Retrieval [Kienreich et al. 2005a], 2005.
- MISTRAL - Measurable Intelligent and Reliable Semantic Extraction and Retrieval of Multimedia Data [MISTRAL 2005], 2005 - 2006.
- SCI - Semantic Competitive Intelligence (Visually supported analysis of semantic patterns in knowledge repositories) [Sabol et al. 2009a], 2007 - ongoing, project lead at the Know-Center.
- RAVEN - Relation Analysis and Visualization for Evolving Network [RAVEN 2008], 2008 - 2009, project lead at the Know-Center.
- SMT - Visual Semantic Mediation Tool, 2009 - ongoing, project lead at the Know-Center.
- DIVINE - Dynamic Integration and Visualization of Information from Multiple Evidence Sources [DIVINE 2011], 2011 - ongoing.

I have served as member of the program committee at several conferences and workshops. I was the co-chair of the several special tracks at the I-Know conference series, involving topics such as Information and Knowledge Visualization, Knowledge Discovery, and in 2011, Visual Analytics.

A.2 Publication List

The following list includes 46 publications including journal contributions, book chapters, conference papers, workshop publications, posters, demos and others. 38 publications out of 46 were peer-reviewed or refereed. Publications are grouped by type and sorted by date in descending order, with references linking to the same publication in the Bibliography.

Journal Contributions:

1. [Granitzer et al. 2010] Granitzer, M., Sabol, V., Onn, K.W., Lukose, D. and Tochtermann, K., Ontology Alignment A Survey with Focus on Visually Supported Semi-Automatic Techniques, Future Internet, Volume 2, Issue 3, 238-258, MDPI AG, 2010.

2. [Sabol et al. 2008a] Sabol, V., Andrews, K., Kienreich, W., Granitzer, M., Text mapping: Visualising Unstructured, Structured, and Time-Based Text Collections, *Intelligent Decision Technologies*, Vol 2, No. 2, IOS Press, 2008, pages 117 - 128.
3. [Andrews et al. 2002] Andrews, K., Kienreich, W., Sabol, V., Becker, J., Kappe, F., Droschl, G., Granitzer, M., Auer, P., Tochtermann, K., *The InfoSky Visual Explorer: Exploiting Hierarchical Structure and Document Similarities*, Palgrave Journals: Information Visualization, London, England, 2002.

Book Chapters:

1. [Sabol et al. 2008b] Sabol, V., Kienreich, W., Granitzer, M., *Visualisation Techniques for Analysis and Exploration of Multimedia Data*, Granitzer, M., Lux, Spaniol : *Multimedia Semantics The Role of Metadata*, ISBN: 978-3-540-77472-3, 219-238, Springer-Verlag, 2008.

Conference Proceedings:

1. [Bertschi et al. 2011] Bertschi, S., Bresciani, S., Crawford, T., Goebel, R., Kienreich, W., Lindner, M., Sabol, V., Vande Moere, A., *What is Knowledge Visualization? Opinions on Current and Future State*, in *Proceedings of the 15th International Conference Information Visualization (IV'11)*, 2011.
2. [Sabol et al. 2010b] Sabol, V., Syed, K.A.A., Scharl, A., Muhr, M., Hubmann-Haidvogel, A., *Incremental Computation of Information Landscapes for Dynamic Web Interfaces*, *Proceedings of the 10th Brazilian Symposium on Human Factors in Computer Systems*, 205-208, 2010.
3. [Muhr et al. 2010] Muhr, M., Sabol, V., Granitzer, M., *Scalable Recursive Top-Down Hierarchical Clustering Approach with implicit Model Selection for Textual Data Sets*, *IEEE Computer Society: 7th International Workshop on Text-based Information Retrieval in Proceedings of 21th International Conference on Database and Expert Systems Applications (DEXA 10)*, 2010.
4. [Seifert et al. 2010a] Seifert, C., Sabol, V., Kienreich, W., *Stress Maps: Analysing Local Phenomena in Dimensionality Reduction Based Visualizations*, *European Symposium Visual Analytics Science and Technology (EuroVAST)*, 2010.

5. [Seifert et al. 2010b] Seifert, C., Sabol, V., Granitzer, M., Classifier Hypothesis Generation Using Visual Analysis Methods, NDT: Networked Digital Technologies, 98-111, 2010.
6. [Sabol et al. 2009a] Sabol, V., Kienreich, W., Muhr, M., Klieber, W., Granitzer, M., Visual Knowledge Discovery in Dynamic Enterprise Text Repositories, Proceedings of the 13th International Conference on Information Visualisation (IV09), IEEE Computer Society, 2009.
7. [Klieber et al. 2009a] Klieber, W., Sabol, V., Muhr, M., Kern, R., Öttl, G., Granitzer, M., Knowledge Discovery Using the KnowMiner Framework, IADIS International Conference Information Systems 2009, 307-314.
8. [Klieber et al. 2009b] Klieber, W., Sabol, V., Muhr, M., Granitzer, M., Using Ontologies For Software Documentation, Malaysian Joint Conference on Artificial Intelligence, 2009.
9. [Granitzer et al. 2009] Granitzer, M., Augustin, A., Kienreich, W., Sabol, V., Taxonomy Extraction from German Encyclopedic Texts, In Proceedings of the Malaysian Joint Conference on Artificial Intelligence 2009, Kuala Lumpur, Malaysia
10. [Sabol et al. 2008c] Sabol, V., Scharl, A., Visualizing Temporal-Semantic Relations in Dynamic Information Landscapes, GeoVisualization of Dynamics, Movement and Change Workshop at the AGILE 2008 Conference, Spain.
11. [Sabol et al. 2007] Sabol, V., Granitzer, M., Kienreich, W., Fused Exploration of Temporal Developments and Topical Relationships in Heterogeneous Data Sets, 3rd International Symposium of Knowledge and Argument Visualization. Proceedings of IV07, 11th International Conference Information Visualisation, IEEE Computer Society, London, UK, 2007.
12. [Kienreich et al. 2007] Kienreich, W., Zechner, M., Sabol, V., Comprehensive Astronomical Visualization for a Multimedia Encyclopedia, 3rd International Symposium of Knowledge and Argument Visualization. Proceedings of IV07, 11th International Conference Information Visualisation, IEEE Computer Society, London, UK, 2007.
13. [Klieber et al. 2006] Klieber, W., Sabol, V., Granitzer, M., Kienreich, W., Kern, R., KnowMiner - Ein Service orientiertes Knowledge Discovery Framework, GI-Edition 2006, Bonner Kllen Verlag, 2006.

14. [Kienreich 2006] Kienreich, W., Granitzer, M., Sabol, V., Klieber, W., Plagiarism Detection in Large Sets of Press Agency News Articles, TAKMA Workshop Proceedings of 17th International Conference on Database and Expert Systems Applications (DEXA 06), IEEE Computer Society, Krakow, Poland, 2006.
15. [Tochtermann et al. 2005] Tochtermann, K., Granitzer, M., Sabol, V., Klieber, W., MISTRAL: Service Orientierte Cross-Media Techniken zur Extraktion von Semantic aus multimedia Daten und Deren Anwendung, In proceedings of Semantics 2005, Vienna, Trauner Verlag
16. [Kienreich et al. 2005a] Kienreich, W., Sabol, V., Granitzer, M., Klieber, W., Lux, M., Sarka, W., A Visual Query Interface for a Very Large Newspaper Article Repository, Proceedings of 16th International Conference on Database and Expert Systems Applications (DEXA 05), IEEE Computer Society, Copenhagen, Denmark, 2005.
17. [Kienreich et al. 2005b] Kienreich, W., Granitzer, M., Sabol, V., Klieber, W., Lux, M., Sarka, W., Visual Analysis of Search Results Obtained from Very Large Newspaper Article Repository, Proceedings of ISGI 2005, CODATA International Symposium on Generalization of Information, Berlin, Germany, 2005.
18. [Kienreich et al. 2005c] Kienreich, W., Sabol, V., Ley, T., Lindstaedt, S. N., Koronakis, P., Droschl, G., MagIR: Distributed Creation, Administration and Reutilization of Multimedia Presentation Content, Proceedings of WM05 - Workshop IT Tools for Knowledge Management Systems, 2005.
19. [Granitzer et al. 2004] Granitzer, M., Kienreich, W., Sabol, V., Andrews, K., Klieber, W., Evaluating a System for Interactive Exploration of Large, Hierarchically Structured Document Repositories, InfoVis 2004, the tenth annual IEEE Symposium on Information Visualization, Austin, Texas, USA
20. [Andrews et al. 2004] Andrews, K., Kienreich, W., Sabol, V., Granitzer, M., The Visualisation of Large Hierarchical Document Spaces with InfoSky, Proceedings of CODATA Prague Workshop on Information Visualisation, Presentation and Design, Prague, Czech, 2004.
21. [Kienreich et al. 2003b] Kienreich, W., Sabol, V., Granitzer, M., Kappe, F., Andrews, K., InfoSky: A System for Visual Exploration of Very Large, Hierarchically Structured Knowledge Spaces, Proceedings der GI

Workshopwoche, Workshop der Fachgruppe Wissensmanagement, Karlsruhe, 2003.

22. [Lux 2003] Lux, M., Granitzer, M., Sabol, V., Kienreich, W., Becker, J., Topic Cascades: An interactive interface for exploration of clustered web search results based on the SVG standard, In Proceedings of the Seventh International Conference on Knowledge-Based Intelligent Information, Bd. I, 967-974, Springer, Oxford, England, 2003.
23. [Kappe et al. 2003] Kappe, F., Droschl, G., Kienreich, W., Sabol, V., Becker, J., Andrews, K., Granitzer, M., Tochtermann, K., Auer, P., InfoSky: Visual Exploration of Large Hierarchical Document Repositories, Proceedings of HCI 2003 International, Creta, Greece, 2003.
24. [Granitzer et al. 2003] Granitzer, M., Kienreich, W., Sabol, V., Dsinger, G., WebRat: Supporting Agile Knowledge Retrieval through Dynamic, Incremental Clustering and Automatic Labelling of Web Search Result Sets, Proceedings of 1st IEEE Workshop on Knowledge Management for Distributed, Agile Processes, Linz, Austria, 2003.
25. [Tochtermann et al. 2003] Tochtermann, K., Sabol, V., Kienreich, W., Granitzer, M., Becker, J., Enhancing Environmental Search Engines with Information Landscapes, Proceedings of International Symposium on Environmental Software Systems, Semmering, Austria, 2003.
26. [Sabol et al. 2002a] Sabol, V., Kienreich, W., Granitzer, M., Becker, J., Tochtermann, K., Andrews, K., Applications of a Lightweight, Web-Based Retrieval, Clustering and Visualisation Framework, Proceedings of the 4th International Conference on Practical Aspects of Knowledge Management, Vienna Austria, 2002.
27. [Tochtermann et al. 2002] Tochtermann, K., Sabol, V., Kienreich, W., Granitzer, M., Becker, J., Intelligent Maps and Information Landscapes: Two new Approaches to support Search and Retrieval of Environmental Information Objects., Proceedings of the International Symposium on Environmental Informatics, Vienna Austria, 2002.
28. [Andrews et al. 2001] Andrews, K., Gütl, C., Moser, J., Sabol, V., Lackner, W., Search Result Visualisation with xFIND, in Proceedings of the Second International Workshop on User Interfaces to Data Intensive Systems (UIDIS 2001), Zurich, Switzerland, 2001.

Posters:

1. [Onn et al. 2011] Onn, K. W., Sabol, V., Granitzer, M., Kienreich, V., Lucose, D, A Visual SOA-based Ontology Alignment Tool, poster, in Proceedings of the Sixth International Workshop on Ontology Matching (OM-2011), 2011.
2. [Sabol et al. 2010a] Sabol, V., Seifert, C., Kienreich, W., Integrating Node-Link-Diagrams and Information Landscapes: A Path-Finding Approach, Poster and Demo at EuroVis 2010.
3. [Sabol et al. 2009b] Sabol, V., Kienreich, W., Visualizing Temporal Changes in Information Landscapes, Poster and Demo at the EuroVis 2009.
4. [Sabol et al. 2005] Sabol, V., Granitzer, M., Tochtermann, K., Sarka, W., MISTRAL Measurable, Intelligent and Reliable Semantic Extraction and Retrieval of Multimedia Data, 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies, London, UK, 2005.
5. [Andrews et al. 2003] Andrews, K., Kienreich, W., Sabol, V., Granitzer, M., Visualising Large Hierarchically Structured Document Repositories with InfoSky, Interactive Poster, InfoVis 2003, Seattle, USA.

Thesis:

1. [Sabol 2001] Sabol, V., Visualisation Islands: Interactive Visualisation and Clustering of Search Result Sets, Masters Thesis at Graz University of Technology, Institute for Information Processing and Computer Supported New Media (IICM), 2001.

Other, non peer-reviewed publications:

1. [Sabol et al. 2011] Sabol, V., Kern, R., Kump, B., Pammer, V., Granitzer, M., Knowledge Extraction and Integration using Automatic and Visual Methods, PlanetData Project One-Day Strategic Workshop for Call 2, November 2011.
2. [Granitzer et al. 2011] Granitzer, M., Sabol, V., Kienreich, W., Lukose, D., Weng Onn, K., Visual Analysis on Linked Data An Opportunity for both Fields, The 2011 STI Semantic Summit, Riga, Latvia, 2011.
3. [Atzmüller & Sabol 2007] Atzmüller, P., Sabol, V., Semantische Patentanalyse und Competitive Intelligence in der voestalpine Stahl GmbH, Slides and Presentation at the Praxisforum of the I-Know'09, the 9th International Conference on Knowledge Management and Knowledge Technologies, Graz, 2009.

4. [Sabol et al. 2007] Sabol, V., Gütl, C., Neidhart, T., Juffinger, A., Klieber, W., Granitzer, M., Visualization Metaphors for Multi-modal Meeting, Workshop Multimedia Semantics - The Role of Metadata (WMSRM 07), Proceedings Band "Aachener Informatik Berichte", Aachen, 2007.
5. [Lux et al. 2004] Lux, M., Granitzer, M., Kienreich, W., Sabol, V., Klieber, W., Sarka, W., Cross Media Retrieval in Knowledge Discovery, Springer Verlag: Lecture Notes in Computer Science, Vienna, Austria, 2004.
6. [Kienreich et al. 2003a] Kienreich, W., Sabol, V., Granitzer, M., Becker, J., Tochtermann, K., Themenkarten als Ergänzung zu hierarchiebasierter Navigation und Suche in Wissensmanagementsystemen, 4. Oldenburger Forum Wissensmanagement, Oldenburg, Germany, 2003.
7. [Becker 2002a] Becker, J., Granitzer, M., Kienreich, W., Sabol, V., WebRat, Telematik Ingenieur Verband: published in TELEMATIK 03/2002, Graz, Austria
8. [Becker 2002b] Becker, J., Lux, M., Klieber, W., Sabol, V., Kienreich, W., Knowledge Discovery Space, Telematik Ingenieur Verband: published in TELEMATIK 03/2002, Graz, Austria

I have been the advisor for one Diploma Thesis, two Bachelor Theses and one Master Praktikum in the area of information visualization and human-computer interaction:

1. [Kandlhofer 2008] Kandlhofer, M., Einbindung neuer Visualisierungskomponenten in ein Multiple Coordinated Views Framework, Endbericht Master-Praktikum, Institute for Knowledge Management, Technical University of Graz, 2008.
2. [Krnjic 2008] Krnjic, V., Usability Evaluierung einer Multiple Coordinated Views Applikation, Bakkalaureatsarbeit an der Technischen Universität Graz in Informatik, Austria, 2008.
3. [Weitlaner 2009] Weitlaner, D., Usability-Evaluierung von Visualisierungskomponenten zur temporal-thematischen Analyse von Textdokumentsätzen, Bakkalaureatsarbeit an der Technischen Universität Graz in Softwareentwicklung-Wirtschaft, Austria, 2009.
4. [Kandlhofer 2009] Kandlhofer, M., Visualisierung und dynamische Aggregation von semantischen Graphen, Masters Thesis at Graz University of Technology, Austria, 2009.

I am the co-inventor of the following patents:

1. European Patent Application #02 007 742.6: Data Processing System, With: Kappe, F., Kienreich, W., April, 2002.
2. US Patent Application #20030231209: Data Processing System, With: Kappe, F., Kienreich, W., April, 2003

Bibliography

- [ACM 1998] The ACM Computing Classification System, 1998, <http://www.acm.org/about/class/ccs98-html>, last visited in August 2011.
- [Agrawal et al. 1998] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopoulos, Prabhakar Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in Proceedings of the ACM SIGMOD '98 international conference on Management of data, pages 94-105, 1998.
- [Andrews et al. 2001] K. Andrews, C. Gütl, J. Moser, V. Sabol, W. Lackner, Search Result Visualisation with xFIND, Proceedings of the Second International Workshop on User Interfaces to Data Intensive Systems (UIDIS'01), 2001.
- [Andrews et al. 2002] Andrews, K., Kienreich, W., Sabol, V., Becker, J., Kappe, F., Droschl, G., Granitzer, M., Auer, P., Tochtermann, K., The InfoSky Visual Explorer: Exploiting Hierarchical Structure and Document Similarities, Palgrave Journals: Information Visualization, London, England, 2002.
- [Andrews et al. 2003] Andrews, K., Kienreich, W., Sabol, V., Granitzer, M., Visualising Large Hierarchically Structured Document Repositories with InfoSky, interactive poster at InfoVis 2003, Seattle.
- [Andrews et al. 2004] Andrews, K., Kienreich, W., Sabol, V., Granitzer, M., The Visualisation of Large Hierarchical Document Spaces with InfoSky, Proceedings of CODATA Prague Workshop on Information Visualisation, Presentation and Design, Prague, Czech, 2004.
- [Andrews 2010a] K. Andrews, Information Visualisation, Course Notes, version of 22 Jan 2010, <http://courses.iicm.tugraz.at/ivis/ivis.pdf>, retrieved in January 2011.

- [Andrews 2010b] K. Andrews, Human-Computer Interaction, Lecture Notes, version of 08 Nov 2010, <http://courses.iicm.tugraz.at/hci/hci.pdf>, retrieved in January 2011.
- [Andrienko & Andrienko] Andrienko, G., Andrienko, N., Coordinated Multiple Views: a Critical View, In CMV '07: Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2007. pp. 72-74.
- [APA-DeFacto 2011] APA-DeFacto GmbH, <http://www.apa-defacto.at>, last visited in January 2011.
- [Arthur & Vassilvitskii 2007] D. Arthur and S. Vassilvitskii, k-means++: the advantages of careful seeding, in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pages 1027-1035, 2007.
- [Atzmüller & Sabol 2007] Atzmüller, P., Sabol, V., Semantische Patentanalyse und Competitive Intelligence in der voestalpine Stahl GmbH, Slides and Presentation at the Praxisforum of the I-Know'09, the 9th International Conference on Knowledge Management and Knowledge Technologies, Graz, 2009,
- [Aurenhammer 1991] Franz Aurenhammer, Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys, Vol. 23, Issue 3, pages 345-405, 1991.
- [Baeza-Yates & Ribeiro-Neto 2011] Baeza-Yates, R., Ribeiro-Neto, B., Modern Information Retrieval: The Concepts and Technology behind Search, 2nd Edition, 944 pages, Addison-Wesley Professional, 2011.
- [Baldonado et al. 2000] Michelle Q. Wang Baldonado, Allison Woodruff, Allan Kuchinsky, Guidelines for Using Multiple Views in Information Visualization, in Proceedings of AVI, 2000, pages 110-119.
- [Barnes & Hut 1986] J. Barnes, P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. Nature, 324(4), December 1986.
- [Becker 2002a] Becker, J., Granitzer, M., Kienreich, W., Sabol, V., WebRat, Telematik Ingenieur Verband: published in TELEMATIK 03/2002, Graz, Austria
- [Becker 2002b] Becker, J., Lux, M., Klieber, W., Sabol, V., Kienreich, W., Knowledge Discovery Space, Telematik Ingenieur Verband: published in TELEMATIK 03/2002, Graz, Austria

- [Berkhin 2002] Berkhin, P., Survey of Clustering Data Mining Techniques, Technical Report, Accrue Software, 2002.
- [Berry et al. 1995] Berry, M., Dumais, S., Landauer, T., O'Brien, G., Using linear algebra for intelligent information retrieval. *SIAM Review*, 37, 4, 573-595., 1995.
- [Berry & Groenen 2010] I. Borg, P. J. F. Groenen, Modern Multidimensional Scaling: Theory and Applications, 2nd edition, Springer, 2010.
- [Bertschi et al. 2011] Bertschi, S., Bresciani, S., Crawford, T., Goebel, R., Kienreich, W., Lindner, M., Sabol, V., Vande Moere, A., What is Knowledge Visualization? Opinions on Current and Future State, in Proceedings of the 15th International Conference Information Visualisation (IV'11), 2011.
- [Boyack et al. 2000] Kevin W. Boyack , Brian N. Wylie , George S. Davidson , David K. Johnson, Analysis of Patent Databases Using VxInsight, Workshop on New Paradigms in Information Visualization and Manipulation, Washington, DC (US), 2000.
- [Boyack et al. 2002] Kevin W. Boyack , Brian N. Wylie , George S. Davidson, Domain Visualization Using VxInsight for Science and Technology Management, *Journal of the American Society for Information Science and Technology*, Vo. 53, pages 764 - 774, 2002.
- [Bradley & Fayyad 1998] P. S. Bradley, P.S., Fayyad, U.M., Refining initial points for K-Means clustering, in Proceedings of the 15th International Conference on Machine Learning, pp. 9199, 1998.
- [Chalmers and Chitson 1992] M. Chalmers, P. Chitson, Bead: Explorations in Information Visualisation, in Proceedings of ACM SIGIR92 (Copenhagen), pp. 330-337, 1992.
- [Chalmers 1993] Chalmers, M., Using a Landscape Metaphor to Represent a Corpus of Documents, in Proceedings of the European Conference on Spatial Information Theory, Vol. 716, pp. 377-390, 1993.
- [Chalmers 1996] M. Chalmers, A linear iteration time layout algorithm for visualising high-dimensional data. In Proceedings Visualization 96, pages 127-132, San Francisco, California, October 1996. IEEE Computer Society.
- [Cao et al. 2010] N. Cao, J. Sun, Y.-R. Lin, D. Gotz, S. Liu, H. Qu, FacetAtlas: Multifaceted Visualization for Rich Text Corpora, in IEEE

- Transactions on Visualization and Computer Graphics, Vol. 16, Issue 6, 2010, pages 1172–1181.
- [Cha 2007] Cha, S.-H., Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions, in *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1(4), 2007, pp. 300-307.
- [Chen 2006] Chaomei Chen, *Information Visualization: Beyond the Horizon*, 2nd edition, 320 pages, Springer, 2006.
- [Chin et al. 2009] George Chin, Mudita Singhal, Grant Nakamura, Vidhya Gurumoorthi, Natalie Freeman-Cadoret, *Visual analysis of dynamic data streams*, *Information Visualization*, Vol. 8(3), Palgrave Macmillan, 2009.
- [Clustermap 2011] Aduna Clustermap, <http://www.aduna-software.com/technology/clustermap>, last visited in January 2011.
- [Cutting et al. 1992] D. R. Cutting, J. O. Pedersen, D. Karger, and J. W. Tukey, Scatter/gather: A cluster-based approach to browsing large document collections, In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference*, pages 318-329, ACM Press, 1992. (Also available as Xerox PARC technical report SSL-92-02.)
- [Cutting et al. 1993] Douglass R. Cutting, David R. Karger, Jan O. Pedersen. Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections, in *Proceedings of the Sixteenth Annual International ACM SIGIR Conference*, 1993.
- [Das & Martins 2007] Das, D., Martins, A.F.T., *A Survey on Automatic Text Summarization*, Literature Survey, Language Technologies Institute, Carnegie Mellon University, 2007.
- [Davidson et al. 1998] George S. Davidson , Bruce Hendrickson , David K. Johnson , Charles E. Meyers , Brian N. Wylie, Knowledge Mining With VxInsight: Discovery Through Exploration, *Journal of Intelligent Information Systems*, Vol. 11, pages 259 - 258, 1998.
- [Davidson et al. 2001] George S. Davidson, Brian N. Wylie, Kevin W. Boyack, Cluster Stability and the Use of Noise in Interpretation of Clustering , *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, pages 23 - 30.

- [DecisionSite 2001] TIBCO Spotfire DecisionSite Enterprise Analytics Product Suite, 2001, <http://spotfire.tibco.com/products/decisionsite.cfm>, last visited in January 2011.
- [Defays 1977] Defays, D. An efficient algorithm for a complete link method, *The Computer Journal* (British Computer Society), 20 (4), pages 364-366, 1977.
- [Dempster et al. 1977] Dempster, A., Laird, N., Rubin, D., "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society, Series B*, 39, 1, pp. 1-38, 1977.
- [DIVINE 2011] DIVINE - Dynamic Integration and Visualization of Information from Multiple Evidence Sources, FIT-IT Semantic Systems Project, 2011, <http://www.weblyzard.com/divine/>, last accessed in August 2011.
- [dmoz 2004] Dmoz Open Directory Project, <http://www.dmoz.org/>, last visited in January 2011.
- [Dykes 2005] Dykes, J., MacEachren, A.M., Kraak, M.J. (eds.): *Exploring Geovisualization*. Elsevier, 2005.
- [Eibl et al. 2001] Eibl, M., Mandl, T., Stempfhuber, M., *Metaphors vs. Visual Formalisms in Visual Information Seeking*, In *Proceedings of the Panhellenic Conference on Human Computer Interaction*, Greece, 2001, pages 13 - 18.
- [Ester et al. 1996] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226-231, 1996.
- [Euzenat et al. 2004] Euzenat, J., Le Bach, T., Barrasa, J., Bouquet, P., De Bo, J., Dieng, R., Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Maynard, D., Napoli, A., Stamou, G., Stuckenschmidt, H., Shvaisko, P., Tessaris, S., Van Acker, S., Zahrayeu, I., D2.2.3: State of the Art on Ontology Alignment, KWEB/2004/D2.2.3/v1.2; Technical Report for Knowledge Web Project IST-2004-507482, Knowledge Web Consortium, August 2004.

- [Faloutsos & Lin 1995] Faloutsos, C., Lin, K.-I., FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets, in Proceedings of the 1995 ACM SIGMOD international conference on Management of data, pages 163 - 174, 1995.
- [Fayyad et al. 1996] Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; From data mining to knowledge discovery in databases, *AI Magazine*, Vol. 17, pp. 37-54, 1996.
- [Fluit 2005] C. Fluit , AutoFocus: Semantic Search for the Desktop, in Proceedings of the Ninth International Conference on Information Visualisation (IV'05), pages 480-487, 2005.
- [Fodor 2002] Fodor, I.K., A Survey of Dimension Reduction Techniques, LLNL technical report, June 2002. <https://computation.llnl.gov/casc/sapphire/pubs/148494.pdf>, retrieved in January 2011.
- [Fruchterman & Reingold 1991] Thomas M. J. Fruchterman, Edward M. Reingold, Graph Drawing by Force-directed Placement, *Software: Practice and Experience*, Vol. 21, No. 11., pp. 1129-1164, 1991.
- [Gal & Shvaiko 2008] Gal, A., Shvaiko, P., Advances in Ontology Matching, in *Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web*, pp. 176-198, Springer-Verlag, Germany, 2008.
- [Granitzer et al. 2003] Granitzer, M., Kienreich, W., Sabol, V., Dösinger, G., WebRat: Supporting Agile Knowledge Retrieval through Dynamic, Incremental Clustering and Automatic Labelling of Web Search Result Sets, Proceedings of 1st IEEE Workshop on Knowledge Management for Distributed, Agile Processes, Linz, Austria, 2003.
- [Granitzer et al. 2004] Granitzer, M., Kienreich, W., Sabol, V., Andrews, K., Klieber, W., Evaluating a System for Interactive Exploration of Large, Hierarchically Structured Document Repositories, *InfoVis '04*, the tenth annual IEEE Symposium on Information Visualization, Austin, Texas, USA, 2004.
- [Granitzer 2006] M. Granitzer, KnowMiner: Konzeption und Entwicklung eines Generischen Wissenserschließungsframeworks, PhD Thesis, University of Technology Graz (Technische Universität Graz), Austria, 2006.

- [Granitzer et al. 2009] Granitzer, M., Augustin, A., Kienreich, W., Sabol, V., Taxonomy Extraction from German Encyclopedic Texts, In Proceedings of the Malaysian Joint Conference on Artificial Intelligence 2009, Kuala Lumpur, Malaysia
- [Granitzer et al. 2010] Granitzer, M., Sabol, V., Onn, K.W., Lukose, D. and Tochtermann, K., Ontology Alignment - A Survey with Focus on Visually Supported Semi-Automatic Techniques, *Future Internet*, Volume 2, Issue 3, pages 238-258, MDPI AG, 2010.
- [Granitzer et al. 2011] Granitzer, M., Sabol, V., Kienreich, W., Lukose, D., Weng Onn, K., Visual Analysis on Linked Data An Opportunity for both Fields, The 2011 STI Semantic Summit, Riga, Latvia, 2011.
- [Guha et al. 1998] Guha, S., Rastogi, R., Shim, K., CURE: An efficient clustering algorithm for large databases, In Proceedings of the ACM SIGMOD Conference, pages 73-84, 1998.
- [Hartigan & Wong 1979] Hartigan, J., Wong, M., *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 28, No. 1, pp. 100-108, 1979.
- [Havre et al. 2000] Havre, S., Hetzler, B., Nowell, L., ThemeRiver: Visualizing Theme Changes over Time, in Proceedings of the IEEE Symposium on Information Visualization, pages 115 - 123, 2000.
- [Havre et al. 2002] Havre, S. Hetzler, E., Whitney, P., Nowell, L., ThemeRiver: Visualizing Thematic Changes in Large Document Collections, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 8(1), pages 920, 2002.
- [Healey 2009] Christopher G. Healey, Perception in Visualization, Department of Computer Science, North Carolina State University, HTML, last updated 2009, <http://www.csc.ncsu.edu/faculty/healey/PP/index.html>, retrieved in January 2011.
- [Hearst 2006] Hearst, M., Clustering versus Faceted Categories for Information Exploration, *Communications of the ACM*, Vol. 49, Issue 4, pages 59 - 61, April 2006.
- [Heilig et al. 2009] Heilig, M., Demarmels, M., Rexhausen, S., Huber, S., Runge, O., Search, Explore and Navigate - Designing a Next Generation Knowledge Media Workbench, *Flirting with the future*,

- in proceedings of the Fifth Student Interaction Design Research Conference (SIDeR 09), 2009.
- [Herman et al. 2010] Herman, I., Melancon, G., Marshall, M.S., Graph visualization and navigation in information visualization: A survey, IEEE Transactions on Visualization and Computer Graphics, Vol. 6, Num. 1, 2000, pages 24–43.
- [Hewett 1992] T. T. Hewett, ACM SIGCHI curricula for human-computer interaction, ACM Technical Report, 1992, <http://old.sigchi.org/cdg/>, retrieved in January 2011.
- [Hinneburg & Leim 1998] Alexander Hinneburg, Daniel A. Keim, An Efficient Approach to Clustering in Large Multimedia Databases with Noise, Knowledge Discovery and Data Mining, Vol. 5865, pp. 58–65, AAI Press, 1998.
- [Hoffman 2005] D.D. Hoffman, Visual Intelligence: How We Create What We See, New York: W.W. Norton & Company, 1998.
- [Hu & Yoo 2004] Xiaohua Hu, Illhoi Yoo, Cluster ensemble and its applications in gene expression analysis, in Proceedings of the second conference on Asia-Pacific bioinformatics, Volume 29, pp. 297–302, 2004.
- [Hyperwave 2011] Hyperwave GmbH, Graz, Austria, <http://www.hyperwave.com>, last visited in January 2011.
- [Hyperwave IS/6 2011] Hyperwave IS/6, <http://www.hyperwave.com/e/products/is6>, last visited in January 2011.
- [IDC 2007] John F. Gantz, David Reinsel, Christopher Chute, Wolfgang Schlichting, John McArthur, Stephen Minton, Irida Xheneti, Anna Toncheva, Alex Manfrediz, The Expanding Digital Universe, A Forecast of Worldwide Information Growth Through 2010, IDC White Paper - sponsored by EMC, March 2007, <http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf>, retrieved in January 2011.
- [IDC 2008] John F. Gantz, Christopher Chute, Alex Manfrediz, Stephen Minton, David Reinsel, Wolfgang Schlichting, Anna Toncheva, The Diverse and Exploding Digital Universe, An Updated Forecast of Worldwide Information Growth Through 2011, IDC White Paper - sponsored by EMC, March

- 2008, <http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf>, retrieved in January 2011.
- [IICM 2011] Institute for Information Systems and Computer Media (IICM), Technical University Graz <http://www.iicm.tugraz.at/>, last visited in January 2011.
- [INEX 2009] INEX09 Wikipedia Documents, 2009, <http://inex.de-vries.id.au/scoreboard/>, last visited on January 2011.
- [Inselberg & Dimsdale 1987] Alfred Inselberg, Bernard Dimsdale, Parallel coordinates for visualizing multi-dimensional geometry, Proceedings of CG International '87 on Computer graphics, 1987.
- [ISO 1998] Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) Part 11: Guidance on Usability (ISO 9241-11:1998), ISO, 1998.
- [Jain et al. 1999] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Computing Surveys, Volume 31, Issue 3, Sept. 1999.
- [Java SE 6] Java Standard Edition 6, Oracle Corporation, 2006, <http://www.oracle.com/technetwork/java/javase/overview/index.html>, last visited in August 2011.
- [JOGL 1.1.1a] Java Bindings for OpenGL Version 1.1.1a, Oracle Corporation (previously Sun Microsystems), 2009, <http://download.java.net/media/jogl/builds/archive/jsr-231-1.1.1a/>, last visited in August 2011.
- [Jolliffe 2002] Jolliffe, I. T., Principal Component Analysis, 2nd edition, Springer-Verlag, 2002.
- [Jourdan & Melancon 2004] Fabien Jourdan, Guy Melancon, Multiscale hybrid MDS, in Proceedings of the Eighth International Conference on Information Visualisation (IV '04), Proceedings of the Information Visualisation, pages 388-393, 2004.
- [Kaiser & Miksch 2005] K. Kaiser, K., Miksch, S., Information extraction - a survey, Technical Report Asgaard-TR-2005-6, Vienna University of Technology, May 2005, <http://ieg.ifs.tuwien.ac.at/techreports/Asgaard-TR-2005-6.pdf>, last accessed in July 2011.

- [Kapler & Wright 2005] Kapler, T., Wright, W., Geo time information visualization, *Information Visualization*, Volume 4 Issue 2, pages 136 - 146, Palgrave Macmillan, July 2005.
- [Kappe et al. 2003] Kappe, F., Droschl, G., Kienreich, W., Sabol, V., Becker, J., Andrews, K., Granitzer, M., Tochtermann, K., Auer, P., InfoSky: Visual Exploration of Large Hierarchical Document Repositories, *Proceedings of HCI 2003 International*, Creta, Greece, 2003.
- [Kaufmann & Rousseeuw 1990] Kaufman, L., Rousseeuw, P., "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley and Sons, New York, NY., 1990.
- [Kandlhofer 2008] M. Kandlhofer, Einbindung neuer Visualisierungskomponenten in ein Multiple Coordinated Views Framework, Endbericht Master-Praktikum, Institute for Knowledge Management, Technical University of Graz, 2008.
- [Kandlhofer 2009] Martin Kandlhofer, Visualisierung und dynamische Aggregation von semantischen Graphen, Masters Thesis at Graz University of Technology, Austria, 2009.
- [Käki 2005] Käki, M., FindexSearch result categories help users when document rankings fail, in *Proceedings of ACM SIGCHI conference on Human factors in computing systems*, pages 131-140, 2005.
- [Know-Center 2011] Know-Center GmbH, Graz, Austria, <http://www.know-center.at/en/research-areas/knowledge-relationship-discovery>, last visited in January 2011.
- [KC-KRD 2011] Know-Center's Knowledge Relationship Discovery Research Area, Graz, Austria, <http://www.know-center.at/en/research-areas/knowledge-relationship-discovery>, last visited in January 2011.
- [Keim et al. 2008] Daniel A. Keim, Florian Mansmann, Daniela Oelke, and Hartmut Ziegler, *Visual Analytics: Combining Automated Discovery with Interactive Visualizations*. Discovery Science, 2008, pp. 2-14.
- [Keim et al. 2008b] Keim D. A, Mansmann F, Schneidewind J, Thomas J, Ziegler H: *Visual analytics: Scope and challenges*. *Visual Data Mining: 2008*, pages 76-90, Springer-Verlag.

- [Kerwin 2011] Kerwin, T., Survey of treemap techniques, <http://www.cse.ohio-state.edu/~kerwin/treemap-survey.html>, retrieved in January 2011.
- [Kienreich et al. 2003a] Kienreich, W., Sabol, V., Granitzer, M., Becker, J., Tochtermann, K., Themenkarten als Ergänzung zu hierarchiebasierter Navigation und Suche in Wissensmanagementsystemen, 4. Oldenburger Forum Wissensmanagement, Oldenburg, Germany, 2003.
- [Kienreich et al. 2003b] Kienreich, W., Sabol, V., Granitzer, M., Kappe, F., Andrews, K., InfoSky: A System for Visual Exploration of Very Large, Hierarchically Structured Knowledge Spaces, Proceedings der GI Workshopwoche, Workshop der Fachgruppe Wissensmanagement, Karlsruhe, 2003.
- [Kienreich et al. 2005a] Kienreich, W., Sabol, V., Granitzer, M., Klieber, W., Lux, M., Sarka, W., A Visual Query Interface for a Very Large Newspaper Article Repository, Proceedings of 16th International Conference on Database and Expert Systems Applications (DEXA 05), IEEE Computer Society, Copenhagen, Denmark, 2005.
- [Kienreich et al. 2005b] Kienreich, W., Granitzer, M., Sabol, V., Klieber, W., Lux, M., Sarka, W., Visual Analysis of Search Results Obtained from Very Large Newspaper Article Repository, Proceedings of ISGI 2005, CODATA International Symposium on Generalization of Information, Berlin, Germany, 2005.
- [Kienreich et al. 2005c] Kienreich, W., Sabol, V., Ley, T., Lindstaedt, S. N., Koronakis, P., Droschl, G., MagIR: Distributed Creation, Administration and Reutilization of Multimedia Presentation Content, Proceedings of WM05 - Workshop IT Tools for Knowledge Management Systems, 2005.
- [Kienreich et al. 2006] Kienreich, W., Information and Knowledge visualisation - an oblique view, *Mia Journal*, Vol. 0, Num. 1, MAP-CNRS, July 2006, pp 7-16.
- [Kienreich 2006] Kienreich, W., Granitzer, M., Sabol, V., Klieber, W., Plagiarism Detection in Large Sets of Press Agency News Articles, TAKMA Workshop Proceedings of 17th International Conference on Database and Expert Systems Applications (DEXA 06), IEEE Computer Society, Krakow, Poland, 2006.

- [Kienreich et al. 2007] Kienreich, W., Zechner, M., Sabol, V., Comprehensive Astronomical Visualization for a Multimedia Encyclopedia, 3rd International Symposium of Knowledge and Argument Visualization. Proceedings of IV07, 11th International Conference Information Visualisation, IEEE Computer Society, London, UK, 2007.
- [Kienreich & Seifert 2010] Kienreich, W., Seifert, C., An application of edge bundling techniques to the visualization of media analysis results, In Proceedings of the 14th International Conference on Information Visualization, 2010, IEEE Computer Society Press, pages 375 - 380.
- [Klieber et al. 2006] Klieber, W., Sabol, V., Granitzer, M., Kienreich, W., Kern, R., KnowMiner - Ein Service orientiertes Knowledge Discovery Framework, GI-Edition 2006, Bonner Köllen Verlag, 2006.
- [Klieber et al. 2009a] Klieber, W., Sabol, V., Muhr, M., Kern, R., Öttl, G., Granitzer, M., Knowledge Discovery Using the KnowMiner Framework, IADIS International Conference Information Systems 2009, pages 307-314.
- [Klieber et al. 2009b] Klieber, W., Sabol, V., Muhr, M., Granitzer, M., Using Ontologies For Software Documentation, Malaysian Joint Conference on Artificial Intelligence, 2009.
- [Know-Center 2011] Know-Center, Austria's Competence Center for Knowledge Management and Knowledge Technologies, 2011, <http://www.know-center.tugraz.at/>
- [KnowMiner 2011] The KnowMiner Knowledge Discovery Framework, <http://www.knowminer.at/>, last visited in January 2011.
- [Kohonen 1988] T. Kohonen, Self-Organization and associative memory, Springer-Verlag, Berlin, Second Edition, 1988.
- [Kohonen et al. 2000] Kohonen, T., S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, A. Saarela, Self-organization of a Massive Document Collection, in IEEE Transactions on Neural Networks, Vol. 11(3), pp. 574-585, 2000.
- [Kotis & Lanzenberger 2008] Kotis, K.; Lanzenberger, M. Ontology Matching: Status and Challenges. IEEE Intelligent Systems, Vol. 23 (6), pages 84 - 85, 2008.

- [Krishnan et al. 2007] M. Krishnan S. Bohn W. Cowley V. Crow J. Nieplocha, Scalable Visual Analytics of Massive Textual Datasets, 21st IEEE International Parallel and Distributed Processing Symposium. Long Beach, USA, 2007.
- [Krnjic 2008] Vesna Krnjic, Usability Evaluierung einer Multiple Coordinated Views Applikation, Bakkalaureatsarbeit an der Technischen Universität Graz in Informatik, Austria, 2008.
- [Kruskal 1978] Joseph B. Kruskal, Multidimensional Scaling (Quantitative Applications in the Social Sciences), Sage Publications, Inc., 1978.
- [Lengler 2007] Lengler R., Eppler M., Towards a Periodic Table of Visualization Methods for Management, in Proceedings of the Conference on Graphics and Visualization in Engineering, 2007, pages 1 - 6.
- [Lex et al. 2008] Lex, E., Seifert, C., Kienreich, W., Granitzer, M., A generic framework for visualizing the news article domain and its application to real-world data, Journal of Digital Information Management 6, 2008, pages 434 - 441.
- [Likert 1932] Likert, R., A Technique for the Measurement of Attitudes, Archives of Psychology 140, pages 155, 1932.
- [Lux 2003] Lux, M., Granitzer, M., Sabol, V., Kienreich, W., Becker, J., Topic Cascades: An interactive interface for exploration of clustered web search results based on the SVG standard, In Proceedings of the Seventh International Conference on Knowledge-Based Intelligent Information, Bd. I, 967-974, Springer, Oxford, England, 2003.
- [Lux 2004] Mathias Lux, Magick Ein Werkzeug für Cross-Media Clustering und Visualisierung, Masters Thesis at Graz University of Technology, Austria, 2004, <http://know-center.tugraz.at/wp-content/uploads/2010/12/Diplomarbeit-eBook-mlux.pdf>, retrieved in January 2011.
- [Lux et al. 2004] Lux, M., Granitzer, M., Kienreich, W., Sabol, V., Klieber, W., Sarka, W., Cross Media Retrieval in Knowledge Discovery, Springer Verlag: Lecture Notes in Computer Science, Vienna, Austria, 2004.
- [m2n IMF 2011] m2n Intelligence Management Framework, <http://www.m2n.at>, last visited August 2011.

- [Mackinlay et al. 1991] J. D. Mackinlay, G. G. Robertson, S. K. Card, The perspective wall: detail and context smoothly integrated, in Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology, pages 173 - 179. ACM Press, 1991.
- [MDS API 2005] MDS API, 2005, <http://www.lirmm.fr/fjordan/Projets/MDS/MDSAPI.html>, retrieved in January 2011.
- [MISTRAL 2005] MISTRAL - Measurable Intelligent and Reliable Semantic Extraction and Retrieval of Multimedia Data, A FIT-IT Project, <http://mistrall-project.tugraz.at/>, 2005, Austria, last visited in August 2011.
- [Morrison et al. 2001] Morrison, A., G. Ross, M. Chalmers, Combining and comparing clustering and layout algorithms, Technical Report, Department of Computing Science, University of Glasgow, 2001.
- [Morrison et al. 2002] Morrison, A., G. Ross, M. Chalmers, A Hybrid Layout Algorithm for Sub-Quadratic Multidimensional Scaling, In Proceedings of the IEEE Symposium on Information Visualization, 2002, Boston, pages 152-158.
- [Morrison et al. 2003] Morrison, A., G. Ross, M. Chalmers, Fast multidimensional scaling through sampling, springs and interpolation, Information Visualization, Vol. 2(1), 2003, pages 68-77.
- [Morrison & Chalmers 2003] A. Morrison, M. Chalmers, Improving Hybrid MDS with Pivot-Based Searching, in Proceedings of the IEEE Symposium on Information Visualization, Seattle, 2003, pages 85-90.
- [Morrison & Chalmers 2004] A. Morrison, M. Chalmers, A Pivot-Based Routine for Improved Parent-Finding in Hybrid MDS, Information Visualization, Vol. 3(2), pages 109-122, 2004.
- [Mosquera-Caro et al. 2004] M. Mosquera-Caro, S.M. Martin, E. Andries, J. Potter, K. Ar, Y. Xu, H. Kang, X. Wang, M. H. Murphy, P. Helman, R. Veroff, D.M. Haaland, S. Atlas, J. Cowie, C. Fields, V. Sibirtsev, G. Davidson, C. Willman, Application of Multidisciplinary Analysis to Gene Expression, Technical Report, 2004, Sandia National Laboratories, <http://prod.sandia.gov/techlib/access-control.cgi/2004/040161.pdf>, retrieved in January 2011.

- [MPEG7 2004] MPEG-7 Overview (version 10), ISO/IEC, 2004, <http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm>, retrieved in January 2011.
- [MRDCS 2011] MRDCS - Malaysian Research and Development Classification System, 2011, available from <http://www.mastic.gov.my/>, last visited in August 2011.
- [Muller & Schumann 2003] Muller, W., Schumann, H., Visualization methods for time-dependent data - an overview, in Proceedings of the 35th conference on Winter simulation: driving innovation, pages 737 - 745, New Orleans, 2003.
- [MIMOS 2011] MIMOS Berhad, Kuala Lumpur, Malaysia, 2011, <http://www.mimos.my>, last visited in January 2011.
- [Muhr & Granitzer 2009] Muhr, M., Granitzer, M., Automatic Cluster Number Selection using a Split and Merge K-Means Approach, in Proceedings of the 2009 20th International Workshop on Database and Expert Systems Application, pages 363-367, 2009.
- [Muhr et al. 2010] Muhr, M., Sabol, V., Granitzer, M., Scalable Recursive Top-Down Hierarchical Clustering Approach with implicit Model Selection for Textual Data Sets, IEEE Computer Society: 7th International Workshop on Text-based Information Retrieval in Proceedings of 21th International Conference on Database and Expert Systems Applications (DEXA 10), 2010.
- [Müller 2005] Müller, F., Granularity based multiple coordinated views to improve the information seeking process, PhD Thesis, University of Konstanz, Germany, 2005.
- [NameVoyager 2011] NameVoyager: Explore name trends letter by letter, <http://www.babynamewizard.com/voyager>, last visited in January 2011.
- [Nanas et al. 2003] N. Nanas, V. Uren and A. de Roeck, A Comparative Study of Term Weighting Methods for Information Filtering, Technical Report, KMi, No. KMI-TR-128, May 2003, <http://mcs.open.ac.uk/nn79/files/kmi-tr-128.pdf>, retrieved in July 2011.
- [Nardi & Zарmer 1990] Nardi, B.A.; Zарmer, C.L., Beyond Models and Metaphors: Visual Formalisms in User Interface Design, HP Labs Technical Report, Software and Systems Laboratory, HPL-90-149,

- September 1990, <http://www.hpl.hp.com/techreports/90/HPL-90-149.pdf>, last accessed August 2011.
- [Neidhart 2005] Thomas Neidhart, Semiautomatische Erstellung von Wissenslandkarten mittels Knowledge Mining Techniken, Masters Thesis at Graz University of Technology, Austria, 2005, <http://know-center.tugraz.at/wp-content/uploads/2010/12/Diplomarbeit-Thomas-Neidhart.pdf>, retrieved in January 2011.
- [North & Schneiderman 1999] Chris North, Ben Schneiderman, Snap-Together Visualization: Coordinating Multiple Views to Explore Information, University of Maryland Computer Science Dept. Technical Report CS-TR-4020, 1999.
- [North 2000] C. North, User Interface for Coordinating Visualizations based on Relational Schemata: Snap-Together Visualization, Doctoral Dissertation at University of Maryland Computer Science Department, 2000.
- [Nowell et al. 1996] L.T. Nowell, R.K. France, E.A. Fox, Visualizing search results with Envision, in Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, System Demonstrations: Abstracts, pages 338-339, 1996.
- [NVAC 2005] Illuminating the Path: The Research and Development Agenda for Visual Analytics, Book produced by the National Visualization and Analytics Center, Editors: Thomas, J.J., Cook, K.A. August 2005, <http://nvac.pnl.gov/agenda.stm>, retrieved in January 2011.
- [Okabe et al. 2000] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, Sung Nok Chiu, Spatial Tesselations: Concepts and Applications of Voronoi Diagrams, second edition, 671 pages, Wiley, 2000.
- [Old 2002] Old, L. John, Information Cartography: Using GIS for visualizing non-spatial data, Proceedings of the ESRI International Users' Conference, San Diego, CA, 2002, <http://proceedings.esri.com/library/userconf/proc02/pap0239/p0239.htm>, retrieved in January 2011.
- [Pattison & Phillips 2000] T. Pattison, M. Phillips, View coordination architecture for information visualization, in Proceedings of the 2001 Asia-Pacific Symposium on Information Visualisation, Volume 9, pages 165-169, Australia, 2001.

- [Pelleg & Moore 2000] Pelleg, D., Moore, A., X-means: Extending K-means with efficient estimation of the number of clusters, in Proceedings of the 17th International Conference on Machine Learning, pages 727-734, 2000.
- [Plaisant et al. 1996] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, Ben Shneiderman, LifeLines: Visualizing Personal Histories, in Proceedings of the SIGCHI conference on Human factors in computing systems: common ground (CHI '96), pages 221 - 227, 1996.
- [Plaisant et al. 1998] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, B. Shneiderman, LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records, Technical Report CS-TR-3943, University of Maryland Computer Science Department, 1998, <http://drum.lib.umd.edu/bitstream/1903/971/2/CS-TR-3943.pdf>
- [prefuse 2007] prefuse information visualization toolkit. <http://www.prefuse.org>, last visited in January 2011.
- [RAVEN 2008] RAVEN - Relation Analysis and Visualization for Evolving Network, FIT-IT Semantic Systems Research Project, 2008, <http://www.modul.ac.at/nmt/raven/>, last accessed in August 2011.
- [RCV1 2000] Reuters Corpus, Volume 1, English language, 2000, <http://trec.nist.gov/data/reuters/reuters.html>, retrieved in January 2011.
- [Risch et al. 1998] Risch, J.S., Rex, D.B., Dowson, S.T., Walters, T.B., May, R.A., Moon, B.D.: Readings in information visualization. chap. The STARLIGHT information visualization system, pp. 551560. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [Roberts 2007] Jonathan C. Roberts, State of the Art: Coordinated & Multiple Views in Exploratory Visualization, in Proceedings of the 5th International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV 2007). IEEE Computer Society Press, July 2007.
- [Sabol 2001] V. Sabol, Visualisation Islands: Interactive Visualisation and Clustering of Search Result Sets, Masters Thesis at Graz University of Technology, Austria, 2001.

- [Sabol et al. 2002a] Sabol, V., Kienreich, W., Granitzer, M., Becker, J., Tochtermann, K., Andrews, K., Applications of a Lightweight, Web-Based Retrieval, Clustering and Visualisation Framework, Proceedings of the 4th International Conference on Practical Aspects of Knowledge Management, Vienna Austria, 2002.
- [Sabol et al. 2004] V. Sabol, W. Kienreich, M. Granitzer, M. Lux, K. Andrews, K. Tochtermann, WebRat: Dynamic, Incremental Visualisation, Clustering and Automatic Labelling of Search Result Snippets and Document Collections, Technical Report, Know-Center GmbH, 2004. (available on request)
- [Sabol et al. 2005] Sabol, V., Granitzer, M., Tochtermann, K., Sarka, W., MISTRAL Measurable, Intelligent and Reliable Semantic Extraction and Retrieval of Multimedia Data, 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies, London, UK, 2005.
- [Sabol et al. 2007] Sabol, V., Gütl, C., Neidhart, T., Juffinger, A., Klieber, W., Granitzer, M., Visualization Metaphors for Multi-modal Meeting, Workshop Multimedia Semantics - The Role of Metadata (WMSRM 07), Proceedings Band "Aachener Informatik Berichte", Aachen, 2007.
- [Sabol et al. 2007] Sabol, V., Granitzer, M., Kienreich, W., Fused Exploration of Temporal Developments and Topical Relationships in Heterogeneous Data Sets, in Proceedings of the 3rd International Symposium of Knowledge and Argument Visualization, 11th International Conference Information Visualisation (IV07), IEEE Computer Society, London, UK, 2007.
- [Sabol et al. 2008a] Sabol, V., Andrews, K., Kienreich, W., Granitzer, M., Text mapping: Visualising Unstructured, Structured, and Time-Based Text Collections, Intelligent Decision Technologies, Vol 2, No. 2, IOS Press, 2008, pages 117 - 128.
- [Sabol et al. 2008b] Sabol, V., Kienreich, W., Granitzer, M., Visualisation Techniques for Analysis and Exploration of Multimedia Data, book chapter in Multimedia Semantics - The Role of Metadata, Eds. Granitzer, M., Lux, Spaniol, M., Springer-Verlag, 2008.
- [Sabol et al. 2008c] Sabol, V., Scharl, A., Visualizing Temporal-Semantic Relations in Dynamic Information Landscapes, GeoVisualization of

Dynamics, Movement and Change Workshop at the AGILE 2008 Conference, Spain.

- [Sabol et al. 2009a] Sabol, V., Kienreich, W., Muhr, M., Klieber, W., Granitzer, M., Visual Knowledge Discovery in Dynamic Enterprise Text Repositories, Proceedings of the 13th International Conference on Information Visualisation (IV09), IEEE Computer Society, 2009.
- [Sabol et al. 2009b] Sabol, V., Kienreich, W., Visualizing Temporal Changes in Information Landscapes Poster and Demo at the EuroVis 2009.
- [Sabol et al. 2010a] Sabol, V., Seifert, C., Kienreich, W., Integrating Node-Link-Diagrams and Information Landscapes: A Path-Finding Approach, Poster and Demo at EuroVis 2010.
- [Sabol et al. 2010b] Sabol, V., Syed, K.A.A., Scharl, A., Muhr, M., Hubmann-Haidvogel, A., Incremental Computation of Information Landscapes for Dynamic Web Interfaces, Proceedings of the 10th Brazilian Symposium on Human Factors in Computer Systems, 205-208, 2010.
- [Sabol et al. 2011] Sabol, V., Kern, R., Kump, B., Pammer, V., Granitzer, M., Knowledge Extraction and Integration using Automatic and Visual Methods, PlanetData Project One-Day Strategic Workshop for Call 2, November 2011.
- [Sebastiani 2002] Sebastiani, F., Machine learning in automated text categorization, ACM Computing Surveys (CSUR), Volume 34, Issue 1, pages 1-47, March 2002.
- [Seifert et al. 2008] Seifert, C., Kump, B., Kienreich, W., Granitzer, G., Granitzer, M.: On the beauty and usability of tag clouds, In Proceedings of the International Conference on Information Visualisation (IV), IEEE Computer Society, Los Alamitos, USA, 2008, pp. 1725.
- [Seifert et al. 2010a] Seifert, C., Sabol, V., Kienreich, W., Stress Maps: Analysing Local Phenomena in Dimensionality Reduction Based Visualizations, European Symposium Visual Analytics Science and Technology (EuroVAST), 2010.
- [Seifert et al. 2010b] Seifert, C., Vedran Sabol, V., Granitzer, M., (2010), Classifier Hypothesis Generation Using Visual Analysis Methods, NDT: Networked Digital Technologies, Volume 87, 2010, pages 98 - 111.

- [Scharl & Tochtermann 2007] Scharl, A., Tochtermann, K.: *The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society* (Advanced Information and Knowledge Processing), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
- [Shneiderman 1991] Shneiderman, B., Tree visualization with Tree-maps: A 2-d space-filling approach. *ACM Trans. Graphic.* 1991, 11, pages 9299.
- [Shneiderman 1996] B. Shneiderman, *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336-343, Washington. IEEE Computer Society Press, 1996.
- [Shneiderman 1998-2009] Shneiderman, B., Treemaps for space-constrained visualization of hierarchies, <http://www.cs.umd.edu/hcil/treemap-history/index.shtml>, retrieved in January 2011.
- [Shneiderman 2002] Shneiderman, B. *Inventing discovery tools: Combining information visualization with data mining*. *Information Visualization*, 1(1), pp. 5-12, Palgrave Macmillan, 2002.
- [Shneiderman et al. 2009] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 5th Edition, 624 pages, Addison Wesley, 2009.
- [Sibson 1973] Sibson, R., SLINK: an optimally efficient algorithm for the single-link cluster method, *The Computer Journal* (British Computer Society), 16 (1), pages 30-34, 1973.
- [Skupin 2004] Skupin, A., *The World of Geography: Visualizing a Knowledge Domain with Cartographic Means*, *Proceedings of the National Academy of Sciences*, Vol. 101 (Suppl. 1), pages 5274-5278, 2004.
- [Spence 2007] Robert Spence, *Information Visualization: Design for Interaction*, 2nd edition, 304 pages, Prentice Hall, 2007.
- [Tianamo 2008] Tianamo, search result visualization, <http://www.tianamo.com/>, last visited in January 2011.
- [Tochtermann et al. 2002] Tochtermann, K., Sabol, V., Kienreich, W., Granitzer, M., Becker, J., *Intelligent Maps and Information Landscapes:*

- Two new Approaches to support Search and Retrieval of Environmental Information Objects., Proceedings of the International Symposium on Environmental Informatics, Vienna Austria, 2002.
- [Tochtermann et al. 2003] Tochtermann, K., Sabol, V., Kienreich, W., Granitzer, M., Becker, J., Enhancing Environmental Search Engines with Information Landscapes, Proceedings of International Symposium on Environmental Software Systems, Semmering, Austria, 2003.
- [Tochtermann et al. 2005] Tochtermann, K., Granitzer, M., Sabol, V., Klieber, W., MISTRAL: Service Orientierte Cross-Media Techniken zur Extraktion von Semantik aus multimedia Daten und Deren Anwendung, in proceedings of Semantics 2005, Vienna, Trauner Verlag.
- [Tominski et al. 2004] Tominski, C., Abello, J. Schumann, H., Axes-based visualizations with radial layouts, Proceedings of the 2004 ACM symposium on Applied computing, pages 1242 - 1247, Cyprus, 2004.
- [Tou & Gonzales 1974] Julius T. Tou, Raphael C. Gonzales, Pattern Recognition Principles. Isodata Algorithm, in Chapter 3, Pattern Classification By Distance Functions, Reading, MA: Addison-Wesley. pp. 97-104, 1974.
- [Treisman 1985] Treisman, A., Preattentive processing in vision. Computer Vision, Graphics and Image Processing 31, pages 156177, 1985.
- [TROVE v2.1.0] TROVE High Performance Collections for Java v2.1.0, <http://trove.starlight-systems.com>, last visited in August 2011.
- [Tufte 1990] Edward R. Tufte, Envisioning Information, 4th Printing edition, 126 pages, Graphics Press, 1990.
- [TUG 2011] Graz University of Technology (Technische Universität Graz), Austria, <http://www.tugraz.at/>, last visited in January 2011.
- [Tukey 1977] Tukey, J.W., Exploratory Data Analysis. Addison-Wesley, Reading, Massachusetts, 1977.
- [van Ham et al. 2009] F. van Ham, F., Wattenberg, M., Vidas, F.B., Mapping Text with Phrase Nets, IEEE Transactions on Visualization and Computer Graphics, Volume 15 Issue 6, 2009, pages 1169 - 1176.

- [Voorhees 1986] Voorhees, E. M., Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval, Information Processing and Management, Volume 22 Issue 6, pages 465-76, 1986.
- [WinDirStat 2007] WinDirStat, Disk usage treemap visualization software, 2007, <http://windirstat.info/index.php>, retrieved in January 2011.
- [Wang et al. 1997] Wei Wang, Jiong Yang, Richard R. Muntz, STING: A Statistical Information Grid Approach to Spatial Data Mining, in Proceedings of the 23rd International Conference on Very Large Data Bases, pages 186-195, 1997.
- [Ware 2004] Colin Ware, Information Visualization - Perception for Design, 2nd edition, 486 pages, Morgan Kaufmann, 2004.
- [Ware 2008] Colin Ware, Visual Thinking: for Design, First Edition, 256 pages, Morgan Kaufmann, 2008.
- [Weber et al. 2001] M. Weber, M. Alexa, W. Müller, Visualizing Time-Series on Spirals, in proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01), pages 7 - 13, 2001.
- [Weitlaner 2009] Doris Weitlaner, Usability-Evaluierung von Visualisierungskomponenten zur temporal-thematischen Analyse von Textdokumentsätzen, Bakkalaureatsarbeit an der Technischen Universität Graz in Softwareentwicklung-Wirtschaft, Austria, 2009.
- [Onn et al. 2011] Onn, K. W., Sabol, V., Granitzer, M., Kienreich, V., Lucose, D, A Visual SOA-based Ontology Alignment Tool, poster, in Proceedings of the Sixth International Workshop on Ontology Matching (OM-2011), 2011.
- [Wise et al. 1995] Wise, J.A., Thomas, J.J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., Crow, V., Visualizing the non-visual: spatial analysis and interaction with information from text documents, Proceedings of the 1995 IEEE Symposium on Information Visualization, pages 51-58, 1995.
- [Wise 1999] James A. Wise, The ecological approach to text visualization, Journal of the American Society for Information Science - Special issue on integrating multiple overlapping metadata standards, Vol. 50(13), pages 1224-1233, 1999.

- [Wordnet 2011] WordNet lexical database of English Language, <http://wordnet.princeton.edu/>, retrieved in January 2011.
- [Xu et al. 2005] Xu, R., Wunsch, D. II, Chang, S.-F., Survey of clustering algorithms, *IEEE Transactions on Neural Networks*, Volume 16, Issue 3, Pages 645-678, May 2005.
- [Yang & Pedersen 1997] Yang, Y., Pedersen, J. O., A Comparative Study on Feature Selection in Text Categorization, *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.
- [York et al. 1995] York, J., Bohn, S., Pennock, K., Lantrip, D., Clustering and dimensionality reduction in SPIRE. In AIPA Steering Group (Eds.), *Proceedings of the Symposium on Advanced Intelligence Processing and Analysis*, 1995.
- [Zanasi 2005] A. Zanasi, *Text Mining And Its Applications To Intelligence, CRM and Knowledge Management*, WIT Press, 2005.
- [Zhang et al. 1996] Zhang, T., Ramakrishnan, R. and Livny, M., BIRCH: an efficient data clustering method for very large databases, In *Proceedings of the ACM SIGMOD Conference*, pages 103-114, 1996.
- [Zhao & Karypis 2002] Y. Zhao and G. Karypis, Evaluation of hierarchical clustering algorithms for document datasets, in *Proceedings of the eleventh international conference on Information and knowledge management (CIKM)*, McLean, Virginia, pp. 515-524, 2002.