Casual Analytics: Advancing Interactive Visualization by Domain Knowledge

Von der Fakultät Informatik, Elektrotechnik und
Informationstechnik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Harald Bosch

aus Göppingen

Hauptberichter:	Prof. Dr. Thomas Ertl
Mitberichter:	Prof. Dr. Leo Wanner
Tag der mündlichen Prüfung:	23. 12. 2014

Institut für Visualisierung und Interaktive Systeme
der Universität Stuttgart

2014

Acknowledgements

The creation of this thesis was made possible by several people, who I want to credit. First of all, I thank my adviser Thomas Ertl for giving me the opportunity to pursue this degree, for his confidence in my abilities, for his ongoing support, and for creating a magnificent and unique work environment, the Institute for Visualization and Interactive Systems (VIS). It was a pleasure to work with my co-examiner Leo Wanner on the projects that funded this thesis, *PATExpert* and *PESCaDO*, and I thank him for all the time and effort he put into organizing them. With these projects, I was privileged to meet many friendly colleagues throughout Europe.

I thank the members of VIS and the Visualization Research Center (VISUS) for the great discussions, fruitful collaborations, all the help, and all the fun. Most notably, I am grateful for my *Büroabschnittsgefährten*, friends, co-authors, VAST contest collaborators, and proof-readers: Steffen Koch, Mark Giereth, Dennis Thom, Robert Krüger, Qi Han, Michael Wörner, Florian Heimerl, Guido Reina, Christoph Müller, and Alexandros Panagiotidis. Addtionally, I would like to thank Martin Falk for his typesetting template.

It was a pleasure to work with the students that I was allowed to supervise. Specifically, I want to thank Geoffrey-Alexeij Heinze and Stefan Wokusch for their contributions to the *PESCaDO* project as well as Edwin Püttmann and Dominik Jäckle for their additions to the social media analysis tool set.

Above all, I am grateful for my family, for their love and their support, for my wife, helping me through the hard times and sharing with me the good times, and for my daughter, who is the new center of my galaxy.

Stuttgart, Harald Bosch
January 2015

Contents

List of Figures

Chapter 4

Chapter 5

List of Tables

Chapter 3

Chapter 4

Chapter 5

List of Abbreviations and Acronyms

EC	European Commission
API	Application Programming Interface
HTML	Hypertext Markup Language
HCI	Human-Computer-Interaction
IEEE	The Institute of Electrical and Electronics Engineers
IPC	International Patent Classification
IR	Information Retrieval
JSON	JavaScript Object Notation
MCV	Multiple Coordinated Views
NLP	Natural Language Processing
OWL	Web Ontology Language
PESCaDO	Personalized Environmental Service Configuration and Delivery Orchestration
PDL	Problem Description Language—The part of PESCaDO's ontology that relates to the formulation of a problem for which users seek support
RDF	Resource Description Framework
RDFS	RDF-Schema
SVM	Support Vector Machine
tf-idf	term frequency – inverse document frequency
UI	User Interface
URI	Uniform Resource Identifier
VAST	Visual Analytics Science and Technology —Annual IEEE symposium and (since 2010) conference
VC'11	IEEE VAST Challenge 2011
W3C	World Wide Web Consortium
XML	Extensible Markup Language

Abstract

The often cited information explosion is not limited to volatile network traffic and massive multimedia capture data. Structured and high quality data from diverse fields of study become easily and freely available, too. This is due to crowd sourced data collections, better sharing infrastructure, or more generally speaking user generated content of the Web 2.0 and the popular transparency and open data movements. At the same time as data generation is shifting to everyday casual users, data analysis is often still reserved to large companies specialized in content analysis and distribution such as today's internet giants Amazon, Google, and Facebook. Here, fully automatic algorithms analyze metadata and content to infer interests and believes of their users and present only matching navigation suggestions and advertisements. Besides the problem of creating a *filter bubble*, in which users never see conflicting information due to the reinforcement nature of history based navigation suggestions, the use of fully automatic approaches has inherent problems, e.g. being unable to find the unexpected and adopt to changes, which lead to the introduction of the Visual Analytics (VA) agenda.

If users intend to perform their own analysis on the available data, they are often faced with either generic toolkits that cover a broad range of applicable domains and features or specialized VA systems that focus on one domain. Both are not suited to support casual users in their analysis as they don't match the users' goals and capabilities. The former tend to be complex and targeted to analysis professionals due to the large range of supported features and programmable visualization techniques. The latter trade general flexibility for improved ease of use and optimized interaction for a specific domain requirement. This work describes two approaches building on interactive visualization to reduce this gap between generic toolkits and domain-specific systems.

The first one builds upon the idea that most data relevant for casual users are collections of entities with attributes. This least common denominator is commonly employed in faceted browsing scenarios and filter/flow environments. Thinking in sets of entities is natural and allows for a very direct visual interaction with the analysis subject and it stands for a common ground for adding analysis functionality to domain-specific visualization software. Encapsulating the interaction with sets of entities into a filter/flow graph component can be used to record analysis steps and intermediate results into an explicit structure to support collaboration, reporting, and reuse of

filters and result sets. This generic analysis functionality is provided as a plugin-in component and was integrated into several domain-specific data visualization and analysis prototypes. This way, the plug-in benefits from the implicit domain knowledge of the host system (e.g. selection semantics and domain-specific visualization) while being used to structure and record the user's analysis process.

The second approach directly exploits encoded domain knowledge in order to help casual users interacting with very specific domain data. By observing the interrelations in the ontology, the user interface can automatically be adjusted to indicate problems with invalid user input and transform the system's output to explain its relation to the user. Here, the domain related visualizations are personalized and orchestrated for each user based on user profiles and ontology information.

In conclusion, this thesis introduces novel approaches at the boundary of generic analysis tools and their domain-specific context to extend the usage of visual analytics to casual users by exploiting domain knowledge for supporting analysis tasks, input validation, and personalized information visualization.

Zusammenfassung

Die oft zitierte Informationsexplosion beschränkt sich nicht auf vergängliche Kopien zum Zwecke der Datenübertragung und große Mengen an multimedialen Aufnahmen. Auch hochqualitative, strukturierte Daten aus diversen Forschungsrichtungen werden immer leichter und freier zugänglich. Das liegt hauptsächlich an gemeinschaftlich erzeugten Datensammlungen und einer besseren Infrastruktur zum Datenaustausch oder, allgemeiner formuliert, an den von Nutzern erzeugten Inhalten des „Web 2.0“ sowie Initiativen zur Erhöhung der Transparenz bei öffentlichen Daten. Während die Datenerzeugung vermehrt durch Gelegenheitsnutzer erfolgt, bleibt die Datenanalyse in der Hand von großen Unternehmen wie Amazon, Google und Facebook, um nur einige zu nennen, die auf die Analyse und die Bereitstellung von Inhalten spezialisiert sind. Hierbei kommen vollautomatische Algorithmen zum Einsatz, um Metadaten und Inhalte zu analysieren und die Interessen und Einstellungen der beteiligten Nutzer abzuleiten, um im Weiteren passende Empfehlungen und Werbeeinblendungen präsentieren zu können. Neben dem Problem, dadurch eine „Filterblase“ zu erzeugen, in welcher dem Nutzer – auf Grund des selbstverstärkenden Charakters einer auf dem bisherigen Verlauf basierenden Empfehlung – niemals seine Einstellung kontrastierende Informationen gezeigt werden, haben vollautomatische Ansätze inhärente Probleme dabei, unerwartete Erkenntnisse zu liefern oder sich einer Veränderung anzupassen. Diese Probleme führten zur Einführung der Forschungsrichtung Visual Analytics (VA).

Wenn normale Nutzer eigene Analysen auf den verfügbaren Daten betreiben wollen, sehen sie sich mit generischen Werkzeugen und derren Fülle an Funktionen und Einsatzmöglichkeiten und VA-Speziallösungen für einzelne Domänen konfrontiert. Beide Ansätze sind nicht geeignet, um Gelegenheitsnutzer bei ihrer Analyse zu unterstützen, da sie nicht zu den Zielen und den Fähigkeiten des Benutzers passen. Die Erstgenannten sprechen aufgrund ihres Funktionsumfangs und ihrer programmierbaren Visualisierungen eher Analyseexperten an. Die Letzteren tauschen allgemeine Mächtigkeit gegen leichtere Bedienbarkeit und eine an die Domäne angepasste Benutzung. Diese Arbeit beschreibt zwei Ansätze, um mit der Hilfe von interaktiven Visualisierungen die Lücke zwischen allgemeingültigen Werkzeugen und Speziallösungen aus einzelnen Domänen zu verringern.

Der erste Ansatz basiert auf der Idee, dass die meisten für die Allgemeinheit interessanten Datensätze auf Entitäten und Attribute reduziert werden

können. Dieser kleinste gemeinsame Nenner findet beim facettierten Browsen und bei Filter/Flow-Umgebungen bereits häufig Anwendung. In Mengen zu denken entspricht unserer alltäglichen Erfahrung und erlaubt einen sehr direkten, visuellen Zugang zu den Objekten der Analyse und ist daher eine gute Basis, um Analysefunktionalitäten umzusetzen.

Die Interaktion mit Entitätsmengen in eine Filter/Flow-basierte Graphstruktur zusammenzufassen, kann dazu genutzt werden, die Analysetätigkeit zentral aufzuzeichnen und dadurch die Kollaboration, Berichtserstellung und Wiederverwendung von Filtern und Zwischenergebnissen zu unterstützen. Diese generischen Analysefunktionen werden als Plug-In-Komponente angeboten, welche in einige domänenspezifische Datenanalyse- und Visualisierungsprototypen integriert wurde. Dadurch kann die Komponente von dem impliziten Fachwissen des Domänenwerkzeugs (z.B. durch semantisch aufgeladene Entitätswahl und spezialisierte Datenrepräsentation) profitieren, während sie zur Strukturierung und Aufzeichnung des Analyseprozesses des Benutzers herangezogen wird.

Der zweite Ansatz nutzt bereits kodiertes Domänenwissen in Form einer Ontologie, um dem Benutzer die Interaktion mit den sehr spezifischen Daten der Domäne zu erlauben. Durch die Betrachtung der Zusammenhänge zwischen Konzepten der Ontologie kann die Benutzungsoberfläche automatisch erweitert werden, um auf Probleme in der Benutzereingabe hinzuweisen und die Resultate des Systems bedarfsgerecht anzuzeigen. Dabei werden die domänenspezifischen Visualisierungen für jeden Benutzer aufgrund seines Profils und der ontologischen Informationen personalisiert und orchestriert.

Zusammenfassend beschreibt die Arbeit damit neue Ansätze an der Schnittstelle zwischen generischen Analysewerkzeugen und dem durch den Anwendungsfall vorgegebenen Kontext. Dies erlaubt auch Gelegenheitsnutzern die Vorgehensweise der Visuellen Analyse zur Verfügung zu stellen, indem Analyseaufgaben unterstützt, Eingaben validiert und Informationsdarstellungen personalisiert werden.

Chapter 1

Introduction and Motivation

The often cited *information explosion* [Gantz and Reinsel, 2011] is still ongoing, increasing its pace, and is not limited to volatile network traffic, such as video streaming. Structured data is captured from a multitude of sensors such as videos and photo cameras, GPS trackers, privately run weather stations, or any other type of embedded data logger. Making these data publicly available is facilitated by cloud based infrastructures and low cost embedded devices providing, e.g., web services or sharing capabilities. Movements such as the open data initiative support the publication of public data, which leads to a broad availability of government supplied information and structured data archives. Finally, casual internet users themselves generate huge amounts of unstructured information in the sense of unparsed textual content, e.g. as messages, product reviews, public posts within online social media communities, and other commentary. This means that enormous amounts of data and embedded information are not only generated and transferred but also made available to the interested public.

At the same time as data generation is shifted from professional sources to everyday casual users, methods for data analysis are often reserved to large companies specialized in content analysis and distribution such as Amazon, Google, and Facebook, to name just a few. Here, fully automatic algorithms analyze metadata and content to infer interests and believes of their users and present only matching navigation suggestions and advertisements in order to provoke high click-through rates. Due to the positive reinforcement effect that

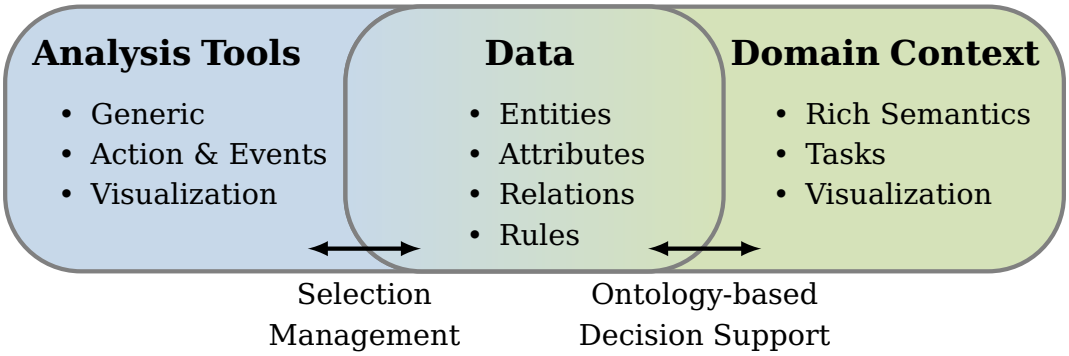


Figure 1.1 — The context of casual analytics consists of generic analysis techniques and the domain-dependent tasks and semantics. This thesis introduces two techniques to increase the overlap between these two areas and thus provide analysis capabilities to support domain tasks (Selection Management) and infer domain semantic to support decisions of lay users in a complex domain (Environmental Decision Support).

current navigation suggestions will probably be the future browsing history, this can lead to the *filter bubble* effect [Pariser, 2011] where users rarely see information conflicting with their point of view. This situation introduces two linked problems. The increased availability and visibility of data has created a public demand for analysis, while the analysis expertise is tied to the service providers. But even if access to the required functionality would be present, fully automatic algorithms are not suitable for serving varying interests and goals for immense data sets. The relatively young research field of visual analytics provides techniques for integrating human capabilities and automated methods for processing massive data by using interactive information visualization. This allows for supporting vague analysis goals (such as finding something unexpected) and interactively working with both immense and unstructured data sets.

Today's users seek information for supporting many day-to-day decisions. The user group may vary from professional users, over enthusiasts spending large amounts of time analyzing data sets about their hobby, to customers researching product specifications and reviews. On the one hand, domain-specific solutions may lack the analysis capabilities to efficiently support decisions in the presence of large, unstructured data sets and incomplete task definitions. On the other hand, many generic visual analytics solutions are targeted at analysis experts that can program and parametrize the desired

visualizations and data transformations in order to complete their tasks.

Figure 1.1 shows the context of such an analysis task and will be explained using an example from the intellectual property domain. A task such as ‘perform a freedom-to-operate search for a new patent’ has a domain-specific meaning and has to be translated by an expert into several subtasks depending on the scope of the patent application. Similarly, domain knowledge is needed to interpret the results of subtasks, e.g. retrieved sets of related patent documents. Here, the analysis may use domain-specific visualization that matches the nature of the involved entities (e.g. documents, inventors, categories). On the other side of the spectrum are generic analysis actions such as ‘perform query’ or ‘highlight selection’. Their intended outcome is independent of the domain in which they are executed. Visualizations can still be involved but are limited to general purpose approaches such as scatter plots that need to be configured accordingly to provide value to the analysis. These two areas communicate through the data which shares aspects with both areas. On the one hand, entities, attributes, and relations are generic items that can be filtered, sorted, and processed without interpreting their meaning. On the other hand, they bear meaning and interpreting the data is infeasible without domain knowledge.

1.1 Research Questions

In context of this work the following research questions arise:

- Can interactive visualization mitigate the conflict between tool generality and domain-specific needs?
- How can complex data and relations be analyzed and visualized appropriately to lay users?
- How to automatically support users in querying domain-specific computation modules?

1.2 Contribution and Structure

In this thesis two approaches are presented at the boundaries between generic analysis tools and domain context (see Figure 1.1). The first approach is a filter/flow-based selection management component. Abstracting from the meaning of domain entities, this component can be introduced into existing domain-specific data visualization systems [Giereth et al., 2008b,a; Koch

and Bosch, 2011; Chae et al., 2012] to provide generic analysis functionality. Otherwise volatile selections or filter definitions from the visualization system can be persisted in a graph structure and combined freely to formulate complex selection criteria. In this thesis, it is shown that such an approach can enhance existing systems' analysis capabilities with visual analytics aspects such as reusing selections for explorative analysis and query building [Koch et al., 2009, 2011], collaboration and reporting [Bosch et al., 2011b], analysis provenance [Bosch et al., 2009], and filter orchestration for real-time data streams [Bosch et al., 2013; Andrienko et al., 2013; Thom et al., 2012b,a, 2014]. Relying on the host environment for domain-dependent data interpretation, visualization, and filter configuration, the filter/flow selection management encapsulates analysis capabilities in a 'plug-in' fashion.

The second approach mitigates the users' missing familiarity with a domain by exploiting formally represented domain knowledge [Bosch et al., 2011a, 2012]. Its prototype is integrated in an environmental decision support system [Wanner et al., 2010, 2011, 2012a,b, 2013]. It consists of two separate areas involving user interaction: query creation and result interpretation. As interdependencies between query parameters may be complex and not obvious to a new user unfamiliar with the system, problematic input that might create errors during execution must be prohibited. An automatic algorithm extracts these interdependencies from the domain ontology to construct a set of generic rules for the input elements of the query construction form. The rules can then be used to support users in their tasks. Further, a web-based environmental data visualization framework for the interpretation of the system's result is introduced. Interpreting domain-related data measurements and forecasts is supported by translating them into subjective ranges for the lay user. The framework is capable of orchestrating several data visualization types for concurrently displaying multiple relevant information sources as well as personalizing the visualization parameters to allow for individual preferences and sensitivities.

After an introduction into the research areas and vocabulary most related to this thesis in Chapter 2, the filter/flow selection management approach, its visual analytics capabilities, and several application domains are presented in Chapter 3. The two areas of exploiting domain knowledge to support the query construction and result interpretation of lay users is presented in Chapter 4. Both approaches have been evaluated in several ways. Chapter 5 presents the evaluation effort [Heimerl et al., 2012; Vrochidis et al., 2012] and discusses their results as well as the general outcome of the thesis. An outlook on possible future work and the applicability of the approaches for

future systems is given in the final Chapter 6.

Beyond the scope of this thesis, work has been published concerning: the exploitation of semantic annotation for intra-document navigation using fisheye distortions [Giereth et al., 2008c]; a focus+context technique for inspecting and interacting with bundled edge collections in graph structures [Panagiotidis et al., 2011]; situation awareness for network traffic anomalies [Krüger et al., 2012a]; a predictive visual analytics approach for forecasting box office success and user ratings for movies based on social media messages and metadata [Krüger et al., 2013a]; a focus+context technique for filtering and exploring trajectory data [Krüger et al., 2012b, 2013b]; and the aggregation of movement patterns from social media users for crisis response scenarios [Jäckle et al., 2013].

Chapter 2

Foundations

The systems, prototypes, and approaches presented in this thesis are concerned with data analysis, generating insights from data, as well as supporting decisions. For this purpose, most of them employ *visual analytics* approaches. Coined as a term by Wong and Thomas [2004], "visual analytics is the formation of abstract visual metaphors in combination with a human information discourse (interaction) that enables detection of the expected and discovery of the unexpected within massive, dynamically changing information spaces." From this definition one can see that the relatively young field of visual analytics is a multidisciplinary research area connecting visualization, human-computer interaction, and machine learning. This combination shall allow processing massive data while harnessing the human's broad visual channel to detect patterns and steer the analysis through interaction. The definition also nicely characterizes problems that are suitable for a visual analytics approach. The data sets are typically too large to be handled manually by human effort alone but they are also dynamically changing. In particular, the target of the analysis is often unclear ("discover the unexpected"), which rules out a purely automatic processing. In these cases, the tight integration of all three disciplines can allow for an exploration and analysis of the data which is supported by automatic methods such as machine learning.

Of course, systems developed prior to the definition of the visual analytics term already combined parts of these aspects and investigated ways to address these kinds of problems. Keim [2002] stated that "future work

will involve the tight integration of visualization techniques with traditional techniques from such disciplines as statistics, machine learning, operations research, and simulation." Nevertheless, with the definition of the 'Visual Analytics Research and Development Agenda' by Thomas and Cook [2005], a proliferation of the research area took these ideas to a broad spectrum of application domains and instilled new interest in the examination of the analytic process. Thomas and Kielman [2009] defined a list of current challenges for visual analytics and identified, among others, the need to bring visual analytics approaches to individual or personal uses. This is one motivation of this thesis.

In order to understand how data analysis can be supported, one needs to have an understanding of the process of transforming raw data into a presentable insight, i.e. the product of the analysis. Therefore, the following section will provide views on this process and its activities. Visual analytics is by definition a data-driven approach and scalable user interfaces have to allow the exploration of large data sets. This often leads to Multiple Coordinated Views (MCV) environments providing ways to interact with the data. These environments are detailed in Section 2.2. One of the presented approaches is based on a filter/flow graph structure and thus, a selection of graph based data analysis tools are examined in Section 2.3. Finally, the basics of Semantic Web technology are briefly introduced in Section 2.4 in order to lay the foundation for Chapter 4.

2.1 The Software-Supported Analytic Process

Deriving knowledge from data is a complex process involving many factors such as the data's nature and representation, the analyst's previous knowledge and goals, as well as the interaction between the data and the analyst. Several works have examined how analysts interact with the data and the analysis system. The subset that will be presented in the following focuses on different aspects and thus these approaches complement each other to provide a broad view on the software-supported analysis process.

2.1.1 Information Foraging & Sensemaking

Pirolli and Card [2005] performed cognitive task analyses and think-aloud studies with several (business) intelligence analysts. They determined that many forms of intelligence analysis are 'sensemaking' tasks, consisting of

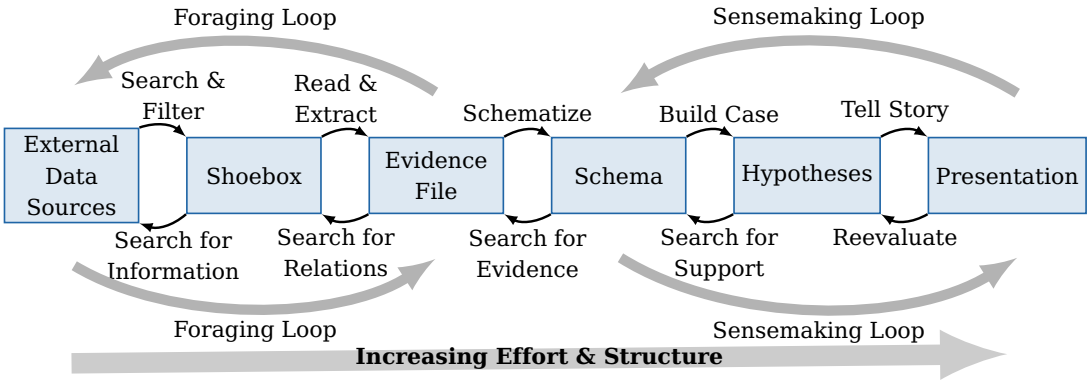


Figure 2.1 — Data and process flow of the sensemaking model by Pirolli and Card [2005].

gathering information, representing it in a schema, developing insights through the modification of the representations, and creating a knowledge product, such as reports or decisions. Based on their observations, they have presented a notional model, whose terminology is frequently used in the visual analytics community to describe which parts of the analysis process are supported by a system. The key elements of the model are depicted in Figure 2.1.

Boxes depict data and arrows represent process flows. The large number of loops and possible transitions emphasize the dynamic nature of an analysis. An example workflow through this model, going from source to presentation, may be observed as follows. The analyst has access to a data source and searches for relevant documents that are stored for further reference and processing (the ‘shoebox’). Single statements or larger parts of these documents are extracted into evidence files and schematized into a structure that allows for perceiving relations (e.g. timelines, mind maps, story boards). A theory, supported by the schemata, is built and the final product is presented. At any stage, the direction may be reversed, e.g. in order to search for additional support for a theory or evidence invalidating it. Here, it is interesting to note that the reversed process flow is almost entirely constructed from search activities targeted at different intermediate analysis products.

Two basic loops separate the activities into a foraging phase and a sensemaking phase. The foraging loop is focused on searching and extracting bits of information that are relevant for the analysis. Here, the analyst tries to go from large data volumes of uncertain relevance to highly relevant information in multiple steps to allow for investing the limited resources such

as effort and time efficiently on promising excerpts. The sensemaking loop is building upon the collected evidence to derive structure and knowledge. While the evidence file grows in size linearly with every added document, the complexity of the schemata is based on the relations between them and thus grows at least quadratically. Analysis tools supporting sensemaking can therefore provide ways to externalize thought processes and human memory onto visual displays.

In this context, the proposed selection management approach (Chapter 3) can be seen primarily as a sensemaking-supporting tool that externalizes hypothesis definitions and can semi-automatically create presentations of the result. However, it also is a way to have several shoeboxes because each node of the graph represents a collection of entire documents. For the representation along timelines or geographical features the approach relies on the domain-specific visualization environment in which it is embedded. Here, it should be noted that the model does not perfectly match every analysis approach. Microblog messages, used as an example in Sections 3.2.1 and 3.5, are very short and thus condensed to a single statement, which renders further extraction of statements into evidence files unnecessary.

2.1.2 Generality and Domain Dependence

An alternative view, especially on the visual analytic processes, is presented by Gotz and Zhou [2009], focusing on the interaction between human and software. They defined four levels of activities with decreasing semantic value: task → sub-task → action → event, with the intention of capturing and documenting the analysis process on a semantically rich level. Events are basic user interactions such as clicking, dragging, and typing. On their own, they have little to no semantic value and a recorded sequence of them may be used to replay an analysis, but not to capture its intention. However, small sequences of events can be identified as actions. Double clicking a map display may constitute a zoom action. Often, there are multiple possible event sequences for a single action. Clicking and dragging a zoom slider may be an alternative representation of a zoom action. Actions are the central level for recording an analysis as they contain the user's intention and thus have more semantic value. Zooming a view shall change the display of a view and provide more details. However, they are still generic enough to find the same action sets in visual analytics approaches for different domains. Actions are taken to answer subtasks and finally tasks. Both have a clear relation to the application domain and rich semantics. The overall tasks drive the analysis

but may not be directly translatable into actions to fulfill them. Therefore, they are divided into smaller subtasks following a divide-and-conquer approach. The distinction between domain-related semantic tasks and generic analysis actions is integrated into the casual analytics model described in Figure 1.1.

With actions as the central entity, Gotz and Zhou have collected and classified actions from several visual analytics solutions into three areas. *Exploration* actions are mostly related to the foraging loop of Pirolli and Card [2005] and are composed of *data exploration* actions such as filtering, inspecting and querying a data item or data source, as well as *visual exploration* actions that change the representation of the data such as zooming, panning, and brushing. The second category consists of *insight* related actions that could be matched to sensemaking loops as they structure the analysis outcome by annotations, bookmarking, and managing free-form notes. Lastly, *meta* actions concern the history of recorded actions used for, e.g., undo and redo operations.

2.1.3 Queries

Queries assume a very prominent role in both models either as top-down search for supporting sensemaking or as data exploration actions. The intention for executing a query may vary depending on the situation.

Jansena et al. [2008] have defined a classification of user intent for web search queries consisting of *navigational*, *transactional*, and *informational* searching. While navigational searches (i.e. the targeted resource is known and only has to be found) may also occur in an analysis context, transactional searches (i.e. searches targeted at obtaining, booking, or buying something) are exclusive to web searches. The third category, however, corresponds to queries employed in the analysis process. The search is either *directed*, in the sense of finding an evidence or fact to (dis)prove a hypothesis, or *undirected*. In the latter case, a search scope is approximated to narrow down the data collection in order to monitor and explore it. For instance, searching for the epicenter of an earthquake is a directed search, while searching for information about the earthquake without further specifying which kind (e.g. its strength, damage, power outages) is an undirected search. In combination with the intent to find and list relevant information, the latter constitutes the category of *exploratory search*.

Boolean Retrieval

Perpendicular to the search intent, several mechanisms exist to formulate and process a query in order to find the desired information. The most prominent techniques are Boolean retrieval and the vector space model. Boolean retrieval queries consist of search fragments, such as single terms that have to appear in the desired document, and Boolean operators to define the combination logic of these fragments. To search for flooding related documents one could use the query "`flood OR (heavy AND rain)`". The benefit of Boolean retrieval is that both the evaluation of each search fragment and their combination are very comprehensive concepts and thus the reason why a document is an element of the result set can be explained easily. The ability to shape result sets iteratively by small additions to the query is the reason why the Boolean retrieval model is still very common in domains that require finding every related document, e.g. patent and legal searches.

Vector Space Model Search

One drawback of Boolean retrieval is the inability to rank results according to their similarity to a query. By definition, documents are either relevant or not, but not to a certain degree. Contrarily, the vector space model estimates the importance of each term for a document and can thus rank the results according to the search terms. In the simplest case this is done by counting the term's frequency in the document ($tf_{t,d}$). Each term constitutes its own dimension in the vector space model and a document is represented as a vector of term frequencies. Similarly, the query can be perceived as a new document vector and a distance to each other vector can be computed using cosine similarity [Manning et al., 2008, chap. 6]. This similarity can then be used to rank the result set and truncate it at a certain threshold. This also allows a query-by-example approach by computing the similarity to an existing document. Using only term frequencies, however, suffers from low discriminative power concerning common words that are frequent in all documents. Therefore the weighting schema is often a combination of term frequencies and the inverse document frequencies $idf_t : log(N/df_t)$. Here, N is the size of the document collection to normalize the number of documents (df_t) that contain the term t. This constitutes the term frequency – inverse document frequency (tf-idf) measure ($tf_{t,d} \cdot idf_t$).

Query Performance Measures

As already indicated, different tasks have different requirements on the completeness and accuracy of the result set. While it is enough to find a single evidence to disprove a hypothesis, the opposite task (being confident that there is no such disproving evidence) requires finding any closely related information. This difference is named according to which of the evaluation measures used to evaluate the performance of a query should be optimized to support the task. Based upon the information need, every data collection can be divided into relevant and irrelevant documents. A search result set has an optimum *recall* if it contains all relevant documents and any number of irrelevant ones. It has an optimum *precision* if it only contains relevant documents, potentially missing many of them. This can be formalized as:

$$\text{recall} = \frac{\text{number of relevant documents in the result set}}{\text{total number of relevant documents}}$$

$$\text{precision} = \frac{\text{number of relevant documents in the result set}}{\text{total number of documents in the result set}}$$

Both measures can be optimized trivially when they are regarded in isolation (by returning either all documents or only the most promising one). Therefore, two additional measures are used in the evaluation chapter of this thesis: F_1-score and accuracy. The first one is the harmonic mean of recall and precision, and the latter one is the percentage of correct decisions. Concerning search queries it would translate to

$$\text{accuracy} = \frac{\text{included relevant documents} + \text{omitted irrelevant documents}}{\text{total number of documents}}$$

Scatter/Gather Foraging

The search techniques presented so far rely on the user's ability to formulate a query to describe their information need. This ability may not always be present. Several tasks in the visual analytics domain concern the search for the unexpected. Additionally, users unfamiliar with a domain's vocabulary might use the wrong terms for their initial query and have no possibility to rectify this problem in the further course of the analysis.

The *Scatter/Gather browsing* approach [Cutting et al., 1992] is based on clustering, an unsupervised learning approach which does not need any initial user input. The information retrieval of a scatter/gather approach is an alternating sequence of clustering (scatter) and selecting potentially relevant clusters (gathering). The selected cluster are then unified and establish the

new basis for another scattering step. This is repeated until the information need of the analyst is met. In order to be applicable to interactive information retrieval, the scatter/gather approach requires fast and scalable algorithms for clustering as well as summarizing the contents of a cluster. The latter is required to facilitate the quick selection of clusters for the next iteration.

Several approaches of this thesis are related to the scatter/gather idea. A statistically motivated keyword filter creation tool (Section 3.5.2) iteratively updates the document collection used to suggest new keywords for inclusion into the filter. The graph-based selection management component presented in Chapter 3 can 'scatter' a result set by using several filters and collect individual sets as needed by combining them again.

2.2 Data-Driven Analysis Environments

Visual analytics systems for analyzing large data collections most often are *visual data exploration* [Keim, 2002] tool sets. They include the human creativity and perception capabilities for more efficient data mining. Therefore, these tools have to provide visual forms that allow analysts to interpret large amounts of data in a suitable, efficient way. The *Information Seeking Mantra* by Shneiderman [1996] emphasizes the important aspects of such an interaction process in order to be efficient:

> Overview first, zoom and filter, then details-on-demand

Automatic methods are employed to aggregate the data and derive further attributes to allow for more meaningful overviews (e.g. clustering or topic and sentiment analysis for better aggregation scopes). The analysis continues by zooming into identified patterns of interest and drilling down into subsets by using filter operations. Finally, details of individual entities may be accessed. Especially if the goal of an analysis is not yet clearly defined, alternative overviews can be provided for a free exploration. Here, multiple coordinated views environments (MCVs) can help to have the necessary overview and detail views available to the analyst simultaneously.

This section will introduce briefly the reference model of Information Visualization used to generate the individual views of an MCV environment, the terminology of MCVs, and ways to provide details-on-demand.

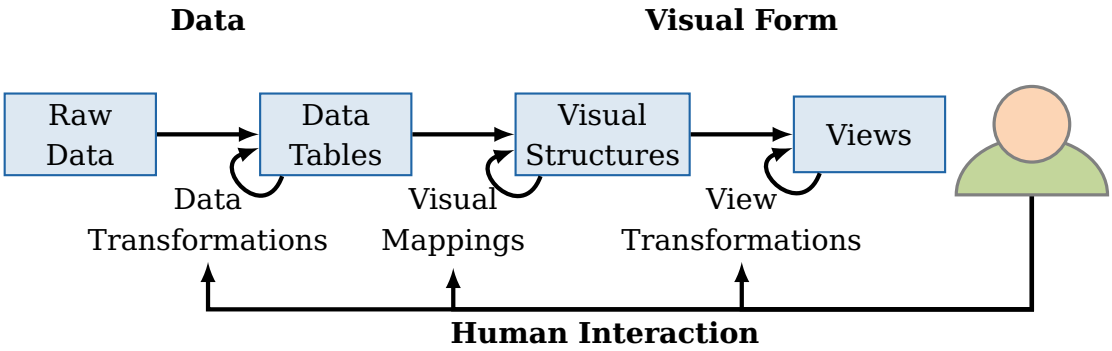


Figure 2.2 — Information Visualization Reference Model according to Card et al. [1999].

2.2.1 Information Visualization Reference Model

Analog to the foraging & sensmaking loops with their sequence from data source to analysis product, the information visualization reference model by Card et al. [1999] characterizes visualization as a mapping from data to a visual form in multiple steps (see Figure 2.2). Each arrow indicates one or more transformations. The central 'Visual Mappings' from data tables to visual structures define the design of a visualization. Here, attributes of the entities to be displayed are mapped to spatial (position and size) and graphical (shape, color, pattern, etc.) properties that constitute the visualization. Data transformations are essential for information visualization dealing with sources of abstract data,[1] such as text documents, from which representable entities (e.g. key terms or involved persons) and their attributes and relations first have to be derived. View transformations create views from visual structures by defining view parameters such as the position, rotation, scale, and clipping frame of a view port.

In interactive visualization, the observing human can influence each of these transformations to, e.g., change the view port by panning and zooming, change the visualization design by adding a color highlight, or change the data transformation with filter operations. Koch [2012] has extended the reference model to include additional facets of multiple coordinated views (branching additional views on the same data) and visual analytics (provenance of human interaction and semi-/automatic processing).

Alternative and more conceptual models exist, e.g. in the form of the

[1] In contrast to scientific visualization usually concerned with source data that has an inherent spatial domain.

Data State Model by Chi [2000] and the Visual Analytics Process by Keim et al. [2008]. However, the model by Card et al. and its extension can be directly used for the structure of software frameworks (e.g. prefuse[2]) as well as structuring collaborative interaction using explicit branching [Tobiasz et al., 2009].

2.2.2 Multiple Coordinated Views

Multiple Coordinated Views (MCV) have already been introduced as a suitable environment for exploratory visualization and most of the prototypes presented in this thesis are MCVs. The following brief introduction is based on the state of the art report of Roberts [2007].

The term 'multiple views' describes any instance where data is presented using multiple windows. This can take the form of (*a*) using different visualization designs on the same data where each is suited to see different kinds of patterns (e.g. adjacency matrix and node-link diagram of a graph); (*b*) showing different attributes of a data set's entities (e.g. timeline showing temporal attributes and a map showing spatial data); or (*c*) showing the same visualization but for different data sets allowing an easy comparison.

Additional views are either chosen from a fixed set of predefined views or are dynamically generated. The latter approach is more common in generic and multipurpose analysis systems where the suitable views cannot be defined a priori. *Improvise* [Weaver, 2004], for instance, uses a visual abstraction language to program new views. In other scenarios, the developer may choose to create new views when the parameters of the transformations from the information visualization reference model change. Roberts describes three options when reacting on parameter changes: replace, replicate, overlay. Replacement exchanges the contents of a view. Replication creates a new view with the updated parameters, retaining the old configuration in the existing view. Overlaying embeds the new configuration in an existing view. This is the standard method to draw the analyst's attention to a certain part of a view by highlighting the related data points. The developer of a system has to find a balance between these approaches by trading limited screen space for a spatial history of the analysis. The *ExPlates* system [Javed and Elmqvist, 2013] (see also Section 2.3) distinguishes *mutating operations* that change the underlying data of a visualization, and *invariant operations* that

[2] `http://prefuse.org/doc/manual/introduction/structure/`

change only the view port or formatting. The former uses replication and the latter replacement.

The term ‘coordination’ refers to coordinating a user interaction between different views, preferably all views. The interaction may be indirect in the sense of adjusting a filter value on a slider which hides filtered items from the coordinated views. It can be direct in the sense of selecting an item or a range of items in one view which replicates the action to the coordinated views. In the latter approach the selection action is called *brushing* [Becker and Cleveland, 1987] and the coordination is named *linking*. Supporting brushing & linking in domains with big data sets introduces a scalability requirement on the employed filter mechanisms. Any user interaction should cause a response in a limited amount of time, providing intermediate results if necessarily. Depending on the intended audience, visualization, and action, the time limit for being considered interactive varies. Brushing and panning operations that require a constant updates to identify the next desired position have higher requirements as executing a complex filter operations. To maintain interactivity for brushing operations, optimized data structures are used, e.g. based on grids or binned aggregation [Liu et al., 2013].

Coordination is not limited to filters and highlighting, two spatialization views may also be linked to always present the same view port. Dedicated ‘detail views’ may present additional information or the raw data element of the current visual structure under the cursor.

2.2.3 Details-on-Demand

Dedicated detail views and zooming interaction are two ways to access details on demand. In a review of methods to provide details in the context of overview information Cockburn et al. [2008] structure their work into four areas:

The *overview+detail* approach spatially separates the detail information from its contextual overview. Detail views and overview insets (regularly used in maps, showing a zoomed out view of the surrounding of the current location) fall under this category.

Zooming uses a temporal separation replacing the overview with detail view about one part of it. This conserves the limited screen space but hinders the analyst to ‘quickly glance’ on the overview for reference. One distinguishes structural, sometimes semantic, zooming from plain graphical zooming. While the latter only renders the view in a finer resolution, the former changes the visual representation of the elements to use the additional

space for detail information. For instance it changes the visibility of otherwise hidden features such as small streets that are only visible while zoomed in.

Focus+context approaches reduce or even eliminate the separation between detail and context information by embedding the structurally zoomed-in view within its un-zoomed context. Most notably, the works of Furnas [1986] use the optical distortion of 'fisheye' lenses for a seamless embedding. This lens metaphor is often used for focus+context approaches and Bier et al. [1993] propose *Magic Lenses* as overlays that also change the interaction for the focused region.

The last category are *cue-based* techniques that indicate the focused items by changing their representation or provide necessary context information by including abstract representations of the surrounding into the detail view. These approaches are often used for mobile devices where the limited screen space makes distortions and additional views infeasible. Baudisch and Rosenholtz [2003] present a technique to show the direction and distance to the nearest points of interest by arcs at the display border. They are created by circles around the point of interest with a radius large enough to enter slightly the display area. This way the direction and distance can be perceived through the position and curvature of the arc.

The prototypes presented in this thesis utilize all of these detail-on-demand techniques.

2.3 Graph-based Interactive Analysis

Typical visual analytics application domains are concerned with entities, their attributes, and their relations, e.g. documents with similarity measures, persons involved in an event, co-authorship of publications. Graph-based interaction therefore suggests itself for the analysis for two purposes. First, to model the entity relationships as a graph structure and using visualization to explore the data sets. The work of Munzner et al., e.g. *TreeJuxtaposer* [Munzner et al., 2003] and *GrouseFlocks* [Archambault et al., 2008]) for exploring alternative hierarchies, as well as Delest et al. [2006] and the *Tulip* framework [Delest et al., 2004] are prominent examples. Second, and more related to this thesis are filter/flow-based query creation interfaces [Young and Shneiderman, 1993] that use the metaphor of water running through pipes and filters as an intuitive way to formulate Boolean queries. Filters in a linear sequence represent AND combined phrases and the flow passing through them is further reduces by each filter. ORs are constructed either by specifying

alternative attribute values within one filter or by arranging them in a parallel path. Negation is handled by inverting the selection of attribute values in a filter. The branches are always merged into the stream again which results into a single output.

More recently, Haag et al. have extended the classical filter/flow metaphor [2013] and applied it to the creation of SPARQL[3] queries [2014]. They allow multiple drains in the graph structure creating different result sets and provide more abstract filter types covering often needed constraints, such as splitting the flow with a filter definition instead of reducing it, allowing splitting by number ranges, and including sequences into a branching node. This greatly reduces the node and edge count and thus the visual clutter for complex queries. During a study of the concept, the participants had to manually evaluate traditional and condensed query graphs and it could be shown that the condensed graphs could be interpreted at least as well as the traditional variant.

The *ExPlates* system [Javed and Elmqvist, 2013] is closely related to the filter/flow approach presented in this thesis. It shares a graph structure, dedicated join nodes, and a canvas for spatializing the analysis progress. However, the presence of visualization nodes and different entities establishes a different usage and interaction scenario. *ExPlates* is a standalone application that can draw structured data from data base connections and provides each column as a stream of data, linked by a common row index. Each plate can have multiple input and output connections, e.g. a scatter plot node takes two numeric inputs as coordinate source. Therefore, the users can configure visualization nodes with the graph structure, similar to visual programming environments. Whenever the configuration of a view is changed, the old one is not replaced, but a new plate is created. As described in the previous section, view transformations such as panning and zooming are excluded. This creates a history of the analysis session which also can be manually annotated with drawings and labels.

FindFlow [Hansaki et al., 2006] also shares the principle filter/flow approach but has less constraints on the structure, e.g. allowing multiple drains. It has no explicit join nodes but nodes can receive multiple incoming connections, thus functioning as a union. Each edge can take a filter definition and, after choosing the targeted attribute, a small histogram guides the selection of the filter parameter. Intermediate results are presented as textual lists in each node with the possibility to access details of an entry by selecting it.

[3] A Semantic Web query language: `http://www.w3.org/TR/rdf-sparql-query/`

Similar, to *ExPlates*, *FindFlow* is designed as a standalone application.

gFacet [Heim et al., 2008] is a hybrid approach to filter Semantic Web data, having a graph structure, with a graph-based filter editor. From a central set of entities to be filtered, the user can create facets on the data by following a semantic relation to another set of entities. For instance one can filter a list of cities using the information which music band was founded there. This can be repeated for all known relations originating from the central entity, as well as from the facets, e.g., filtering the bands based on their music genre. The structure thus resembles the snowflake schema of OLAP[4] approaches for data warehouses [Chaudhuri and Dayal, 1997].

2.4 Semantic Web Basics

Parts of this thesis exploit codified domain knowledge to support users in formulating queries and interpreting results. This knowledge is presented and processed using Semantic Web technology which is based on the standards of the World Wide Web Consortium (W3C). The basic terminology is briefly introduced below, based on the textbook of Hitzler et al. [2008].

The World Wide Web as we see it is a vast resource of information based upon few technologies intended to present information to the human reader (HTTP, HTML, CSS). Understanding the content of web pages is an easy task for the reader but imposes a large effort for computers. Added services, such as web searching, are thus based on statistical features of the hyper-linked texts and avoid understanding the content. However, an understanding of the content is necessary for any task involving the meaning of a word, e.g. disambiguation of homographs and covering synonyms and hyperonyms.[5] Also, the heterogeneity of information presentation, which may vary from website to website poses a challenge for their automatic processing.

The goal of the Semantic Web agenda is to provide a standardized way to assign meaning to data and make them machine readable. This shall allow novel services and interoperable applications building on a semantically rich data base. The Semantic Web stack builds upon the Extensible Markup Language (XML) and defines two languages to define semantic resources and ontologies: RDF(S) and OWL.

---

[4] Online Analytical Processing

[5] There are, of course, statistical approaches for disambiguation, but they require enough context to infer the correct meaning.

The common syntax for the Semantic Web technology stack is XML which allows the standardized representation of the structures and facilitates the automatic parsing of documents. However, following the XML definition each document can define their own tags to describe the entities that it contains. Therefore, the problem of understanding the content is shifted to a problem of understanding its description.

2.4.1 Resource Description Framework

The Resource Description Framework (RDF) is an XML dialect that defines the way how documents describe their information. Each RDF document describes a directed graph using (subject, predicate, object)-triples. Each node of the graph is either a resource identified by a Uniform Resource Identifier (URI) or a literal. Literals are drains of the graph and cannot be the subject of a relation. They are used as links to atomic values that will not be described further, such as the name of a person or the value of a measurement. Using a graph structure allows for an easy integration of multiple RDF documents, connecting the individual graphs by merging nodes with the same identifier. Using triples, one can specify binary relations between entities. To specify higher order connections, a third node type is introduced. So called ‘blank’ nodes can be the target of a relation and then link to an arbitrary number of further resources. Contrary to other resources, blank nodes only have a document-specific identifier instead of an URI.

2.4.2 Describing Ontologies

While the structures and elements of documents are standardized by RDF, the semantic of employed predicates, such as `hasName`, is still opaque to the machine and cannot be put into relation with predicates that bear the same meaning, possibly from other languages. For this purpose, ontologies encode the taxonomy and relations of a domain, again into semantic structures.

RDF-Schema

RDF-Schema (RDFS) introduces meta-level structures such as classes, predicates, and literals to allow statements about these concepts in order to build taxonomies and typed relations. While RDF only allowed statements such as ‘Benjamin is a researcher’, RDFS can define that ‘researcher’ is a ‘class’ and, further, a subclass of ‘person’. At this level, new information

can be derived automatically through inference, such that Benjamin is also a `person`. However, RDFS has limitations that were deliberately designed to keep the logical relations simple enough to be computable. RDFS cannot model negated expressions or define the disjointness of two classes.

OWL

The Web Ontology Language (OWL) allows the modeling of more complex ontologies by providing high level predicates, e.g. for implicit class definitions. Further it allows to define cardinality for predicates and logical relations such as `unionOf`, `intersectionOf`, `complementOf`. However, this allows an expressiveness which can no longer guarantee decidability or complete reasoning. Therefore, three variants of OWL were introduced: Lite, DL, and Full, of which DL is the most frequently employed variant. DL stands for description logic and this version includes all language constructs of the unrestricted Full version, with certain restriction to guarantee the decidability.

Chapter 3

Selection Management – Domain-Adaptable Visual Analytics

The definition and management of sets is a crucial functionality for the interactive analysis of discrete information entities. They are an essential part of the foraging loops in the sense-making process described by Pirolli and Card [2005], where each *search*, *filter*, and *extraction* step creates a result set of entities. Having a persistent and explicit representation of these results is beneficial for many subsequent and related tasks. Most obviously, they can be used for contrasting them in order to evaluate hypothesis and for defining more complex sets for more sophisticated analysis questions.

In this chapter, an approach to enhance existing domain-specific visualization environments with explicit selection management is presented. The importance of such a management approach became obvious during the development of several visual analysis systems, covering multiple domains such as situational awareness from social media, information retrieval for intellectual property management, and social network analysis. All of these approaches form the background of this chapter. Here, the domain-specific tools featured rich interactive views but were missing the capabilities for inspecting subsets of the data, formulating hypotheses about their relations, and comparing query results for hypothesis validation, which was essential for the analysis.

The following sections will first introduce the basic idea and design of the selection management component and then discuss several specific problem

types from different application domains which can be supported by it, each time covering a generic aspect of visual analytics. Section 3.2 examines the benefits of selection management for the microblog analysis scenario of the IEEE VAST Challenge 2011 (VC'11). Section 3.3 extends the application to high recall problem domains such as patent search. Search query feedback loops as an tool for iterative query refinement are supported by integrating the approach into the *PatViz* patent search and analysis prototype. Section 3.4 discusses the benefits and incentives of explicit selection management for collaboration and reporting scenarios, also based on the VC'11 scenario. Section 3.5 showcases the applicability of the approach to real-time scenarios with streaming data, which has implications for the definitions of 'filters', 'sets' and 'selections'. Section 3.6 deals with uncertain set definitions while applying fuzzy rules of a social network pattern matching task featuring multiple entity types. Here, the filter/flow graph is used to structure and explore the combinations of different fuzzy rules.

3.1 Generic Filter-Flow Selection Management

Within the views of result set visualizations environments, traditional interaction techniques for selecting data subsets may not be sufficient. Especially if the selection of a particular group of entities requires separate subselections, posing constraints on multiple attributes at once may be difficult or even impossible. Therefore, this section proposes a selection management component in which selections can be persisted and combined. It has the form of an interactive canvas on which a filter graph of persisted selections is freely organized by the analyst. The filter graph has three basic types of nodes: *Set nodes* represent a set of entities. *Filter nodes* receive a set of entities as input and apply their individual filter function to create another set of entities as an output set node. The available filter functions depend on the application scenario. *Join nodes* combine multiple set nodes into one output node by applying a specific set operation such as a union, intersection, or symmetric difference. Employing symmetric operations simplifies the interaction because only the adjacency of nodes defines the combination result and not the order of the incoming sets. The initial graph consists of a single root set node containing the whole data set. Any other set note of the graph is derived either by adding a filter or a junction node to an existing set node. Edges depict the flow of entities through the filter graph. They link the filter and junction nodes to their respective input and result nodes. Therefore,

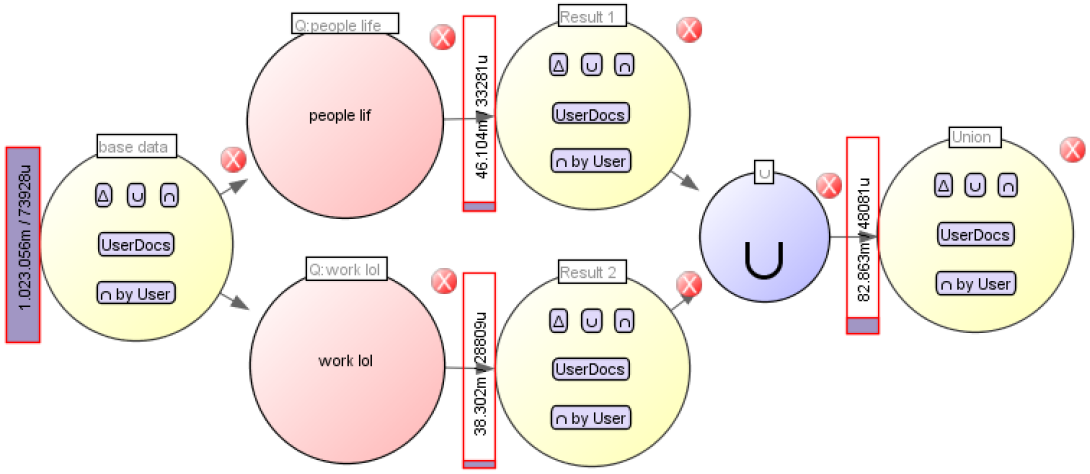
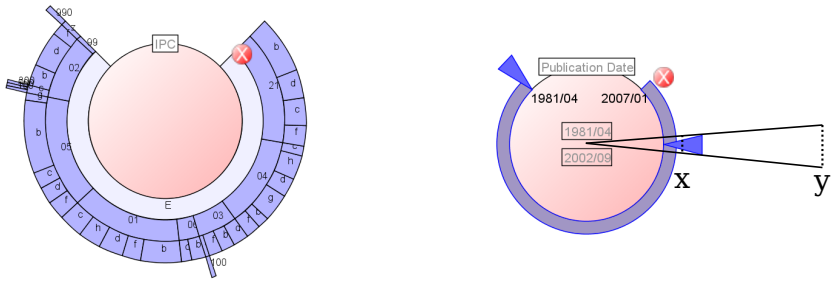


Figure 3.1 — The different node types of the selection management component: *set nodes* (yellow), *filter nodes* (red), and *join nodes* (blue). On the left hand side of each set node is the *load bar*, showing the amount of contained entities as a count and as fill level relative to the root node's data set. The widgets within the set nodes are frequently used node types that can be spawned as a child by dragging the widget into an open area of the canvas.

there is always at least a filter or junction between any two set nodes. Due to these characteristics, each producible filter graph is an acyclic, directed, and bipartite graph. Cycles would theoretically be possible but are prevented by disallowing the creation of additional edges that would create a closed walk. The semantic of the resulting graph is that all entities of a set node meet all filter definitions on a path from the root to its set node, or all paths in the case of intersection join nodes.

The three types of nodes are distinguished by their color, decoration, and size (see Figure 3.1). Set nodes have an additional *load bar* that indicates which percentage of the root node's data set arrived at this node. This allows for a quick assessment of the selectivity of a filter and size comparison of two sets. A label also contains the absolute count of contained entitites. Each node is itself a widget and contains labeled elements that can be dragged from within the node to a free spot on the canvas in order to create new filter or join nodes. An additional result set node for the output is automatically created as well. In the depicted example, these elements consist of the



(a) (b)

Figure 3.2 — Additional interactive widgets to configure filter nodes. (a) Sunburst widget to select elements from a hierarchy, (b) Range slider widget to select a date range. The dotted lines at location x and y illustrate a benefit of radial sliders: the user can move the mouse away from the node while dragging the slider control to increase the resolution of the adjustment.

three generic Boolean combination operators ∪, ∩, and Δ as well as two special purpose filters of the analysis system in which the component is embedded (UserDocs, ∩ by User). Nodes can be moved freely within the canvas and potential target nodes for new edges are highlighted while nodes are dragged. Dropping the dragged node on a highlighted one creates the new edge. Additional interactive elements can decorate a filter node depending on the parametrizability of the contained filter function (see Figure 3.2). For example, time filters can be adjusted with additional range sliders that are placed on the border of the node. Here, the circular shape of the node helps in changing the sensitivity of slider ad hoc. Mouse movements in the close vicinity of the node while dragging the slider result in bigger changes per pixel than when dragging the slider far away .

3.1.1 Design Considerations and Related Approaches

The design of the selection management component is influenced primarily by the guidelines and techniques presented by Shneiderman, Ahlberg, Young, et al., i.e., it can be seen as a way to integrate the principles of dynamic queries [Ahlberg et al., 1992], the filter/flow representation of Boolean queries [Young and Shneiderman, 1993], and direct manipulation [Shneiderman, 1982]. The load bar's filter-selectivity indication and overall immediate result update after each interaction is derived from *dynamic queries*. The

semantic of filters and flows is the same as in the filter/flow representation of Young & Shneiderman where the Boolean combinations of the filters are expressed by the graph structure. Filters organized in a sequence are linked with `AND` operations and filters in branches are linked with `OR` operations. In contrast to the filter/flow metaphor, our approach additionally allows the usage of explicit set operators in the joining nodes. These operators facilitate the combination of arbitrary sets of data objects, e.g. from otherwise unrelated branches, without the need to duplicate filter paths. Additionally, input nodes can be created by performing a normal selection operation in the specialized views of the domain-specific application. The overall interaction relies on dragging operations and interaction elements directly on the visual representation of filters and thus follows the *direct manipulation* style.

Besides these influences and the graph-based interactive analysis systems already described in Section 2.3, there are other approaches described in the literature on how to generate queries for different kinds of data storage systems visually. Spoerri as well as Jones et al. focus on the visualization of the Boolean combination of single search fragments. InfoCrystal [Spoerri, 1993] establishes icons that can show all possible relations between the search terms. The icons are derived from Venn diagrams and their complexity grows exponentially with the number of terms. From this power set, the users could choose the desired combinations for filtering the search results. While InfoCrystals can also be chained to form complex queries, their interpretation is not trivial. Similarly, Jones et al. [1999] allow users to formulate queries directly from Venn diagrams of result sets. Here, the users have to draw the diagrams and subsequently mark the regions that should constitute their final results, which can be laborious for complex analysis. Additionally, this technique relies on the principles of dynamic queries and bookmarking queries for later reuse. DataMeadow [Elmqvist et al., 2008] is a network-based approach to apply filters on multivariate data. Different aspects can be filtered at once using interactive visual metaphors called DataRoses. They unify the data display and filter configuration by visualizing the data in a star plot on which the user can brush range selections on each dimension. Because each node features essentially the same filter capability, building a graph of multiple nodes is mainly for the purpose of comparing different nodes and to allow filtering multiple ranges on the same dimension. A survey of several visual query systems for databases can be found in the work of Catarci et al. [1997]. It is structured by the representation style of the query and the result (form, diagram, icon, hybrid), the interaction strategy (top-down, browsing, schema simplification), and the query formulation (schema

navigation, subqueries, pattern matching, range selection).

In its initial design, the selection management component was intended to integrate query formulation, adaptation, and result visualization while relying only on the direct interaction described above. However, its integration into various applications for different domains has shown that there often is a need for the storage and recombination of otherwise volatile selections. Selections can be seen as intermediate results that are created while interacting with visual analytics systems through brushing and linking operations, detail-on-demands actions, or as results from computation. Therefore, the specialized views of the domain application are the primary source of filter definitions and result sets, which are then represented as nodes in the selection management component.

3.2 Managing Selections for Microblog Analysis

The IEEE VAST community arranges an annual challenge for advancing the field of visual analytics [Scholtz et al., 2012]. An artificial data set is provided in combination with an scenario for each challenge. Due to the synthetic nature of the data sets, a 'ground truth' can be guaranteed to be present within the data and the challenge can be used as a benchmark for visual analytics techniques and prototypes. This means that there is information relating to the scenario and challenge tasks hidden in a vast amount of data. The 2011 VAST Challenge [Grinstein et al., 2011] data set consisted primarily of microblog messages and by using the filter/flow based selection management to organize messages and their authors we could efficiently solve the tasks for our challenge entry [Bosch et al., 2011b].

This section is based on work also presented in:

- H. Bosch, D. Thom, M. Wörner, S. Koch, E. Puttmann, D. Jackle, and T. Ertl. ScatterBlogs: Geo-spatial document analysis. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST '11)*, pages 309–310. IEEE Computer Society, 2011

3.2.1 The IEEE VAST Challenge 2011 and ScatterBlogs

The challenge scenario is situated in the fictional city *Vastopolis*, having about 2 million residents, where an epidemic outspread of respiratory and gastrointestinal diseases is reported among the population. As a starting point for the analysis, frequently observed symptoms such as fever, chills, sweating, headache, and diarrhea are given. The challenge consists of two tasks:

1. Origin [...]: Identify approximately where the outbreak started on the map (ground zero location). If possible, outline the affected area. Explain how you arrived at your conclusion.
2. Epidemic Spread: Present a hypothesis on how the infection is being transmitted. For example, is the method of transmission person-to-person, airborne, waterborne, or something else? Identify the trends that support your hypothesis. Is the outbreak contained? Is it necessary for emergency management

> personnel to deploy treatment resources outside the affected area? [Grinstein et al., 2011]

The provided data consist of three parts: a textual corpus, a geographical map, and weather conditions. Each part is provided with location and time information that allows to link the data sets. The corpus is composed of approximately one million microblog messages, i.e. short text messages with an upper length limit of 140 characters and an average length of roughly 60 characters or 13 words. Each message is accompanied by an identifier of its composer, a timestamp, as well as the GPS coordinate of the precise location from which it was sent. A map of Vastopolis is provided as an image, divided into districts such as *Downtown* and *Westside*, and features points of interests such as airports, hospitals, concert halls, etc. Population densities are available separately for each district. The GPS coordinates of the image corners are also provided to be able to relate the map and messages to each other. The weather data consists of daily measurements of the predominant sky condition (clear, cloudy, rainy), wind direction, and wind strength. With the georeferenced map, it's landmarks, the messages timestamps, and the additional metadata, one can derive further details, e.g. the normalized message distribution depending on the population density, if an message was written at a hospital, and the wind direction at the time of writing.

In order to solve the challenge tasks we developed *ScatterBlogs*, an MCV environment (see Figure 3.3) based mainly on spatiotemporal text analysis and selection management. With this environment, analysts can easily create and redefine subsets of the data for, firstly, a free exploration of the data set by inspecting these subsets of noteworthy similarity and, secondly, formulating and validating hypotheses based on the combination and comparison of message sets. Its main visualization is a map view, where a selected set of messages is scattered according to their latitude and longitude values, hence the name of the prototype *ScatterBlogs*. The temporal dimension of the data is available as the color of the scatter dots, as a third dimension of the scatter plot, or as a filter to slice the data set into subsets of similar time spans. The time filter is presented as a hierarchical time slider that filters time first by day, then by hour and finally by minute. Each layer of the time slider is equipped with a histogram of temporal distribution of the whole message set. Further, a table view allows to read selected messages individually and the selection management component is included as a separate view.

The amount of messages introduces scalability constraints and interacting with all of them at once poses an infeasible load on both analyst and algo-

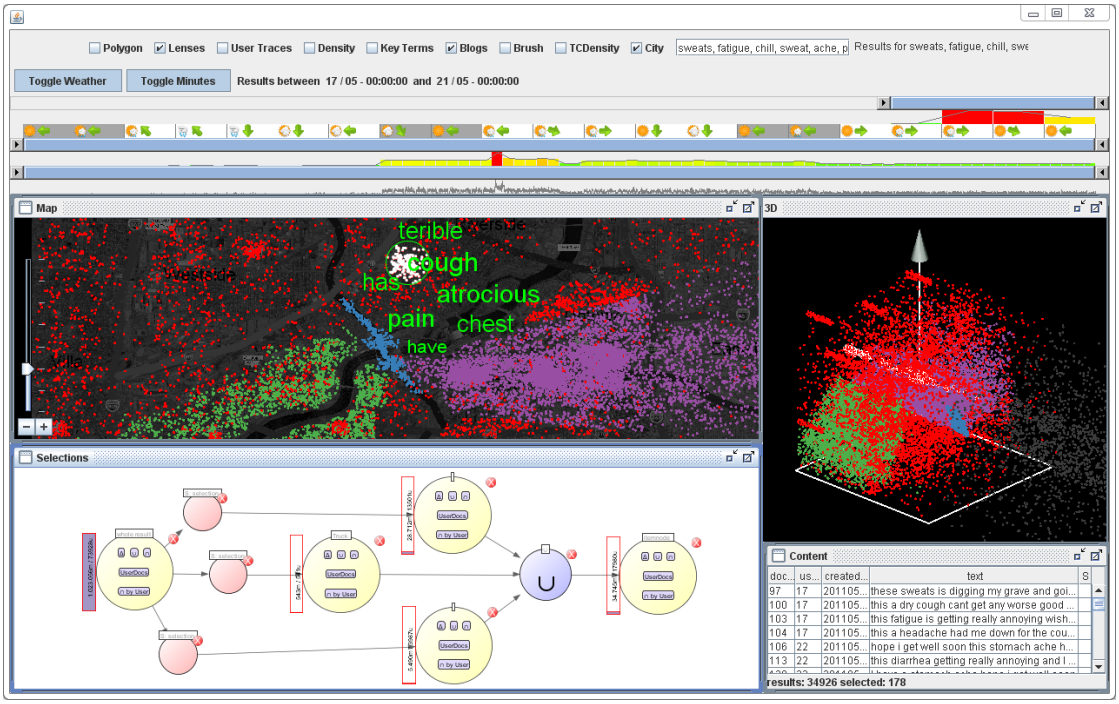


Figure 3.3 — The *ScatterBlogs* desktop from top to bottom: row with overlay selection boxes and textual query input; tree layers of the hierarchical time slider with histograms and weather icons; two and three dimensional scatter plot of the messages on the base map together with a content lens; the selection management component used to separate and color code message sets; content table with the raw data of the messages selected by the Content Lens.

rithms. Therefore, the data set and the behavior of subsets is structured in three conceptually and practically different layers: *base data*, *analysis context*, and *highlighted selection*. The *base data* is the total and immutable amount of the preprocessed, linked, and indexed challenge data set, which is made available for integration into the multiple views of the *ScatterBlogs* desktop. It has no direct visual representation and therefore takes the role of the *Data Tables* of the information visualization reference model (see Section 2.2.1).

Through a user-defined query, a subset of the base data is extracted and established as the current *analysis context*. It is visualized spatially on the map, temporally on the time slider histogram, and textually in the table view. This separation of the analysis context has several scalability

benefits. Primarily, the omission of the general noise reduces the clutter of the message visualization and allows for a better identification of patterns and outliers. For this purpose, the initial query does not have to perfect, because the base data is still available and the query can be modified iteratively in order to optimize the context. Still, a first idea for establishing an analysis context should be available (e.g. the challenge scenario's symptoms list), but it can also be extracted through geospatial text analysis as will be shown later. In addition, an already reduced and more targeted analysis context allows for an easier definition of further subsets and simplifies the subsequent interaction, e.g. allowing more meaningful spatial selections. Last but not least, the performance requirements are lower because fewer data have to be processed for each interaction during the analysis. As each element of the analysis context has a visual representation, it takes the role of *Visual Structures* of the information visualization reference model.

The third level are *highlighted selections*. Following the brushing and linking method, every set of messages from the current context that is selected, e.g., by a textual, temporal, or spatial filter or brushing operation, is highlighted on the map and in the textual view alike. This allows to see whether a set of messages—of a certain similarity defined by the selecting filter definition—also shares other features and therefore should be analyzed further. One can understand selections as a quickly defined *focus* contrasting the current context.[1] Causing only a slight modification of the sort order and change in the rendering mode, one can see highlighted selections as a part of the *View* of the information visualization reference model. For these subset layers, the generic selection management helps in re-purposing current and saved entity sets to the different layers.

The integration of *ScatterBlogs* and the filter/flow selection management component is realized mostly by filters that enumerate message IDs and by tagging as well as a by few specifically developed filters. The single root node contains the base data set. The component is part of the brushing and linking assembly and therefore is notified about every set selection within the base application. If a noteworthy set was selected, the analyst can instantiate it in the selection management component and it is added as an enumeration filter, i.e. a filter that matches only the enumerated entities of its associated set, to the root node. After working with the saved sets in the canvas, the analyst has two options to bring the result back into the other *ScatterBlogs* views. Firstly, any saved or composed set can be defined as analysis context or highlighted

[1] A cue-based technique according to Section 2.2.3

selection, thereby allowing the previously mentioned re-purposing of sets. Secondly, a node can be tagged. This adds a user defined text to the node to allow for capturing the semantic of the employed operations. Additionally, the entities are marked persistently with a user-chosen color in the base data set. This allows the tracking of messages throughout the whole further analysis process.

As has already been hinted, a set of messages can be selected by multiple ways and different message attributes. The hierarchical time slider allows to select messages from a user-defined time span in a per-minute resolution. The combination of histogram, highlighting, and the possibility to browser over the whole time span of a data set provides a fast overview and supports the identification of interesting points in time. For geospatial selections, the user can simply brush areas and draw polygons on the map. An advanced way of defining spatial selection is the *Content Lens*. Here, a circular lens can be moved freely over the map and the textual contents of the underlying messages are summarized by showing the most frequent words in a term cloud around the lens. This allows for a targeted selection due to the preview of the messages that will be selected once the lens is dropped.

The selection by textual content is done by standard keyword queries. This requires an initial idea of which words are used by the community to describe the events of interest. Because this knowledge is not always given, especially during an exploratory task, a technique to provide potential entry points has been developed. This technique finds and presents spatiotemporal clusters of word usage in three basic steps. First, each term of the message collection is inspected individually and the spatiotemporal distribution of the messages containing it is scanned for clusters. The largest cluster is then considered as the term's most significant position and its member count is normalized by the overall frequency of the term and weighted by its spatiotemporal variance. Once the positions of all terms are calculated, we try to find an optimal layout of labels that places each label near to the most significant location of its term. Terms with a low spatiotemporal variance and a large cluster size have a high weight and are thus emphasized by placing them first and in a larger font. The algorithm does not guarantee that every term receives a label because there are upper limits on how far a label is allowed to move away from its term location. The result is a 'map of terms' that indicates possible events based on the observation that many people used the same word in close vicinity and during a short time span. With this map of terms, an analyst can easily click through the labels, thereby setting the keyword query to this term and inspect the distribution and contents of the related messages. An enhanced

and streaming-enabled version of the technique was also developed [Thom et al., 2012b]. In the following, the benefits of integrating *ScatterBlogs* and the filter/flow approach are demonstrated with a use case from the challenge scenario.

3.2.2 Use Case and Results

The first challenge task is to find the origin of the epidemic and outline the affected regions. This can be achieved by using the filter, selection, and visualization mechanism of ScatterBlogs and does not require the creation of more complex sets using the selection management component. We start with establishing the current analysis context by querying for all messages containing any of the symptoms mentioned in the scenario description, i.e. `fever`, `chills`, `sweats`, `aches` and `pains`, `fatigue`, `coughing`, `breathing difficulty`, `nausea` and `vomiting`, and `diarrhea`. Scrolling over the densest regions of the time histogram by one hour intervals while observing the distribution of the respective messages on the map, quickly reveals a large concentration of symptom related messages in the central area (purple dots in Figure 3.3), followed by a second concentration in the south eastern area along the river on the following day (green dots), and finally the appearance of several message hotspots throughout the map (clusters of red dots). A closer inspection of the hotspots shows that they are centered at the locations of hospitals marked on the map.

The outline of the two densest areas form two wedges pointing to a central bridge, which possibly is the place of origin for this epidemic, but there are no messages within the current analysis context that could indicate the event causing the outbreak. As we do not know the correct terms to search for this unknown event, we consult the map of terms related to spatiotemporal density anomalies (see Figure 3.4). The terms `truck` and `trucks` are prominent at this location and a click on each term includes the related messages into the analysis context. Examining this region with the Content Lens reveals that there was a severe truck accident including fire and spilled cargo, which was released into the air and river. At that point one can reason that this could have been the cause for the respiratory and gastrointestinal problems, respectively. This hypothesis is further supported, because this event happened closely before the outbreak (see space-time cube in Figure 3.3), the wind direction on that day matches the affected region to the east, and the affected region along the river is located downstream. An assessment of the contents of the messages from affected regions using the

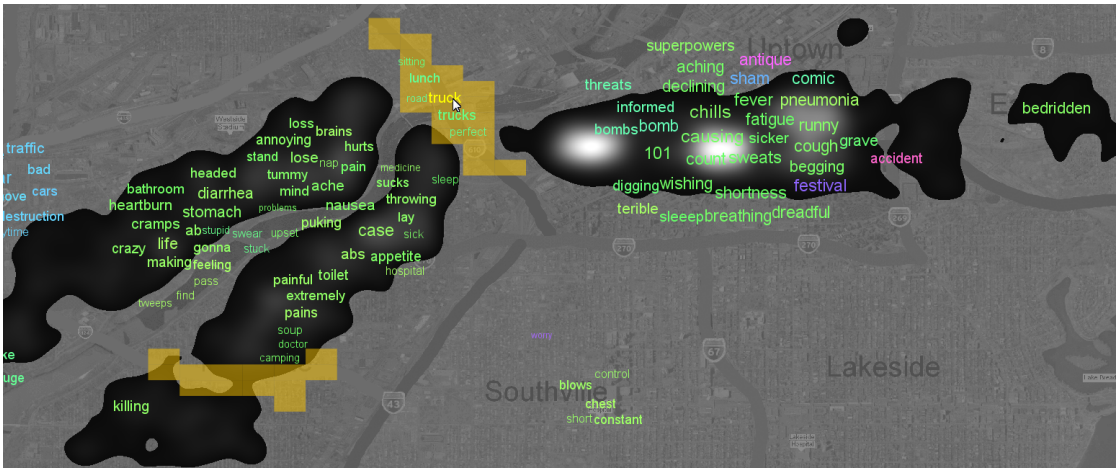
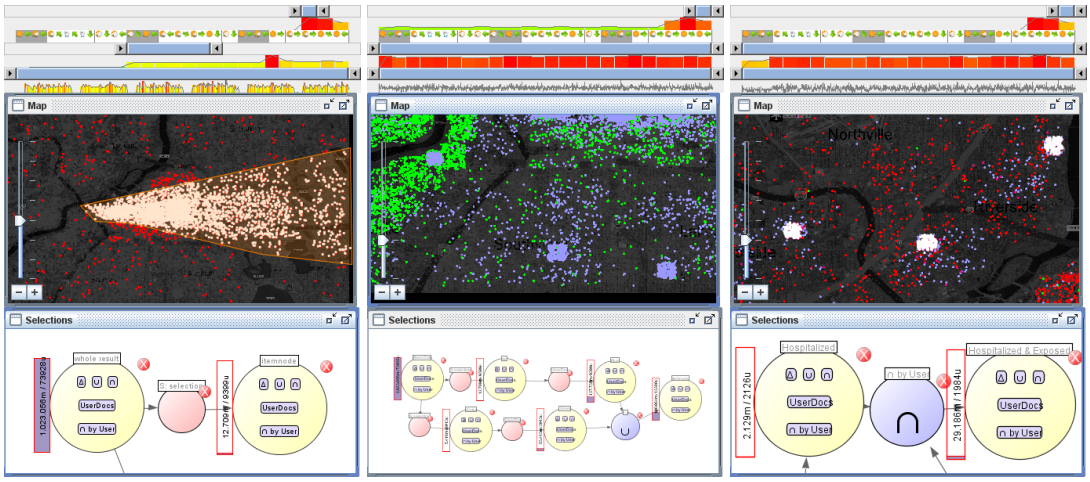


Figure 3.4 — Map of event-indicating terms over the city of *Vastopolis*. Terms of similar color relate to events that are close to each other in the temporal domain and thus potentially related. The dark areas show a high density of disease related messages and form two cones meeting at the location of a `truck` event. The yellow areas show the spatial clusters of messages containing the term `truck` which is currently under the mouse pointer.

Content Lens shows that the symptoms differ between the two areas. The downriver area population reports mainly gastrointestinal symptoms, such as stomach pain and diarrhea, while the people on the western side of the truck accident report more flu-related symptoms, such as fever, coughing, chills and breathing difficulties.

The second task comprised identifying the means of transmission which can be answered by relating groups of users and their messages in order to rule out or confirm a person to person transmission of the illness. This requires more complex filter definitions and can be achieved by using the selection management component. For this purpose, the persons that were exposed to the truck accident event and reported symptoms have to be isolated. As the airborne and waterborne infections result in different symptoms, the groups are examined individually. Then, the persons that report from hospital locations are selected and their relation to the other sets is examined to see if they were directly exposed to the truck event.

Accordingly, we first split the symptom list according to the type of infection and establish the analysis context with the flu-related messages. After we selected the time of the first burst of relevant messages, we apply a spatial filter by drawing a polygon around the affected region which is now clearly



(a) (b) (c)

Figure 3.5 — Screenshots from different phases of the use case description: (a) selecting all messages that indicate that the author was exposed to the air pollution and storing them in the selection management component; (b) contrasting users exposed to air pollution (blue) and water pollution (green) at the hospital locations; (c) comparing the `hospitalized` user group (from the highlighted selection of hospital locations) to the user group of exposed users.

visible on the map (see Figure 3.5a). The respective filters and result sets are then instantiated in the selection management graph as filter and set nodes and intersected by using a join node. The resulting set are the first symptom-mentioning messages from the persons that contracted the disease directly through the truck event. Because we are interested mainly in the message authors and their further course of action, we include all other messages from the same author by joining the set with the base data set in the root node, but this time with a special join node including all messages that were written from a person having authored at least one message in all incoming message sets. In order to identify these users more easily in the further analysis, we tag the content of the result node with a color and add a descriptive label. We now repeat the same procedure for the gastrointestinal symptoms and its affected region. One can now unify both tagged sets, send the data back to the map visualization, and observe at the hospital hotspots that the color of the flu-reporting users dominates this locations (see Figure 3.5b). In a similar fashion we can create a set of all users that have send a message from

a hospital and intersect it with each user group. Here, it can be seen that the intersection with the flu-reporting users does not reduce the number of users in the ‘hospitalized’ set significantly (2126 users compared to 1984 users, see Figure 3.5c). We can therefore rule out that other persons contracted the disease outside of the affected regions through person to person transmission.

These examples show how a generic, entity-centric set management component can be used to enhance existing atomic filter and selection mechanisms. With an expanded filter/flow metaphor it integrates easily with domain tools. In addition, it helps in expanding or reducing query or filter result sets to shape them according to the task at hand, reduce noise, and build the necessary data base for hypothesis testing. However, it can also be seen that adapting the approach to a domain, e.g. through additional filters for related entities, leads to software that can solve domain-related tasks more efficiently. In this example, the central entity of the analysis can shift between a microblog message and an author of such messages, even within a single task. With very limited additions to the generic selection management component, such as including author counts and author join nodes, the applicability to the task can be largely increased.

3.3 Explicit Filter Constraints for Search Query Feedback Loops

The intellectual property domain has specific requirements for searching and analyzing documents due to their legal nature and the high financial risks. Search queries are therefore honed to perfectly match the information need by iteratively integrating new aspects. A feedback loop from result set exploration to query refinement can support this task efficiently. In this section the introduced selection management component's ability to facilitate search query feedback loops within the *PatViz* patent search and analysis system is presented. Therefore, it briefly summarizes the patent domain, its commonly observed tasks, and the *PatViz* system before detailing the role of the selection management component.

This section is based on work also presented in:

- M. Giereth, S. Koch, H. Bosch, and T. Ertl. Visual patent retrieval. In *Internationales Rechtsinformatik Symposion (IRIS'08)*, pages 569–574, 2008
- S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative integration of visual insights during patent search and analysis. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST '09)*, pages 203–210, Oct. 2009
- S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative integration of visual insights during scalable patent search and analysis. *IEEE Trans. Vis. Comput. Graphics*, 17(5):557–569, 2011
- S. Koch and H. Bosch. From static textual display of patents to graphical interactions. In M. Lupu, K. Mayer, J. Tait, A. J. Trippe, and W. B. Croft, editors, *Current Challenges in Patent Information Retrieval*, volume 29 of *The Kluwer Int'l Series on Information Retrieval*, pages 217–235. Springer Berlin / Heidelberg, 2011

3.3.1 Intellectual Property Domain

A patent is an intellectual property right that grants the owner a temporary, exclusive right to exploit and market his/her invention in exchange for making it public.[2] In order for an invention to be patentable, it has to contain a

[2] The term *patent* originates from Latin *patens* meaning 'laying open' [DUDEN, 2001].

non-trivial ‘inventive step’ and has to be novel in the meaning of not being published before the patent application. This right encourages the publication of inventions to allow their free use after the exclusivity protection term, usually 20 years, has expired. Patents are of high relevance in today’s globalized markets due to the increasing complexity of products involving many inventions from various technological fields. Because the patent owner can prevent others from marketing products which include their inventions, the commercialization of these products is threatened in the countries in which the patent is in force. These characteristics make the search and analysis of patent documents, and similar intellectual property rights or technical publications, an often recurring task of many stakeholders in technical domains, e.g. searching for technical solutions, assessing the *freedom to operate* for developing and commercializing a certain product, or finding prior art for a new patent application.

Finding the right patent documents is a difficult task, primarily because of the large number of documents and their language. Patents are both legal documents as well as technical descriptions, and thus the language of the documents, i.e. their structure and terminology, is often complex, abstract, and paraphrasing. In 2012 an estimated total of 2,35 million patent applications have been filled worldwide, with accelerating growth rates over the previous three years [WIPO, 2013]. In the same period an estimated 8,66 million patents were in force, and the *Espacenet* service of the European Patent Office[3] offers access to more than 80 million patent documents. Additionally, these patent-related figures only cover one of the three major intellectual property types next to trademarks and industrial designs. Due to the financial consequences of failing to find a relevant patent (forfeit development investments, law suits, etc.), patent search specialists tend to rely on predictable search mechanism, such as a Boolean search, to be able to shape their search queries iteratively. Here, during each iteration the query is optimized by either widening the search to include new aspects or variations of the search terms that surfaced during the investigation of the results, or by narrowing the search to exclude noise from, e.g., unrelated technological areas or homonyms of the search terms.

[3] `http://ep.espacenet.com`

3.3.2 The PatViz System

The PatExpert project [Wanner et al., 2007] recognized the impact of an efficient patent retrieval process and developed new ways to search for and analyze patent documents. Its main focus was to move from 'textual' to 'semantic' patent processing by introducing content-oriented search engines and improve document summarization and representation. The project consortium developed several search engines targeting different content types of the documents individually and thus allowing to search for similar images, abstract semantic concepts, and semantic relations between concepts in addition to the traditional full text and metadata searching. *PatViz* [Koch et al., 2011] was developed to integrate the visualization of patent document collections, e.g. search results, and the creation of queries for the separate content specific search engines. This integration supports the previously mentioned incremental nature of query refinement by allowing a fast assessment of intermediate result sets and a subsequent update of the query efficiently. This is realized by conveying the information about a patent set with a collection of attribute-specific visualizations and by a visual query editor which are both linked by a MCV environment, the *PatViz* 'desktop' (see Figure 3.6).

Patent documents feature many metadata attributes such as various dates (application, publication, filing, and priority dates), actors (applicants and inventors), relations to other documents (priorities, patent families and citations), locations (designated states, country of origin), and classifications. This renders MCVs especially suited to provide perspectives on all these patent related aspects at the same time using multiple visualizations such as histograms, node-links networks, maps views, etc. The mentioned content oriented search engines add further data about the patents during their indexing of the document collection. Among these are term frequencies, contained semantic concepts and their relationships, and the types of embedded images. *PatViz* employs most of these data types to draw visualizations that provide an overview over a patent collection as well as detail views to examine individual documents.

The *Query View* is a structured editor for Boolean queries. Due to the predictability and controllability of Boolean search queries, they are predominant in the patent domain and thus familiar to the targeted user group. They also facilitate the integration of the various search engines by performing simple set operation on their individual result sets [Codina et al., 2008]. The query editor provides two views on the same query model. The textual view is most similar to the query fields of traditional patent search

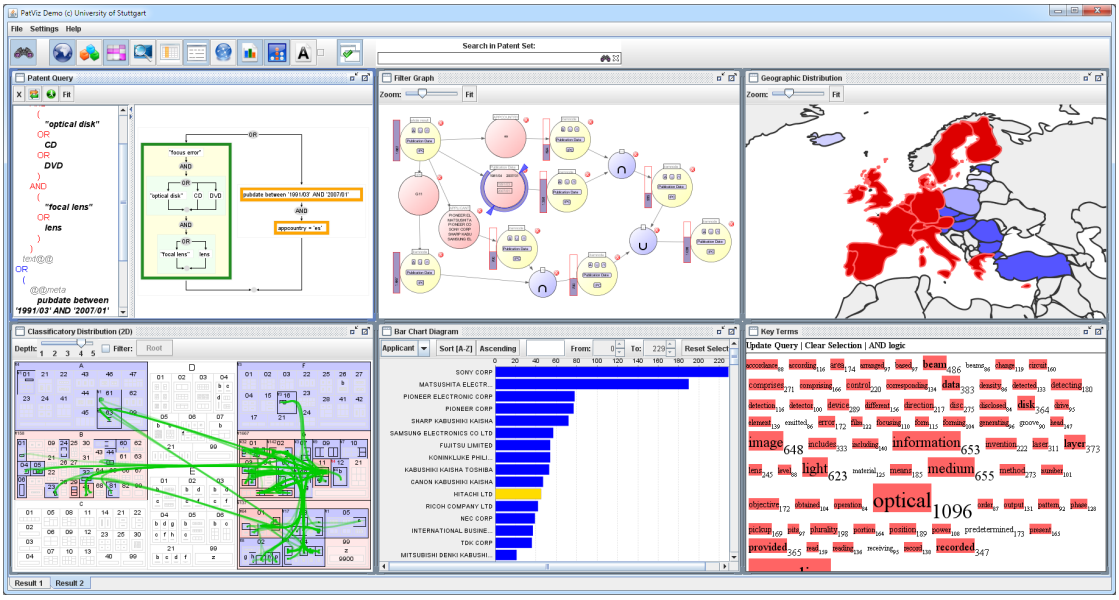


Figure 3.6 — The *PatViz* desktop showing a subset of its available views, from top to bottom and left to right: *Query View*, a combined textual and visual query editor for Boolean queries; *IPC Treemap*, a distribution of the document set over the IPC classification schema shown as a treemap; *Selection Management*, the graph of the presented selection management approach; *Bar Charts*, a bar chart component showing the distribution of the set over a choosable attribute; *World Map*, a choropleth depiction of the set; and *Term Cloud*, an alphabetically ordered sequential tag cloud of the patent set's key terms.

engines but is formatted with indentations to make the Boolean structure of the query more salient. The second view visualizes the structure of the query using containment and branching in a node-link diagram with orthogonalized edges similar to the format of Syntax Diagrams [Wirth, 1973]. The operands of a Boolean combination are nested into a shared bounding box either sequentially in the case of an `AND` or parallel in the case of an `OR` operator. This bounding box is again used as a building block in larger Boolean combination. Both views are coordinated by brushing and linking so that hovering over a part of the query in one view highlights its representation in the other view. Due to a shared model, modifications in either view are reflected in the other view. The clear depiction of the query structure allows for an easy modification of any part to integrate new findings from the result set exploration.

The International Patent Classification (IPC)[4] is a hierarchical classification schema to structure technological areas and is thus an appropriate way to narrow search queries. Each patent document is classified in one or more elements of the IPC. The first level of the schema are the eight 'Sections' from (A) `Human Necessities` to (H) `Electricity`. The *IPC Treemap* depicts up to five levels of this hierarchy in an ordered and squarified treemap layout [Shneiderman, 1992; Shneiderman and Wattenberg, 2001]. Additionally, the elements of the lowest level receive a fixed, uniform area and define the size of its parents. This leads to a stable mental map[5] of the technological areas independently from the currently shown document set. The distribution of a patent document set over the IPC schema is shown by color instead of size, as it is usually the case in treemap displays. For this purpose, the patent set of each IPC element is inherited by its ancestors in order to show also patents that are classified deeper in the hierarchy than the five levels that are depicted. The elements of the treemap have co-classification relations when at least one patent was classified in both IPC categories. This information is shown as additional links over the treemap layout. Because straight lines crossing the display would lead to visual clutter, the edges are routed over their ancestor nodes using the Hierarchical Edge Bundling approach by Holten [2006]. The IPC Treemap is also available as a 3D version, in which the patent set's distribution over the IPC schema is additionally encoded as the height of the tree elements [Giereth et al., 2008a].

The *Bar Charts* can be used to aggregate the patent set by their related legal entities that are involved in the patent applications, i.e. organizations and inventors. It can be sorted alphabetically or according to the aggregated patent count per entity. Selecting elements can be done by clicking on them individually or by drawing a rectangular selection box. Because organizations can use slightly different names in the different countries or translation errors lead to different spellings, a selection using regular expressions is also possible.

The *World Map* is a pan- and zoomable choropleth map of the world. The color saturation for a country depicts the amount of patents that include the country as a designated state. When the actual number is of interest to the analyst, moving the mouse over a country reveals the count as a tool tip.

The *Term Cloud* includes the most frequent terms of the current patent document set. Similar to Tag Clouds, used to show the importance of user

[4] `http://www.wipo.int/classifications/ipc/en/`

[5] In the sense of Misue et al. [1995] retaining orthogonal ordering, proximity relations, and topology.

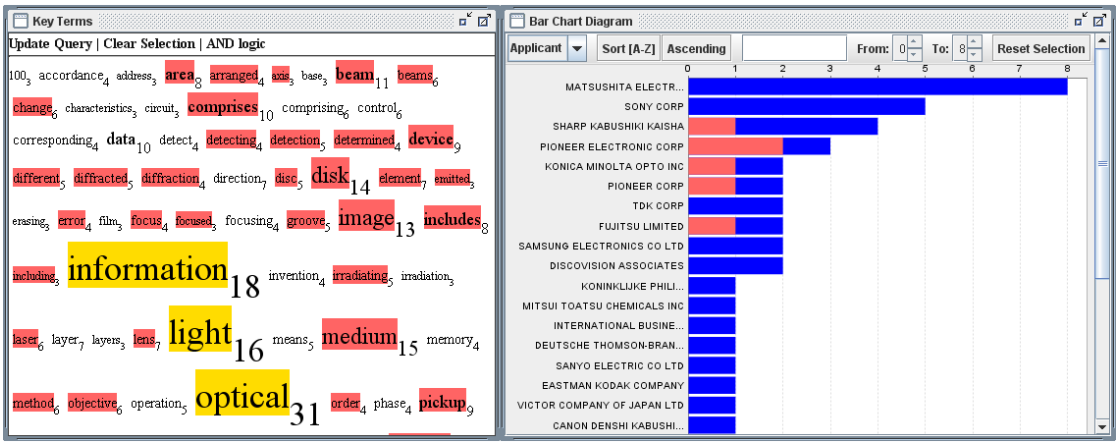


Figure 3.7 — Two linked views of the *PatViz* system. Yellow terms were brushed by the user thereby selecting all patent documents containing all three terms. Red highlights indicate the related applicants and other key terms contained in the related patent texts.

assigned ‘tags’, the more frequent terms are shown with an increased font size to draw the attention of the viewer. In combination with the alphabetic ordering of the terms, this setup is suitable for the tasks of finding the most frequent terms as well as to find an a priory defined term in the list (see Lohmann et al. [2009] for a task-centric comparison of tag cloud layouts). The document frequency is used for measuring the importance and is shown as a subscript to each term. This denotes the number of documents that will be selected when clicking on the term. If the patent set has a very narrow focus due to its search query, a small number of terms will be contained in almost any of the documents. In such cases selecting multiple terms can help in shaping the selection and finding a subset of documents that contain all of them. The combination logic of a multiple terms selection can be switched from `AND` to `OR` to support other scenarios.

More details about individual views of the *PatViz* system as well as visualizing patent information in general can be found in the theses of Mark Giereth [2012] and Steffen Koch [2012].

All the views are coordinated by brushing and linking. Selecting an element in one view highlights all related patents in every view of the system. While there are many different entity types[6] displayed in the views (such

[6] In this section, the term *entity type* is used almost synonymous to *attribute*. The main difference is the notion that an *entity* can also exist without being an *attribute* of something.

as countries, people and companies) they are always related to one central entity type: the patent document. This is the common ground for the linking of views. Each brushing action may at first be related to, e.g., an applicant name within the bar chart view or a set of terms in the term cloud, but is internally translated into the set of patents that where applied for by this applicant. This set of patents is subsequently highlighted in all views of the system. Because a patent document can be linked to multiple entities of the same type, e.g. it can have multiple applicants, it frequently happens that the highlighted parts comprise more entities than the scope of the original selection, even in the view in which the selection took place. This means that selecting one applicant in the bar chart view can lead to the highlighting of additional, unselected applicants. While this may at first surprise the user it leads to a more consistent behavior of the system, as the highlight retains its semantic throughout all the view, including the view of the selection. In order to support users in distinguishing between highlight and selection, we use two different colors for highlighting the selected documents (yellow) and the selected entity (red, see Figure 3.7). Unselected entities are shown throughout the system in various shades of blue depending on the number of documents that relate to the displayed entity.

3.3.3 Integrating Intermediate Insights into the Search Process

The distinction between brushed entities and subsequently selected patent documents leads to two important aspects for the integration of intermediate insights into the analysis process: First, the presence of a central entity type facilitates the integration of the filter/flow based selection management; Second, the selection semantic (‘patent documents related to the selected entity’) is preserved for updating the current query. These two aspects and their interplay are detailed in the following.

Managing Selections

The role of manual selection management in this scenario can be described best by looking at analysis problems that relate to different numbers of attributes. Taking into account only the views of the *PatViz* desktop, we can observe each attribute independently because most views visualize the distribution of the document collection over one entity type, e.g. the choropleth world map distributes the set among the geographic locations, the bar chart shows the patent count per applicant. Without any further interaction we

can therefore only draw insights about how each entity relates to the other entities of the same type and to the total document collection.

The brushing and linking functionality within the *PatViz* desktop allows for evaluating a single filter criterion on the current document collection in a fast and straight-forward way. Here, the user-selected entity of the brushing interaction acts as the filter criterion while the highlighting shows, both, the result of the filter application as well as its relation to the original document collection. For instance, while the prominence of a certain word within the current document collection can be directly seen in the term cloud view, we cannot see whether the term usage varies among the different applicants. Selecting the term leads to a highlight of all documents containing the term and one can then observe which fraction of each applicant's patents contains this term. With this feature it is therefore possible to draw insights about how one entity relates to other entity types in a 'one-to-many' fashion. There are, of course, other solutions to compare these different attributes of the core entities, e.g. user configurable visualization to contrast two attributes directly. However, the *PatViz* solution allows the analysts first to set up their analysis workbench by organizing the most relevant views, and then quickly select different entities to explore the data set freely.

Finally, if the documents that the analyst wants to highlight shall be derived from more complex filter definitions, consisting of criteria spanning different attributes, the employed brushing and linking strategy fails, because each new brushing operation replaces the previous one. Here, we need an increased expressiveness that utilizes the brushing operations as building blocks for more complex extraction strategies. As has already been mentioned, each selection defined by a brushing operation is internally translated into a set of patent documents and it is therefore straightforward to integrate the entity-centric selection management component for this task. Each selection can be stored in the filter/flow graph structure as a filter on the root node, which represents the whole result set. Afterwards, the analyst can combine these filters within the graph view, as has been described in Section 3.1, and highlight the resulting patent documents sets throughout the *PatViz* desktop like any other brushing operation. This interaction is shown in Figure 3.8, where the analyst first selects the three terms `information`, `medium`, and `optical` from the key term cloud and stores the selection in a node. Another node is then created by selecting all patents of the result set that apply to the country of Spain. The selections are intersected and the result is reflected back into the other views. Among them is the table view, which can be used to scroll through selected patents and examine the individual documents.

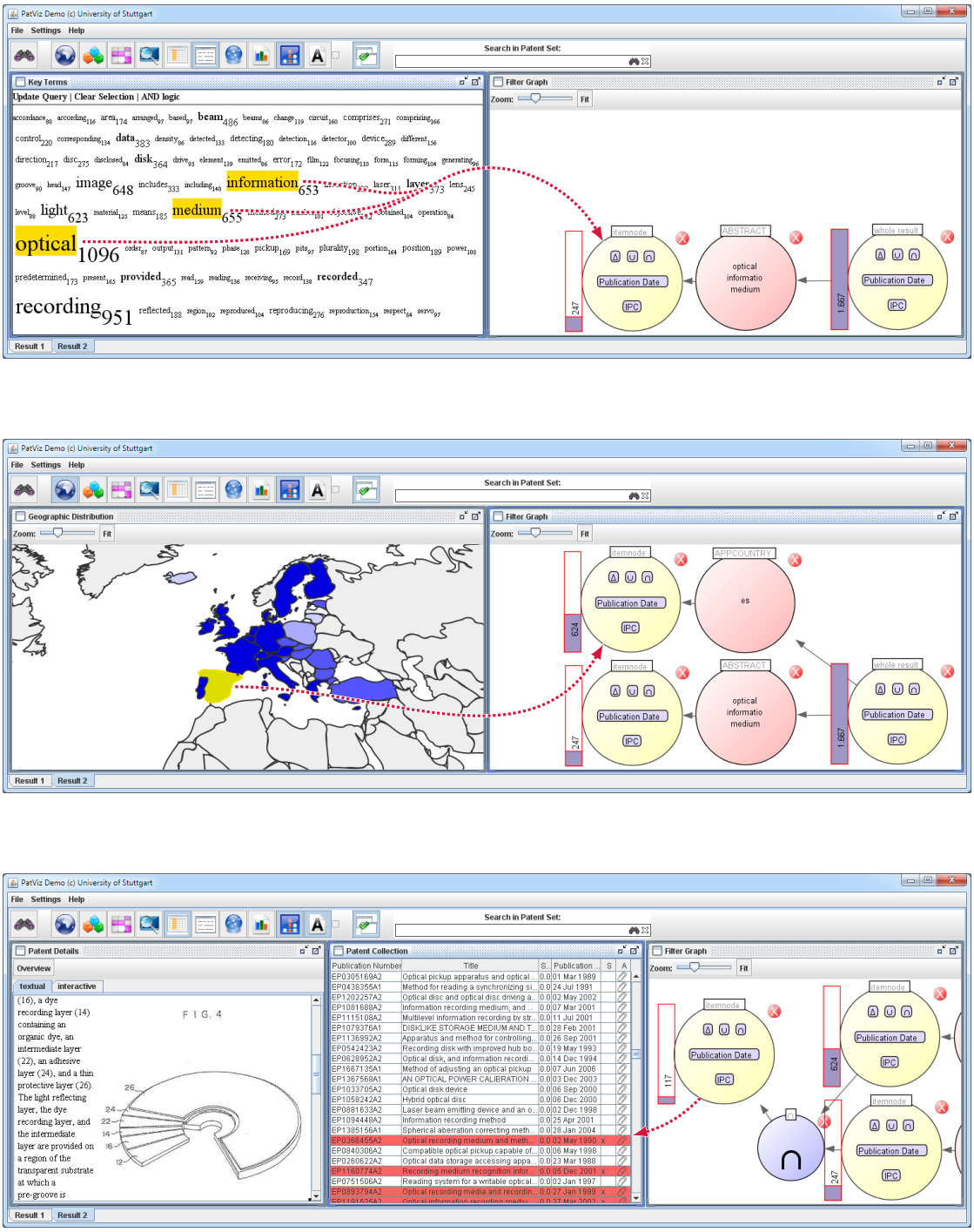


Figure 3.8 — Combining a keyword and geographical selection by an intersection join node and highlighting the result in the table view. Relations between nodes and selections/highlights are shown as red arrows.

Similar to the *ScatterBlogs* prototype, the graph, which can be built in a user-steered process, is composed of three different types of nodes: *set nodes*, *filter nodes*, and *join nodes*. Set nodes have a vertical bar attached to them symbolizing the size of the set they represent in relation to the whole patent collection. Additionally, the bar is labeled with the exact size of the set. Set nodes can be connected to filter nodes, which constrain one of the set's attributes, in order to restrict a node's set of documents. The result of the restriction is another set node with the reduced document set. The third type of nodes constitutes set operations to join the content of multiple nodes and provide the result of the operation as a further set node.

The join node was introduced to express Boolean combinations of filtering constraints either by the respective join node types, i.e. union and intersection, additionally to the graph structure, i.e. sequences and branches. In contrast to the filter/flow metaphor [Young and Shneiderman, 1993] these explicit node operators facilitate the combination of arbitrary sets without the need to generate multiple instances of a particular filter just to apply it in different combinations. This further supports the explorative nature of the analysis because it allows to use arbitrary set nodes, regardless of their origin, without having to integrate them in a special place of the tree structure just to define the combination logic. These inclusions can then easily be reverted or even be ignored if the result was not helpful, because the original set nodes still are available to drive the analysis in other, more promising directions.

Selection Semantic

The brushing and linking of *PatViz* facilitates the analysis of the current result set of a patent search. Through multiple specialized views and by employing brushing operations either as 'single-use' filters or as building blocks for the selection management graph, one can quickly examine the characteristics of the result set and gain insights into the related entities. While these insights might be the desired outcome of the analysis, they are limited to the search result set which is currently shown in the views of the *PatViz* desktop. In order to exploit these insights during the overall information retrieval process, one has to integrate them into the search query. This way, one can retrieve more documents that feature the now known interesting characteristics or avoid certain combinations of attributes that have been identified as misleading, e.g. due to homonyms in different domains.

For the tight integration of result visualizations and search queries, the

current selection within the result set documents can be used to update its associated query. Here, the difference between the brushing interaction and the highlighting is important. While the interplay between the views of the desktop is solely based on patent document sets, the filter criterion needed to update the query is defined by the brushed domain entities such as applicants, countries, etc. Otherwise, the insights would be reduced to a set of patent documents instead of the intention of the selection and would be of no use in finding similar documents. For this purpose, the selection intention which defines the 'similarity' is determined by the view in which the selection took place. For instance, the selection semantic of the map view is to find documents that are valid in the user-selected country.

The combination of enhancing the brushing operations with selection semantics controlled by each individual view and the explicit combination of selections within the controlled environment of the selection management graph is beneficial for both users and developers of analysis applications such as *PatViz*. For the latter group, this approach allows a very modular and adaptive architecture. During brushing operations, the underlying view has to provide a set of patent documents to all other views for identifying the highlight targets and the related portions of their visualization. Views capable of determining the selection semantic of the brushing operation solely attach a new textual property to this event, which contains the atomic filter definition that has to be included into the search query in order to retrieve 'similar' documents. If the view cannot provide this, one can fill-in a trivial representation by simply enumerating the document IDs. However, this fallback does not provide a useful similarity measure and is only included to provide a fail-safe design. When a current selection is placed in the selection management component, the accompanying selection semantics is stored with the filter node. Integrating a node of the graph into the search query is done by traversing each path from the current node to the root node, collecting the filter statements of the nodes, and combine them with the respective Boolean operations. However, the system cannot guess the nature of the insight and, accordingly, the intention of including a filter into the query. The default behavior of the system is to widen the search by joining the old query with the new statements by a Boolean OR. To narrow their search, users would have to use an AND conjunction and may have to negate the newly included statement if they want to exclude similar documents. This can be done with the interactive query visualization that observes the query structure and allows to alter nested parts of the query, such as the new statements which are grouped by their conjunction with the old query, by a context menu.

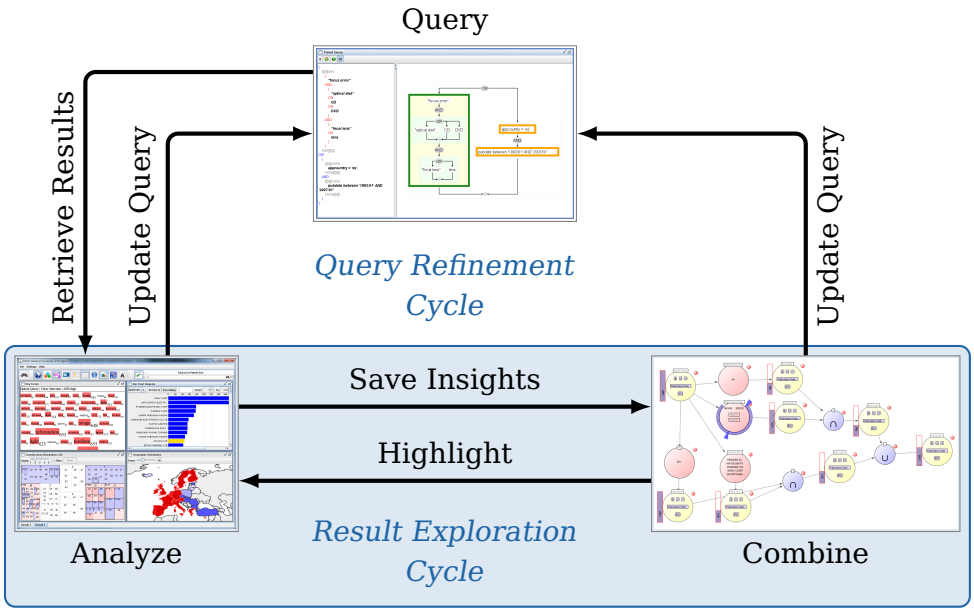


Figure 3.9 — The query refinement cycle and result exploration cycle. The results of an initial query are retrieved and analyzed/explored using the multiple coordinated views of *PatViz*. Noteworthy subsets of the data, potentially relating to an insight about the data set, can be stored and combined in the selection management component. The resulting models can be reflected in the views of *PatViz* for building trust in the filter definitions, and subsequently sent to the query editor for updating the query for another iteration.

In summary, the insight integration approach is modular and adaptive with respect to: (*a*) the responsibility of capturing the selection semantics is distributed among the views, (*b*) the combination is centralized and takes place on a Boolean meta level that handles every selection as an atomic building block, and (*c*) there is a baseline fallback for non-compliant components. The approach is beneficial for users because they can first test various filter criteria and combinations on the locally loaded result in order to build trust in their insights and filter definitions (the baseline of the triangular model presented in Figure 3.9) before they update their query. They can do so without causing further transaction costs[7] or losing their current analysis progress, which is persisted in the graph as well as in their current search query. The additional effects of explicitly persisting preliminary states of the analysis is further detailed in Section 3.4.

[7] Some data providers in this domain charge their clients per executed search query.

3.4 Collaboration and Reporting

The previous two sections discussed interactive data analysis systems and their interplay with one analyst. The systems are MCV environments that present various interactive views on a collection of homogeneous entities and their attributes. Their brushing and linking functionality, used to explore the data set, is enriched with an explicit selection management component that allows to save and recall selections from brushing actions. While supporting analysts to structure the analysis and mark simple findings, these saved selections can also function as building blocks to derive new compound selections in order to follow complex hypotheses about the data. Because they are created from meaningful user operations, selections are equivalent to filter definitions and as such can be used to assess a similarity to the selected entities. This allows to find more or exclude related entities by integrating the filters into search queries.

However, data analysis scenarios usually are not such an isolated system with the software and its single user being the only two participants. Real-world problems are often complex and thus the related analysis tasks are performed by teams of peers that collaborate and report their findings to decision-making supervisors. This section provides more details on the artifacts of such work environments and how the selection management approach supports it by providing a reuse and reporting facility.

This section is based on work also presented in:

- H. Bosch, D. Thom, M. Wörner, S. Koch, E. Puttmann, D. Jackle, and T. Ertl. ScatterBlogs: Geo-spatial document analysis. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST '11)*, pages 309–310. IEEE Computer Society, 2011

3.4.1 Division of Labor, Analysis Products, and Provenance

If an amount of work cannot be handled by a single worker in the desired amount of time, it has to be divided. The division of labor can take different forms depending on the nature of the work task. If the work can be split into mutually independent packages, the division of labor and the integration of results is trivial. Unfortunately, that is rarely the case for analysis tasks where insights from one work package may influence the relevance of an information in another work package. Therefore, the division of labor introduces a

communication overhead of varying degree depending on the type of division:

- *Splitting Work Load* – Splitting large data sets into parts that are not independent (e.g. a large text corpus in which each part may contain hints to a solution) between peers requires a frequent alignment of intermediate results and insights. It therefore requires the externalization and internalization of knowledge, i.e. *knowledge management* (see Figure 3.10). This can be supported by lowering the effort to annotate findings and data sets and semi-automatically generate reports that can be shared between the peers.
- *Embedding into Hierarchy* – If analyzing a data set or situation is a service provided by specialists for decision-making superiors or clients, the final report is the central product. Compared to the internal reports between peers, it has to be self-explanatory because the client has no access to the analysis system to explore the data in order to verify the results. Here, the notion of *provenance* is of great importance to the analysis product. To support high-impact decisions, the data and analysis quality has to be known and fit to the severity of the decision. Again, the analysis system can help by reducing the effort to include the needed information in the report.
- *Task Specialization* – Some subtasks during the analysis might require additional training or knowledge, e.g. in patent search when parts of a task refer to different domains with different terminology. If these subtasks recur often, specialists that help regular users in fulfilling these subtasks are needed, similar to the classical definition of 'division of labor' in the sense of the forming of specialized crafts. In these cases, it is more efficient to include (parametrizable) workflows of these specialists instead of gaining their knowledge when the respective skills are needed. To a certain degree, the selection management approach can also support this requirement by persisting a graph structure for later reuse.

The central element of the three examples above is the 'analysis product'. In each case, tacit knowledge or a transient state of the software has to be made explicit and externalized through exports, documentation, and annotation. This matches the capabilities of the selection management approach presented until now, besides externalization of the analysis product. This product can take the form of document collections (e.g. the most relevant patent documents), provenance enriched reports (e.g. affected regions of an

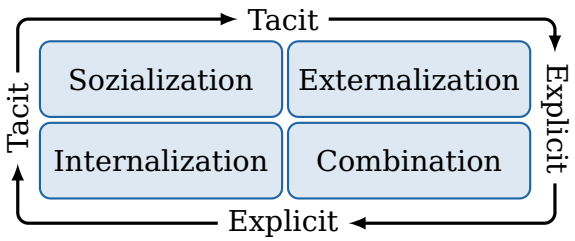


Figure 3.10 — The SECI model adapted from Nonaka and Takeuchi [1995]. Knowledge is either tacit (the knowledge of individuals) or explicit (written). Knowledge transfer can occur in the four ways depicted in the model.

epidemic outbreak with the relevant statistics and related documents), or whole subtask-workflows (e.g. a filter configuration for removing homonyms related to foreign domains). The analysis product, in the notion used in this thesis, does not have to be atomic or final. A report can be derived by exploiting foreign workflows and data sets and can itself be used as an input in other analysis tasks. Because most results will finally be used to support a decision,[8] this interdependency has to be documented as provenance information. The term 'provenance'[9] originally describes the chronological history of ownership that belongs to a piece of art and is important to assess its authenticity, artist, period of creation, and therefore its value. Similarly, the term provenance in computer science describes metadata on what procedures and filter where applied to what original data set, persistently documented from the data source to the final dissemination. This allows to assess the certainty and significance of the derived results by examining, e.g., the sample size, the reliability of the data source, and the loss of information during data conversions.

Of course, the effort that was put into documenting the provenance of an insight-generating analysis has to match the severity of a decision. Casual day-to-day decisions such as planning leisure time and small purchase decisions have low requirements on data quality because the cost of wrong decisions due to false analysis results is negligible. During the (fictional) example of the VC'11 (see Section 3.2.1) the analysis task was supporting the decision if and where medical emergency personnel should be deployed. Here, the

[8] Even the common exception to this statement that analysis results in the academic research are sometimes self sufficient and not aimed to support a concrete decision, the published insights will, of course, influence the decisions of others.

[9] From French *provenir de qc*, 'to result from smt.'

costs of unreliable data or low analysis quality would be much higher both economically and regarding potential loss of life. It would therefore require documented provenance.

3.4.2 Semi-Automatic Reporting with Provenance

As a team effort of computational power and human guidance, only half of the processing can be monitored automatically in visual analytics systems. Barring future extensions such as eye tracking and cognitive models, the sense-making, which is done by the analysts, can only be assessed coarsely by monitoring their input. Therefore, provenance-enabled reporting has to rely on human-provided annotation and is semi-automatic. Because the selection management approach already structures and augments the otherwise limited brushing and linking capabilities of multiple-coordinated-view environments, the annotation overhead is rather small if one can exploit this explicit documentation of interesting, intermediate results.

Under the assumption that the central insight, which shall be documented in the report, can be linked to a view of the analysis system including a highlighted portion of the data set, this insights can be persisted in a node of the selection management graph structure. If the insight relates to a difference between multiple sets of entities, the analyst would create multiple sets and compare them by creating appropriated join nodes. Following this assumption, the starting point for creating a report is a node within the graph structure. On the user's request, the system creates a basic report for this selected node. Because every node is an intentionally persisted state of the system, it is very likely that they represent a noteworthy intermediate result. Especially each node along the paths from the root node, containing the whole data set, to the selected 'report node' is relevant and therefore part of the report's provenance. The base report is a sequential and visual description of each node on these paths. The descriptions include a screenshot of the system's state or a purpose-built visualization along with figures about the node's set of entities such as its cardinality and type as well as additional user annotations, e.g. labels and tags. The report thus documents the views and entity sets that the analyst has seen and worked with while approaching the reported conclusion. A base report is generated as a static HTML[10] document. While this allows for a very simple dissemination of the reports with the ubiquity of web browser, it also allows the analyst to edit the report

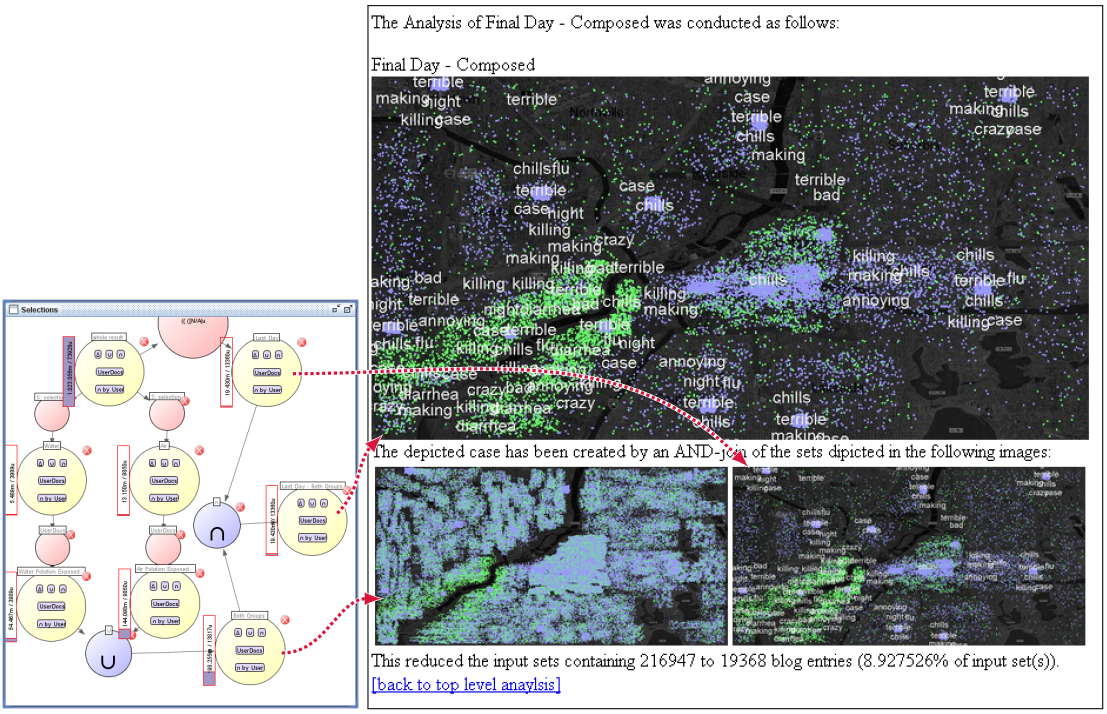
[10] http://www.w3.org/TR/html/

with almost any modern word processor. This is necessary because the semi-automatic report generation can only incorporate the intermediate results that lead to the final outcome, but not the human interpretation of the visualized data sets, which have to be added as further annotation and comments into the base report.

Report Generation in ScatterBlogs

Taking the analysis described in the VC'11 Use Case (see Section 3.2.2) as an example, *ScatterBlogs* created a report of the final node containing a comparison between the two symptom groups. Figure 3.11a shows the selection management graph that was used for answering the task's questions and Figure 3.11b shows the first description element of this report presenting the final node. In this case, the visualization of a node's content is a slight modification of the central map view of the *ScatterBlogs* desktop. The visualization contains further elements that shall substitute the other, unavailable views, e.g. the additional tag clouds around major message clusters are added because the full texts of individual messages are not contained in the report. Next to the image, text elements hold the user-assigned labels and descriptions, the cardinality of present entity types, as well as the join or filter operation that was employed in the creation of the node. As each node, except the root node is either created by joining or filtering other sets, these base sets are represented by the thumbnails at the end of the description element.

The thumbnails also link to similar description elements of the respective set nodes, which allows the reader to navigate the otherwise sequential document in a 'drill-down' fashion, exploring the provenance of the report. Even without access to the analysis system, the report consumer can move from the current result node to the most interesting parent node, containing additional detail about their composition. Here, a trade-of between completeness and complexity has to be made. The generated report can contain additional details that might be relevant in some cases but should be limited to the amount of data that can be presented efficiently by using only an interlinked document without further interactive filter and analysis components. For the prototype, it was decided to include only the nodes along the paths from the root to the final node and no raw data, e.g. the messages in the final document set. Irrelevant nodes can easily be ignored and individual messages can be included manually if necessary. Color was used to distinguish the different sets related to the two symptom groups. It can already be seen that



(a) (b)

Figure 3.11 — Analysis provenance in the form of selection management graph and automatically generated report. The red arrows indicate which node resulted in which thumbnail. Similar content elements exist for every node of the graph in the subsequent pages of the report.

their distribution differs, but the finding that the difference is greatest at the locations of hospitals has to be added manually by the report-generating analyst.

3.4.3 Workflow Reuse

The second externalization of an analysis process is persisting whole subtask-workflows for reintegration into other tasks. In contrast to semi-automatic reporting, the state of the selection management graph and their filter semantics has to be persisted instead of the system's views at the end of the analysis. Therefore, it is related to the aforementioned integration of intermediate insights into the patent search process (see Section 3.3.3). Again, this means that it is important to capture the selection semantics for creating

meaningful filters that are also applicable in similar scenarios but on other data sets, e.g. the same analysis at a later point in time with updated data. Simply storing the brushed documents IDs is not sufficient. Taking *PatViz*, the patent search system, as an example, an analyst has used the multiple coordinated views to clean a result set of irrelevant patents of other technical domains besides the one he or she is interested in. This was accomplished by selecting combinations of terms and IPC classifications and inspecting the highlighted patents for relevance. All irrelevant combinations were then unified by an OR-join node, and integrated into the search query as a negated exclusion fragment. The artifact which is most valuable for the analyst's peers is not the cleaned result list, but the term/classification combinations to clean related search tasks within the same technical domain. In this example, one could either reuse the resulting search fragment integrated into the query or the graph that was used to create this fragment. The latter option has the benefit that one can first explore the performance of the subgraph in its new context by highlighting the matching documents, and if necessary making adaptions to the employed filters before integrating them into the search query.

To be able to reintegrate a workflow, the externalized subgraph needs to be self-sufficient and therefore consist of a defining source, one or more result sinks, and each node on the path from the source to each sink. Here, 'self-sufficient' and 'defining source' mean that the content of the result sinks is only dependent of the source's content and the filter definition of the nodes on the paths. Specifically, only the source node may have incoming edges from nodes that are not part of the subgraph. Each other node only has outgoing edges or edges coming from the subgraph's nodes. Because the filter/flow graph of the *PatViz* system has only one root node, a trivial solution for this requirement exists in taking the whole graph. But in order to maintain a collection of reusable workflows, the persisted subgraph should be minimal in the sense of modular designs in software engineering. The subgraph should have a high cohesion, i.e. only contain relevant filters, and be loosely coupled with other parts of the analysis graph, only interfacing through the source and result nodes.

If a persisted subgraph is reused in a new scenario, it is attached to the new graph by replacing the subgraph's source node with a user-selected anchor node. The new data flows through the imported subgraph and the filter criteria at each node are evaluated. Afterwards, the imported nodes are treated as any other node of the original graph and the user can branch-of new sets at any location, not just the originally intended result nodes. This

can be seen as a ‘copy and paste’ approach to reuse. For the sake of simplicity and creating a better separation of concern, one could alternatively import the subgraph bundled as a single filter node. However, due to the inability to distinguish multiple outgoing flows of these representative nodes, one could not handle multiple result nodes in the persisted workflow, at least not without changes to the interaction design of the selection management component. Also, the copy and paste approach allows to modify filter parameters within the imported subgraph. In the *PatViz* system, a filter setting, such as a date range, applicant name, or country, can be changed directly at the filter node using appropriate interactive elements such as range sliders or sunburst element selection. This way, one can persist parametrizable subgraphs and obtain a certain generality when managing a library of workflows, similar to the saving of often used search queries as templates with variables, as it was done in the *PatViz* system [Koch, 2012, p. 51]

3.5 Real-Time Streaming Data

Social Media providers such as Twitter generate near real-time data streams transferring hundreds of millions of messages per day.[11] Especially in the case of Twitter, these short messages often are status reports about something the users have seen or experienced. Some of them report from very noteworthy events such as crises, disasters, emergencies, etc. During the 2011 England Riots, users shared information within Tweets such as “Where are the police? Debenhams Clapham Junction is just being freely looted – disgusting.”[12] Due to the proliferation of mobile devices and location-based services, an increasing portion of messages is tagged with the location from which it was sent. When combined, these geolocated eye-witness reports are a valuable resource for crisis response and disaster management tasks [Judex and Zisgen, 2013]. Of course, the majority of the stream will be unrelated to an ongoing situation and has to be regarded as noise. Even the potentially relevant messages have to be challenged because they are coming through an unfiltered public channel. Neither human effort nor computational power alone suffice to process this enormous volume of dynamic data, but a combined visual analytics approach can tackle the challenge.

The facets of the approach presented until now are also not sufficient to process this real-time, massive-volume data treasure in an interactive way that allows an analyst to establish a meaningful situational awareness from it. They were targeted on static data sets which, at most, could be exchanged entirely with a subsequent re-evaluation of all filters employed in the selection management graph. Although a post hoc analysis of a situation is of course conceivable, like in the VC’11 scenario, a real-world situation would need to account for up-to-the-minute reports when managing the emergency response efforts. Therefore, an analysis system capable of dealing with streaming data sources, has to support incremental updates to the data base and emphasize the ‘flow’ in the filter/flow metaphor instead of managing selections. At the same time, the use of raw user-generated content as input for obtaining situational awareness increases the requirements on the employed (semi-)automatic filters and preprocessing compared to the curated patent document domain or artificial challenge data sets. Additionally, being a fast-paced and ever-changing medium, the content describing the current situation likely poses a very different analysis context than the messages that

[11] http://articles.washingtonpost.com/2013-03-21/business/37889387_1_tweets-jack-dorsey-twitter

[12] https://twitter.com/statuses/100667179489431552

were once used to configure the employed filters. This necessitates ways for the analyst to adjust the filters to unforeseen changes on the fly.

The following sections describe how these challenges can be met to create an analysis system that is capable of ingesting vast amounts of real-time social media messages and organize them in an ad hoc modifiable and filter/flow-based classification in order to obtain situational awareness. It focuses on the incremental nature of real-time data streams and on a separation of concerns between the generation of filters and their use by non-expert analysts. For this purpose, the section is further structured to first introduce the real-time situational awareness scenario and its requirements. Then, *ScatterBlogs2*, the new analysis system in which the modified filter/flow approach was embedded, is described. Because brushing and linking as well as selections and highlights are no longer the central means of interaction with *ScatterBlogs2*, we introduce tagging as a generic interface between our approach and the domain software. Separating the creation and the use of content filters allows for exploiting sophisticated filters while keeping their usage simple. Finally, a use case is presented to showcase the feasibility and usefulness of the approach.

Parts of this section have been published in:

- H. Bosch, D. Thom, F. Heimerl, E. Püttmann, S. Koch, R. Krüger, M. Wörner, and T. Ertl. ScatterBlogs2: Real-time monitoring of microblog messages through user-guided filtering. *IEEE Trans. Vis. Comput. Graphics*, 19(12):2022–2031, 2013
- G. Andrienko, N. Andrienko, H. Bosch, T. Ertl, G. Fuchs, P. Jankowski, and D. Thom. Thematic patterns in georeferenced tweets through space-time visual analytics. *Computing in Science & Engineering*, 15(3):72–82, 2013

3.5.1 Real-Time Situational Awareness

Using social media content for situational awareness has become a topic in the public perception since the global media coverage of the 2010 Haiti earthquake. Calls for donations and situation reports were distributed by social media channels. These channels are very heterogeneous and vary in their content types and ways to access them. Facebook alone has publicly accessible content, private conversations, and posts intended for a limited audience. It allows the sharing of text, images, video, and links to arbitrary

content on the web. In this section, we focus on Twitter as one prominent example of a social media service because it is a broadcast medium and features real-time access via a streaming API.[13] The vast majority of its content are public messages from individual users that are freely accessible to anyone. This greatly diminishes privacy concerns because we only process information that was intentionally published openly by the users, discerning us from other, more problematic, scenarios.[14] Nevertheless, analyzing and relating large data sets of this nature might reveal more information about the users than they intended to share by each individual message. Thus, we strive to anonymize the collected data thoroughly.

Because of the ease of accessing Twitter, several research projects have already made use of the data and have shown that they are capable of relating artifacts within the data set to events within the real world. These events may be human-made [Heverin and Zach, 2010; Hughes and Palen, 2009], natural catastrophes such as hurricanes [Hughes and Palen, 2009] and earthquake [Mendoza et al., 2010], or epidemics [Chew and Eysenbach, 2010]. Sakaki et al. [2010] successfully calculate the epicenter of an earthquake from the delays of related messages. Zhao et al. [2011] used topic models to establish a connection between Twitter and traditional news.

These systems, as well as our filter/flow approach presented so far, work on fixed-sized data sets and omit the real-time character of the source. Several recently-proposed visual analytics approaches address real-time microblog analysis, often by combining interactive exploration and anomaly detection. Twitcident [Abel et al., 2012] uses web-based analysis to connect twitter messages with news feeds from emergency responders. Twitinfo [Marcus et al., 2011] automatically detects and labels unusual bursts in real-time Twitter streams. LeadLine [Dou et al., 2012] connects events in the message stream with recognized entities. Whisper [Cao et al., 2012] visualizes geosocial information diffusion and Senseplace2 [MacEachren et al., 2011] provides an integrated geovisualization environment to filter and automatically localize messages and events based on textual content.

The related work allows analysts to follow information diffusion, analyze topics, and enables them to relate messages to each other, to events, or to locations. However, some pieces are yet missing to achieve situational

[13] `https://dev.twitter.com/docs/streaming-apis/streams/public`

[14] For instance circumventing the user-intended visibility constraints either by having access to the service provider's infrastructure (see Edward Snowden's surveillance disclosures) or being the service provider (`http://arstechnica.com/security/2013/05/think-your-skype-messages-get-end-to-end-encryption-think-again/`).

awareness and oversee an ongoing event thoroughly. Making sense of the social media data stream is a hard task for four main reasons.

- The message content is unstructured and the data volume is large.
- Few important messages may be hidden in the majority of unrelated ones.
- Making sense of already collected related messages has to be done at the same time as scanning the stream of arriving messages.
- A set of messages from one point in time and space may be no longer representative at a later point in time or different location.

The first point demonstrates the need for a visual analytics approach. While the data volume is too large to analyze manually, building filters that help in categorizing the incoming data automatically is also difficult due to the unstructured nature of the content. Approaches based on starting the analysis with an initial set of potentially useful keywords and improving the filters iteratively, similar to the *PatViz* system, bear the danger of missing relevant portions of the stream. Iteratively improving keyword lists draws much of the attention on the analysis of collected data that is also needed to follow the constant updates. Additionally, the use of language in social media often seems peculiar to a person not involved in an ongoing discussion and guessing the correct keywords is not assured. A better recall can be expected if algorithms try to learn the appropriate keywords from the data itself. Here, supervised learning methods are likely to outperform clustering based approaches due to the skew in the proportion between relevant messages and unrelated noise. Clustering requires a sufficiently large number of messages to identify clusters, which is likely to delay the detection of low-frequency incidents, rendering them unsuitable for live-monitoring situations. However, supervised learning needs training data and creating it requires time, which is often missing while assessing an ongoing situation. The changing spatiotemporal context makes it difficult to use already prepared training sets from older sessions, too. When these training sets are created on older instances of events that feature different toponyms, temporal term usage trends, and spatial language varieties, their performance on other events is impaired. Examples for these differences are the names of weather phenomena like Hurricane *Irene* that will have a high frequency in the old events but will not occur in new ones. The learning algorithm may put too much weight on these terms, or miss neologisms that frequently appear in form of hashtags for current events, e.g. the tag `#earthquakepocalypse`, which

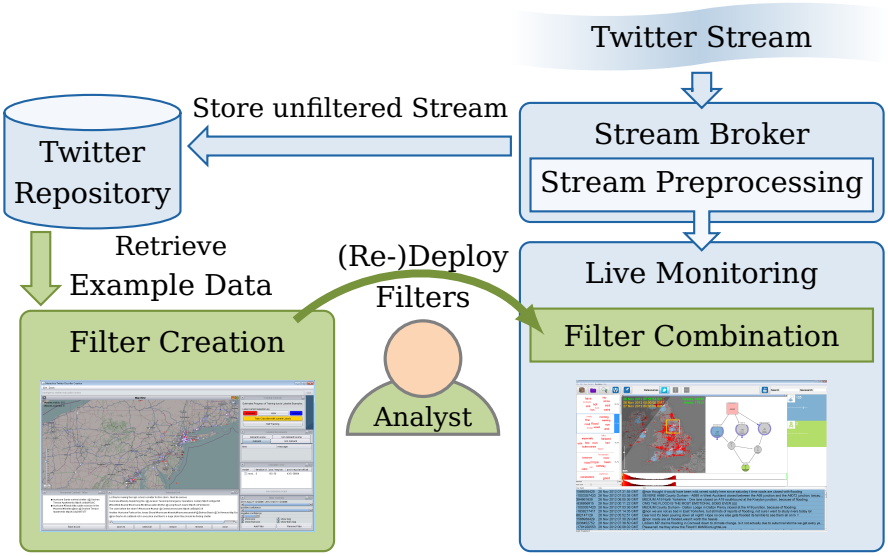


Figure 3.12 — Schema of the overall real-time monitoring approach. The life stream is recorded in a repository to support the creation of statistically motivated filters. It is also preprocessed and visualized for the live monitoring session. Here, preprocessing includes a stream-enabled term anomaly clustering approach to identify potential event locations and related messages. Elements of the schema denoted in green are directly influenced by the analysts.

was coined by the location based service Foursquare[15] during the 2011 major earthquake on the United States's east coast.

ScatterBlogs2, is a visual analytics system that facilitates sense-making of geolocated messages using a novel approach (see Figure 3.12). It relies on classification based filters that are trained on historic data and made generic by cleaning instance related characteristics like the aforementioned language varieties. To adjust the filters to the specific details of new situations during live monitoring, they are orchestrated and augmented with ad hoc geospatial and term based filters in a user-defined filter/flow graph. The graph thus provides a structure to organize the incoming data stream and can tag each message that arrives at certain nodes with a user-selected tag. The data is visualized as message density on top of a geographical map that accentuates: newly arriving messages, tagged messages, and automatically detected poten-

[15] https://foursquare.com/v/earthquakepocalypse-2011-east-coast-ny/4e53e9bda8097c30ee85740d

tial events that create anomalous spatiotemporal term counts. The message density is calculated over a temporal sliding, window which places the new messages in the context of the local history. Untagged messages therefore become stale and vanish after a situation dependent, user-selected time. The map is also the background for inspecting spatially close message collections using a focus+context technique that summarizes the message contents.

In order to understand Twitter data, one has to be aware of some of the conventions that were formed by the community to enhance the original plain-text service with links and mechanisms to organize conversation. The most familiar ones are hashtags, which act as links and keywords to group messages of a similar topic. Here, the topic bearing word in the message is prefixed with the "hash" sign (#). Similar, twitter usernames mentioned in the message are prefixed with an "at" sign (@). They link to the mentioned user's profile and the message is listed on this user's Twitter page. Links to other websites are usually shortened by using redirecting services with very short URLs to save on the 140 character length limitation of Twitter messages. Finally, other messages are quoted as a 'retweet' by copying its content and prefix it with "RT" and optionally the username of the original author. The Twitter data stream, which ScatterBlogs2 processes, provides not only the textual content but also additional entity sets from preprocessing and resolving the above mentioned links and annotations.[16] This metadata also includes the author, timestamp of creation, geocoordinates, source client, information whether the message is a retweet, and much more. For each involved user it contains the user id and name. For the author, it additionally contains the information provided in the user's profile, such as home location, time zone, status, etc. For our purpose, the author, message location, and textual content are sufficient. The author is needed to exclude spamming users during the automated event detection and is stored pseudonymized as a hash value. Additionally, all mentioned users are replaced by a constant placeholder value as an additional measure for maintaining information privacy while processing mass data.

ScatterBlogs2 listens for geolocated messages by using Twitter's filtered streaming API. Except from a small percentage due to rate limitations imposed by the service provider and short-term connectivity outages, we collect almost all of the geolocated messages of the Twitter network. If the geolocation of a message was not attached by its author, a method for inferring it through textual features or user histories was developed [Thom et al.,

[16] `https://dev.twitter.com/docs/platform-objects/tweets`

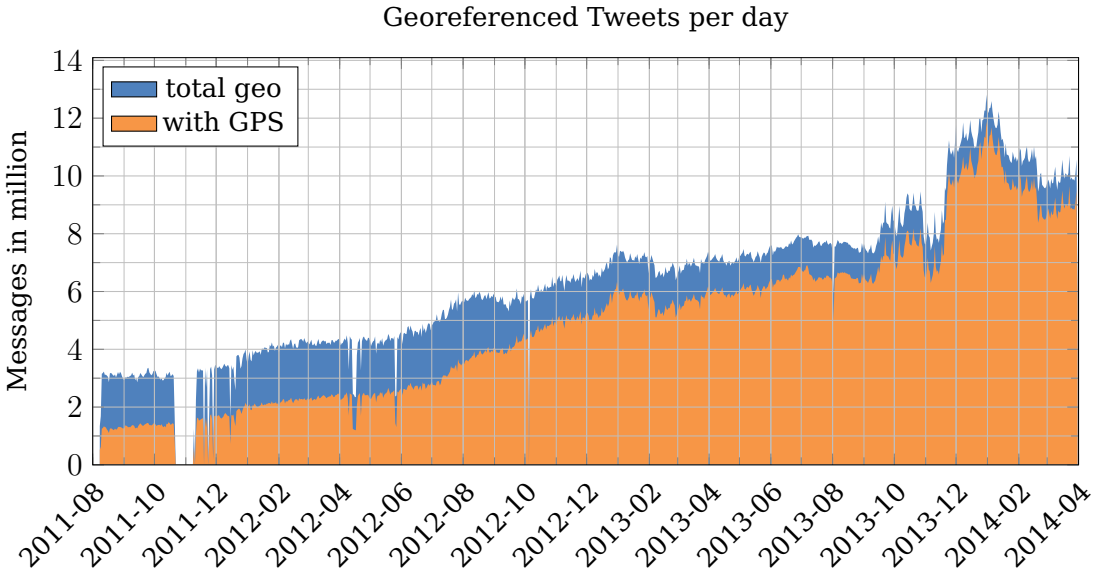


Figure 3.13 — Daily data volume captured by monitoring the stream of geolocated Twitter messages. Small gaps are due to recording interruptions. The blue band shows the amount of messages that were not georeferenced by an exact GPS position but by a toponym such as `Paris`.

2012a]. Since the beginning of our monitoring, the amount of geolocated messages increased, both absolute and relative to the total Twitter volume (see Figure 3.13) and, at the time of writing, totals to about ten million per day.

3.5.2 Ad hoc Customizable Filters using Classification

The stream of messages is constantly recorded into a repository of historic Twitter data, even if no current situational-awareness monitoring takes place. This provides us with the opportunity to learn the media-specific characteristics of events from their recorded occurrences. The benefit of learning the characteristic from actual data over user-defined keyword lists is an increase in recall because it can capture a greater variety of relevance-indicating features from the data and thus generalize better. However, this comes at the price of time-consuming labeling activities and potential overfitting. The former is necessary to create the training data for the supervised learning algorithms. The latter results from learning event-indicating keywords from only one recorded event. For instance, if the algorithm learns to identify

hurricane-related messages from a data set collected during Hurricane Irene, it will erroneously learn that ‘Irene’ is a valuable keyword for detecting relevant messages during other hurricanes.

Due to the fast-paced nature of the medium, the representativeness of historic data for future events will deteriorate over time. Additionally, the volume of the data stream during an investigation might dictate the appropriate selectivity of the filters. Therefore, available filters need to be adaptable to new situations and should provide analysts with means to interactively trade recall for precision, thus enabling them to concentrate on the most relevant messages in time of data abundance (high precision) or investigate the most promising candidates in the absence of good hits (high recall).

We address these problems with two approaches to define statistically motivated filters. They share the following properties:

- Interactive and iterative labeling,
- intermediate result previews to judge the resulting performance,
- data-driven labeling guidance,
- adaptable filter selectivity,
- and a careful selection of training data covering at least two occurrences of an event and also unrelated messages.

The difference of the approaches are their intended use because of the required preconditions on the data set and their results as well as their labeling target. In the first approach users select relevant keywords from a data-driven keyword suggestion and thus label the messages indirectly through their contained keywords. In the second approach, they label messages directly while an active-learning component suggests promising labeling candidates for an efficient filter creation.

Keyword Mining for Weighted Filters

In the context of social media analysis, it can be difficult to identify keywords manually, as term usage in social media can differ from analysts’ expectations. Using past events, the analyst may find new interesting facets and thereby identify relevant terms to be included into the filter. For this purpose, we have built a software that guides the analyst in the process of defining a weighted keyword list from an initial set of potentially relevant keywords and a large data collection. The process is shown in Figure 3.14 and the user interface in Figure 3.15.

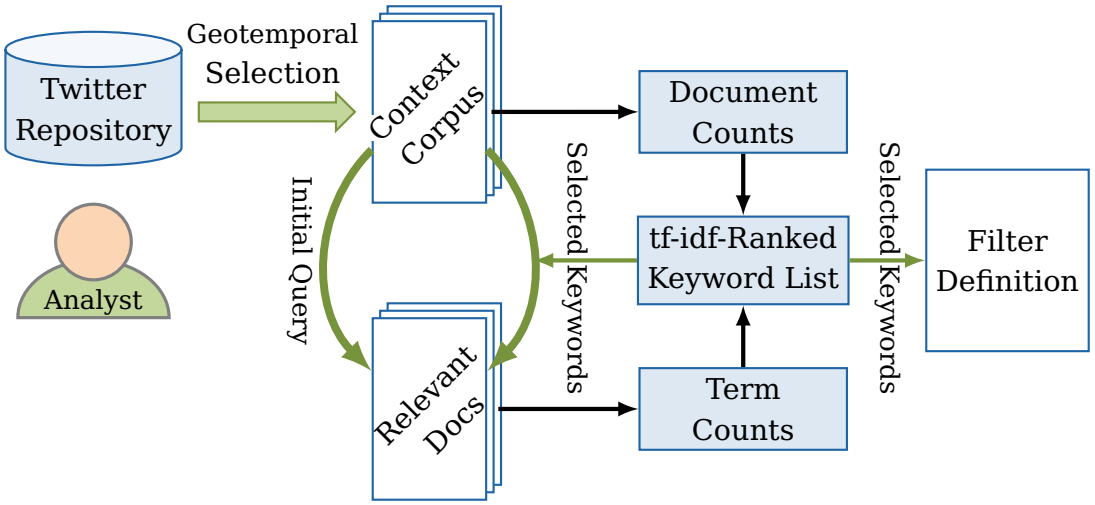


Figure 3.14 — Schema of tf-idf-based keyword mining and filter creation. A coarse geotemporal selection, taken from a historic event establishes the context for the keyword mining approach. The analyst defines an initial set of related keywords to retrieve relevant messages. Comparing the term/document frequencies a keyword ranking is derived and presented to the analyst for a manual selection of additional keywords. This cycle is iterated till a satisfying collection of keywords is produced and a filter definition based on their tf-idf weights is created.

As the first step, a message corpus has to be defined. It should contain relevant as well as irrelevant messages in order to learn how to differentiate the two classes. The corpus can be established quickly by the temporal and geospatial extent of past events applied as filter on the repository of recorded messages. The basic idea for identifying relevant keywords from the corpus is to extract potentially relevant messages and assess if certain terms are used substantially more often in the set of relevant messages than in the base corpus. For this purpose, we rank each term t based on the tf-idf-inspired weight w_t by comparing their frequency $tf_{t,R}$ within relevant set R against

their document frequency $df_{t,C}$ within the base corpus C.[17]

$$\begin{aligned} w_t &= \frac{\log(tf_{t,R})}{\log(df_{t,C})} \\ tf_{t,R} &= \text{Count of } t \text{ in } R \\ df_{t,C} &= \frac{|\{m \in C \mid t \in m\}|}{|C|} \end{aligned}$$

Here, the definition of the set of potentially relevant messages is of course the original problem we tried to solve. Therefore, the analysts have to start by supplying an initial set of keywords for their information need and every message containing one of them is taken into this set. The term weights w_t are then calculated with this initial configuration and the top scoring terms are presented to the analysts as suggestions for improving the list of relevant keywords. The weights are not only used for ranking the most promising terms but are also stored with the resulting keyword lists as a relevance indicator for the filter. When applying the filter during the monitoring phase, a user-configurable threshold defines a minimum weight for messages to be included in the filter result. Here, a threshold of 0 would include every message, while a threshold of 1 would only include messages that consist only of terms with the highest weight. This fulfills the aforementioned requirement to adjust the selectivity of filters during the situational awareness monitoring.

The resulting filter can be applied to the base corpus to preview its performance and judge if the result fits the information need of the analyst. The definition of potentially relevant messages so far only depends on the rough initial query. Therefore, it is possible to iteratively refine this set by widening it with the current filter result and recalculate the weights. This also re-ranks the suggested keywords from which the analyst can again update the keyword list, thus closing the feedback loop.

Figure 3.15 shows the interface with the ranked lists for keyword suggestions for single terms and co-occurring term pairs. Especially in English, the single parts of composite words can lose much of their meaning when considered in isolation. For instance, the term `shot` is only relevant for epidemic analysis in bigrams such as ‘flu shot’. It is therefore important to include co-occurring terms as an element in the filter definitions. Otherwise, the

[17] Due to the length limitation in Twitter non-stopword terms mostly occur only once per message, i.e., the term frequency over a message set is almost equivalent to its document frequency over the same set except for the normalization.

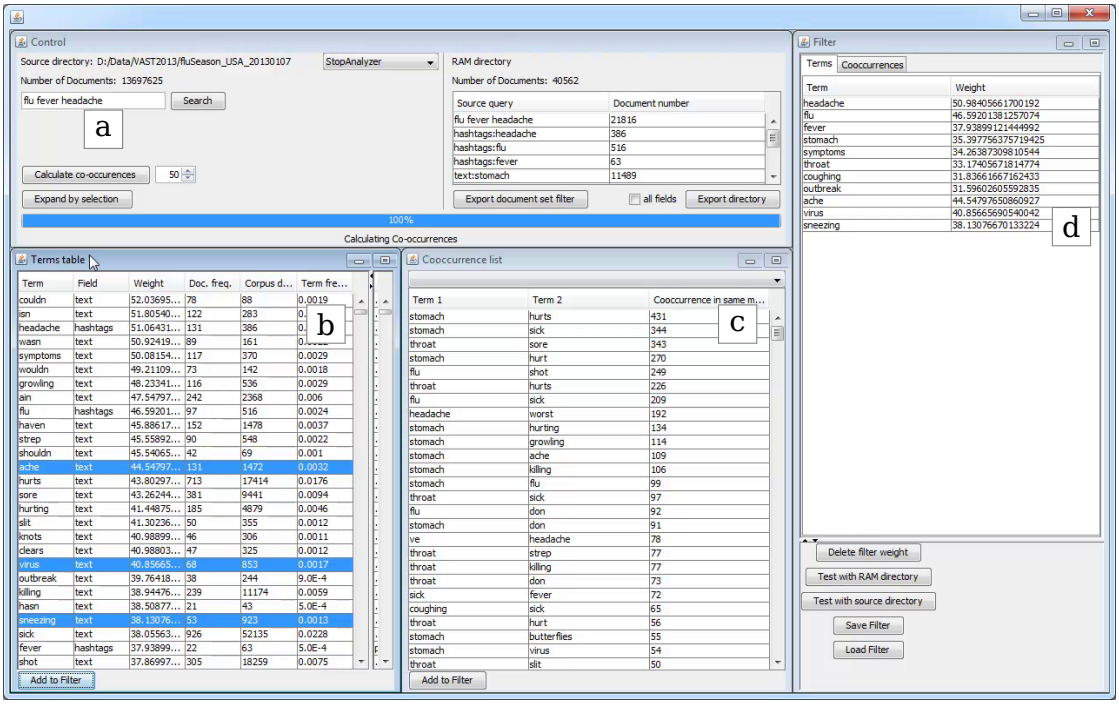


Figure 3.15 — Screenshot of the keyword mining software with: (a) the initial keyword query, (b) the ranked keyword list, (c) the ranked co-occurrence list, and (d) the list of keywords selected for the filter definition.

potentially relevant message set may contain too many false positives (e.g. shot glass) to deduct meaningful keyword weights, thus impairing the filter's performance. The table on the right part of Figure 3.15 lists the collected keywords and their weights that would be used for the filter in the current state.

The data-driven keyword suggestion is a valuable instrument for creating streaming-enabled filters because it can inspire the analyst's imagination about an event and it is scalable to the data volumes of today's social media providers. Starting with an initial idea, i.e. keywords, about an event type, it provides suggestions that are refined with each iteration of user feedback. Working only with counts and frequencies, it does not need the plain text of the message set, but only the textual index that links terms to message IDs and counts. Table 3.1 shows selected entries from a suggestion of hurricane related terms. Even when taking multiple instances of an event type as input data, in this case Hurricane Irene and Hurricane Sandy, the names of singular events are still very frequent in the relevant message set and appear in the

suggestion list. It is therefore important to let an analyst decide based upon this suggestion, which terms actually should be taken to the filter definition and which not.

Interactive Learning for Classification-based Filters

In our second approach to define streaming-enabled filters, the task of finding terms relevant to an event is replaced with the task of providing suitable examples for training a binary message classifier. Binary classifiers try to separate a set of entities in two classes based on their features, in our case separating relevant from irrelevant messages based on their textual content using the bag-of-words model.[18] The biggest problem of using classification for filter definition is the need for training data which has to be labeled manually by a domain expert. However, by choosing the right classification algorithm and efficiently labeling only the most 'informative' entities, one can mitigate this problem. For this purpose, the Support Vector Machine (SVM) framework [Vapnik, 1998] is used, which is known to perform well on textual data. SVMs try to establish a maximum-margin hyperplane between two classes in a high dimensional feature space. The hyperplane's definition is depending on the subset of training instances that touches the area of the resulting margin, the support vectors. *Active Learning* strategies exploit this

[18] Each document is represented as a sparse vector containing the term counts of each indexed term. Its dimension equals the size of the term dictionary.

term	weight	frequency relevant set	frequency corpus
tropical	66,5	1 750	4 256
hurricane	64,9	69 714	93 716
storm	63,8	423	830
irene	60,9	11 696	47 381
surge	60,2	431	1 537
evacuated	50,0	165	1 759
staying	45,0	264	9 879
place	40,0	519	52 856

Table 3.1 — Selected top, medium, and low weighted terms from a set of messages during the hurricanes Sandy and Irene. The set of relevant messages was defined by searching for `storm` and `hurricane` and is a subset of the corpus.

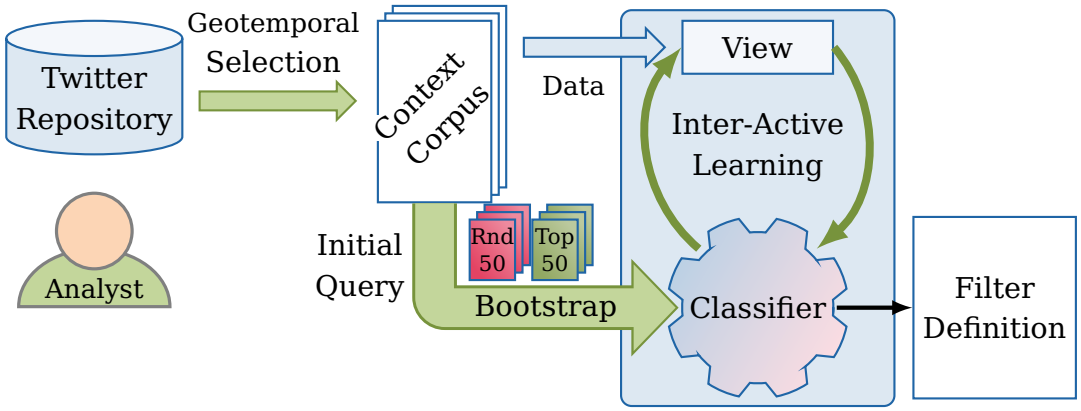


Figure 3.16 — Schema of the interactive classifier training for microblog filters. A coarse geotemporal selection, taken from an historic, event establishes the training and validation corpus. An initial query for related messages bootstraps the classifier by automatically labeling the best 50 hits and 50 random messages as relevant and irrelevant, respectively. The performance of the resulting classifier is visualized using the message corpus and enables the analyst to correct existing and add new labels, in an active learning inspired way, to increase the training data. This is repeated until the desired performance is achieved and the classifiers model is persisted as a filter definition for the live monitoring.

by trying to identify potential support vectors that are still unlabeled and query the user for a label. Uncertainty sampling [Settles, 2012] is one of these strategies that is known to work well for SVMs [Tong and Koller, 2002] Here, the candidate which currently has the lowest classification confidence, i.e. that is closest to the hyperplane, is selected. An additional benefit of SVMs is the availability of a classification confidence. Each entity in the vector space has a computable distance to the hyperplane. The larger the distance, the higher the confidence on the classification result. This allows the definition of an adaptable threshold to influence the selectivity of the filter during its application in situation monitoring. This way, the analyst can either include additional messages classified as irrelevant but with a low confidence (higher recall), or reject low confidence results even if they are classified as relevant (higher precision).

The classifier training software of *ScatterBlogs2* is inspired by Active Learning but provides more control to the expert user for interactively choosing labeling candidates. More information about this approach and its com-

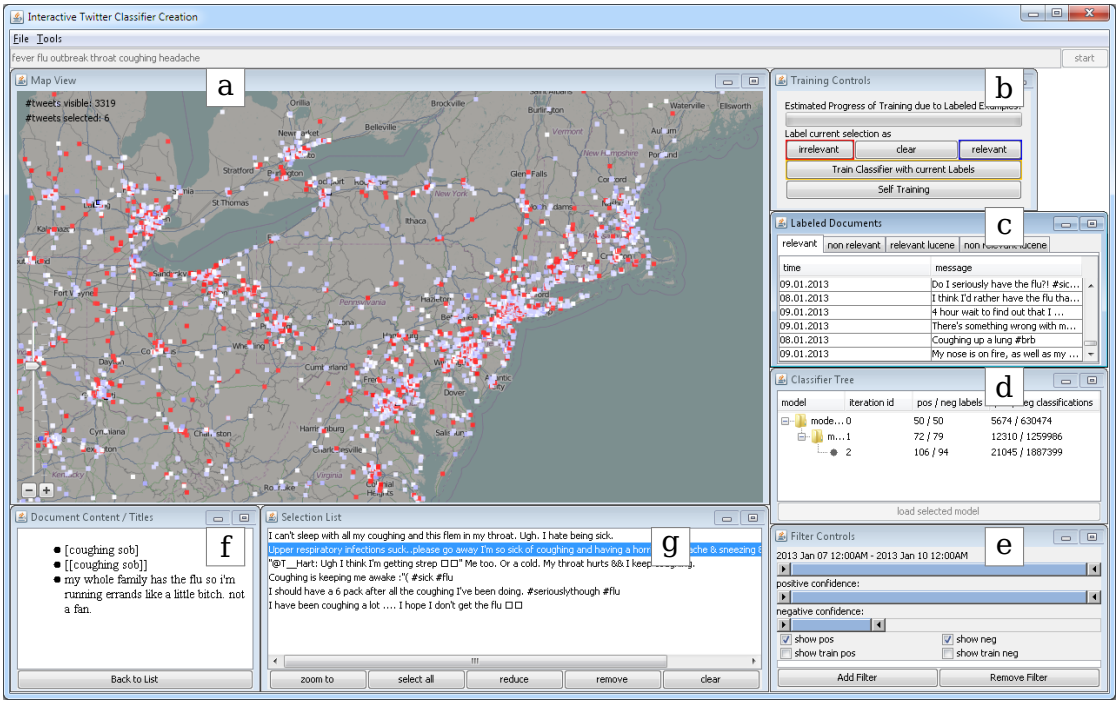


Figure 3.17 — The desktop for interactive message classification training: (a) *Map View*, pan- and zoomable world map indicating position and classification score of the messages; (b) *Training Controls*, used to label messages and proceed to the next training iteration; (c) *Label History*, list of already labeled messages; (d) *Classifier History*, history of previous state of the classifier for undo operations; (e) *Filter Controls*, to filter the currently shown message in the map; (f) *Document Summary*, preview on the message contents; and (g) *Selection List*, list of messages currently selected as labeling targets.

parison to classical active learning can be found in the related evaluation paper [Heimerl et al., 2012]. In *ScatterBlogs2*, it combines free exploration of the data with spatiotemporal, textual, and classification-confidence filters to select messages for bulk labeling. The confidence filters are essential to allow an uncertainty sampling strategy, while the others help in separating relevant from irrelevant messages. Similarly to the keyword-based filter creation, training the classifier is an iterative process and starts with a recorded set of messages from previous event occurrences (see Figure 3.16). The interface is designed for inspecting the current classifier and, based upon its output, select new training instances. Therefore, it needs an initial classifier configuration which is obtained by bootstrapping. Here, highly ranked messages

from the result set of a coarse keyword query are used as positive examples and some arbitrary documents not returned with the result set serve as negative examples to train the initial classifier. During each training iteration, new messages are labeled, and a new, updated classification model is created and visualized.

Figure 3.17 shows the interactive visualization of the data and classification result. The Map View (Fig. 3.17a) depicts messages as small, colored glyphs at the location they were sent from. Messages classified as relevant are colored in blue, those classified as irrelevant in red. Classification confidence is encoded by brightness with higher brightness meaning lower confidence. The Filter Controls (Fig. 3.17e) offer assistance in extracting relevant messages. They include a range slider for constraining the publication timestamp of the messages displayed to show only messages published during the known time frame of an event. The Map View in combination with the temporal filtering enables analysts to find past events by their spatiotemporal extents. Two additional sliders let analysts restrict the confidence range of positively and negatively classified messages on the Map View. Furthermore, information on low confidence classifications is useful to judge the quality of the current classifier, and thus helps to assess training progress. The Filter Controls further allow to turn the display of positive, negative, and training examples on and off separately and to filter messages by keywords. The latter is useful to find messages containing a certain hashtag or place name. The other views contain controls to advance and track the training iterations (3.17b-d), and to inspect message contents and shape the current selection for the next labeling action (3.17f+g). Among these are a history of already assigned labels and previous classifier states to undo the latest iteration in case of undesired outcomes. The Selection List (3.17g) can be used to inspect a set of messages and prune it to a homogeneous selection to which a single label can be assigned collectively.

In its basic form, the SVM's hyperplane defines a linear separation of the data. If the classes are not linearly separable, one usually performs a non-linear transformation to the feature space, sometimes referred to as the 'kernel trick' [Schölkopf and Smola, 2002]. This increases the dimensionality of the feature space, allowing more degrees of freedom to find a separating hyperplane. Due to the high dimensionality of the bag-of-words feature model, this is rarely necessary for large, textual data sets. However, the noisy nature of microblog messages may cause problems when applying only term-based classification methods because they may contain typos or colloquial language. Therefore, a string kernel [Lodhi et al., 2002] can be used optionally, which

compares message strings to assess their similarity. The kernel's computation complexity, however, is much higher compared to the linear model leading to longer training and classification times.

Comparing the two approaches, the keyword-based and the classification-based approaches have slightly different characteristics. Independently of the selected kernel, the classifier filters need access to the textual contents of the training set instead of considering only message counts. This renders the classification approach less scalable during the training phase, but it is still capable of using a corpus of roughly a million messages. The complexity of evaluating the filter result during monitoring is the same for both approaches (summing term weights) considering the linear kernel. On the one hand, the classification filters are, in principle, capable of producing filters with higher precision, because the linear combination of support vectors can also assign negative term weights. On the other hand, the keyword-based approach is more transparent to the users as they can inspect and adjust the term weights and a filter match could be explained easily by highlighting the matching terms. While the string kernel is the best fit to the noisy nature of the data source, it is not capable of processing the full data stream in real-time.[19] Nevertheless, it is valuable as a high precision filter when orchestrating them (see below) with upstream filters of high selectivity to reduce the stream's volume, e.g. using narrow spatial or broad keyword filters.

3.5.3 Monitoring Environment

The monitoring environment of *ScatterBlogs2* is shown in Figure 3.18. It is a multiple-views environment designed for (*a*) monitoring the incomming stream of Twitter messages, (*b*) exploring the content of message sets, (*c*) orchestrating filters for processing the data stream. For this purpose, it comprises the *Map View*, *Timeline*, *Tag Panels*, *Content Table*, *Topic View*, and of course the *Filter Orchestration Graph* based on the selection management approach.

Monitoring the Stream

Its central component is a pan- and zoomable map based on OpenStreetMap[20] providing the context for displaying geolocated microblog messages. The

[19] Real-time in the sense of processing the data at faster rates than it is produced, thus keeping up with the stream.

[20] `http://www.openstreetmap.org`

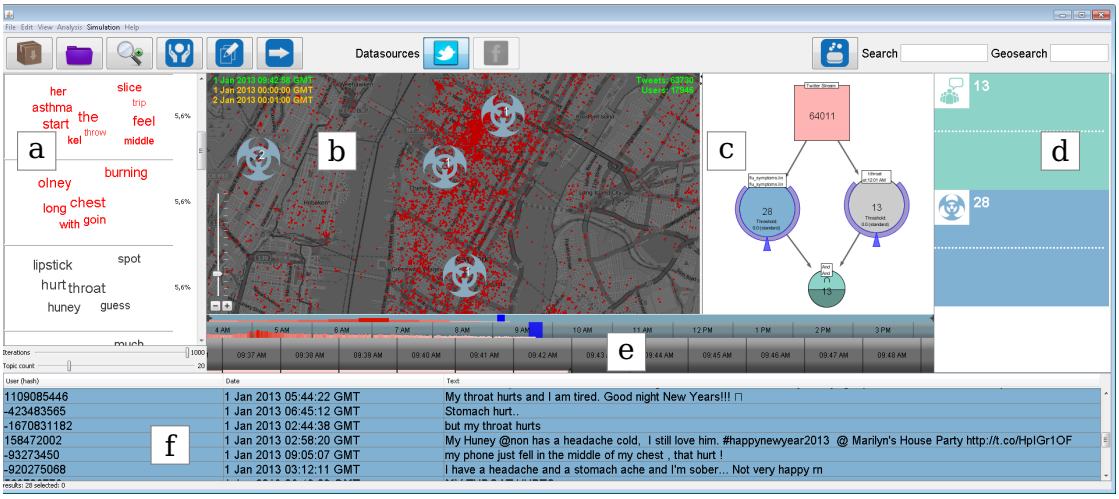


Figure 3.18 — The ScatterBlogs2 desktop for real-time monitoring: (a) *Topic View*, topics extracted from the currently selected messages based on topic modeling; (b) *Map View*, pan- and zoomable map with message locations and search hit cluster icons; (c) *Filter Orchestration Graph*, the filter/flow component to orchestrate ad hoc and pretrained filters and tag the outcome; (d) *Tag Panels*, a panel for each tag indicating search hits during the last minute; (e) *Timeline*, multi-scale, multi-layer timeline for temporal navigation; (f) *Content Table*, table containing the raw data of currently selected messages.

message data is visualized in three layers on top of the map:

L1. The locations of all recently received messages are marked by scattered dots, increasing in opacity if multiple messages hit the same pixel on the map. It is the basis for selecting message sets to explore their contents and gain detailed insights on the situation.

L2. The latest updates are highlighted by bright yellow points that slowly fade into the persistent layer (L1). It shows the analyst that the message stream is currently running, informs about the intensity of data flow, and indicates if the data within the region of interest was updated.

L3. Spatially dense message concentrations that feature an abnormal term usage characteristic are labeled by their common term to highlight potential events [Thom et al., 2012b,a]. The terms are obtained by a clustering approach, which provides a contrasting view to the analyst's

specifically trained filter configuration. It thus helps analysts to obtain further context information that may not be captured by their message filters but might be related to their task, e.g. that a music festival is taking place in an area that might be affected by upcoming severe weather. The prototype was extended to include an additional anomaly detection based on the remainder component of a seasonal trend decomposition of extracted topics [Chae et al., 2012].

To allow for more flexibility in the data visualization, the original map tiles of the provider were desaturated and reduced in contrast. Below the map view, a hierarchical timeline histogram depicts the message distribution over time on several layers of accuracy [Wörner and Ertl, 2013], i.e. days, hours, and minutes. The layers are linked so that the visible range of the lowest level is marked at the overview levels. The Timeline View allows to set the current time to replay elapsed events.

Content Exploration

The monitoring environment supports three basic filters in order to select messages for exploring their contents. If the analysts are interested in a certain region, they can sketch spatial filters by drawing a polygon on the Map View, which selects all messages from that region. Additionally, they can perform textual searches with Boolean keyword querys to filter out messages that do not fit the query. Finally, the Timeline View also allows to mark time spans of interest to filter the message stream. All three basic filters are simultaneously evaluated and each message must pass all filters. For complex filter combinations, the filter/flow graph is available in an extra view.

Filtering messages triggers an update in the Map View, the histogram of the timeline, and the detail views that show the content of the messages, i.e. Content Table, and Topic View. The Topic View shows the result of an LDA-based [Blei et al., 2003] analysis of topics in the message set. Each detected topic is shown as a tag cloud of the prevalent terms that comprise the topic. Here, the size of a term corresponds to its weight in the respective topic and important terms tend to be shown in the middle (such *circular layouts* match the task of finding the most popular terms [Lohmann et al., 2009]). Since the capabilities of detail views are limited and, in the case of the Topic View computational intensive for large message sets, *ScatterBlogs2* adheres to a two stage filter and selection scheme. While the other views will instantly show the result of filter operations, the detailed views require an additional action of the user to update to the current filter set. The Content

Table contains the sender, timestamp, and textual content of the selected message.

Alternatively to the filter and selection scheme, analysts can also deploy the Content Lens on the map view. This is the same focus and context technique that summarizes all messages under the mouse cursor and its vicinity by showing the most frequent terms in a cloud as it has been introduced in Section 3.2.1.

Orchestrating Filters

Because the statistically motivated filters were trained on previously observed data, there is a need to adapt them to the current situation, either by changing their threshold value or orchestrating them with ad hoc filters. This need can arise under one of the following circumstances during a monitoring session: (*a*) The filters do not match any incoming message and the analysts want to widen their scopes to judge if there actually is no indication of an event or if the filters are misaligned and filter out relevant messages. (*b*) Over the course of the session, the monitored situation changes and details become apparent that require other filters or filter combinations to keep an overview. (*c*) Unforeseen anomalies in the data stream, such as trending topics, erroneously trigger important filters in the graph and thus give a false impression of the situation. The analyst has to include preprocessing filters to avoid a retraining of the filter just for this circumstance.

For this purpose *ScatterBlogs2* contains an adapted version of the selection management graph. Similar to the other versions of the approach, the filters that were used for exploring the contents in the MCV (spatiotemporal ranges and Boolean keyword queries) can be instantiated in the graph structure where they filter incoming entity streams of parent nodes and provide the results to their child nodes. Additionally, the analysts can load instances of the statistically motivated filters and include them as filter nodes in the graph. In contrast to the other systems, the nodes, their appearance, and their interactive elements have slightly different semantics. The graph has a single, persistent root node receiving the unfiltered incoming message stream. It is set apart from the other nodes by its rectangular shape (see Figure 3.18c). All other, user-added nodes are either filter or join nodes.[21] Filter nodes have an associated filter constraint that reduces the incoming data and provides the result as new input for other nodes. They can have multiple listening

[21] The *set node* of previous versions of the graph are now merged into the other two node types, which directly offer their result stream.

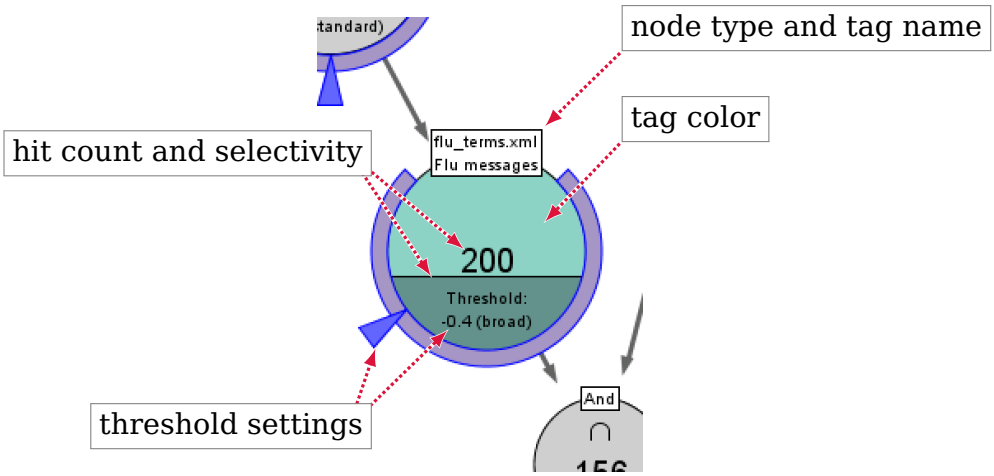


Figure 3.19 — A node in the orchestration graph with indications of the node type and tag name, the selectivity, tag color, and threshold settings.

child nodes but only one incoming edge. This will result in a tree structure. In order to form a graph and merge separate flows of the tree again, the join nodes can have multiple incoming edges and implement symmetric, n-ary operations, i.e. union, intersection and symmetric differences.

In total, four basic operations are needed when interacting with the graph editor: creating/deleting nodes, creating edges, changing thresholds, and tagging nodes. Nodes are created or deleted via a context menu. Connections between nodes are created by dragging the source node onto the target node. During the drag interaction, valid targets are highlighted in green and the creation of circular dependencies is prevented. If a specific filter supports the adjustment of a threshold value, it can be changed directly at the node using a circular slider in a normalized range $[-1; +1]$. The filter implementation is then responsible for translating this value into a meaningful threshold value that maximizes recall for the value -1 and precision for the value $+1$. For instance, the weighted keyword filter can directly map this value to its internal weight threshold, while the SVM classifier can use it to adjust the bias of the decision border.

Fig. 3.19 depicts a node and its context. The topology of the graph represents the flow of messages and the working sets of each filter node. The node itself shows the details of its configuration: Its type and tag name, if assigned, constitute a label for the node. The absolute number of messages that passed its filter are shown in the center of the node, while the ‘selectivity’

(percentage of incoming messages that pass it) is denoted by using a darker fill color for the appropriate portion of the node. Here, the color corresponds to the tag or is gray otherwise. Finally, the current threshold setting is marked with an indication in natural language of what the numeric value roughly means.

On the implementation side, the stream-enabled filter graph is represented as a set of message for each node containing all messages that passed the filter. Each child node listens to changes in the result set of its parent and reacts to changes by evaluating its own filter on the new content in a separate thread. Thereby, set changes propagate through the graph independently and in parallel execution.

3.5.4 Tagging as Generic Interface

Most multiple coordinated views environments either have a tight integration between the individual components by a special purpose event model, or a shared data model that handles selections and highlighting attributes. The integration of the orchestration component with the rest of the monitoring system is realized by *tagging*. A tag consists of a user-defined name, color, and icon and is assigned to a node via a context menu. If a document passes the filter of a tagged node, it collects the tag and will hence be marked throughout the system by the associated color and icon. As an attached property, the tagging does not change the data model or event infrastructure and is thus easily attachable to other domain software. Components that want to display the tag information can access the tag property and utilize the extended attributes.

In *ScatterBlogs2* there are two components that utilize the tag information. The Map View displays tagged messages with their associated colored class icon and aggregates similar icons in close vicinity of each other to one single icon, labeled with the count of aggregated messages (see blue icons in Figure 3.18b). The link between the tagged messages and the tagging node is established by using the same color and labeling it with the tag name. The Tag panel (see Figure 3.18d) contains one entry per tag and shows a histogram of message density over the last minutes for the specific tag. If a tag is assigned extensively and threatens to clutter the map display, it can be ‘minimized’ by clicking its tag panel. This changes the display style of this tag on the map panel to show only correspondingly colored dots instead of tag icons for each tagged message. Additionally, there is a special `current` tag, which, if assigned, functions as a visibility filter and hides all other messages

throughout the monitoring interface. This is helpful as a drill-down operation into a subset of messages for closer inspection.

Using tags as a generic interface was motivated by the real-time data streams that would trigger many events and data model changes. It replaces the management and distribution of message selections to each component and supports the use of multiple tags at once instead of one selection at a time. The tagging of a node's content, which is defined by all filter criteria along the paths from the root node, is the key element of describing and monitoring current events and situations.

3.5.5 Great Britain Flood Scenario

To better illustrate how each of these segments work together to create a visual analytics system for situational awareness, this section describes an analysis session that covers a severe weather period over Great Britain and Ireland in 2012. The weather event caused heavy rain and floods, resulting in numerous flood alerts, evacuations, road-blocking landslides, a derailed train, and even fatalities throughout the British Isles. The total insurance losses through flooding for 2012 have been estimated at over £1,3 billion [Impact Forecasting, 2012]. Typical questions that arise in such a scenario are: (*a*) Do the flood warnings reach the targeted audience and are they disseminated efficiently? (*b*) What is the current situation at the inshore waters and rivers? (*c*) Are there infrastructure problems, such as blackouts or road blockages that are not yet known to the responsible personnel?

In order to evaluate the applicability of our real-time monitoring environment, based on data generated during these events, all georeferenced Twitter messages from six days in June were collected. On these days, flooding events have been reported and this set was used to train a classifier to identify flood related messages.

The resulting filter as well as previously created, general purpose classifiers and keyword based metrics where then used to monitor selected subsequent days of the year. Based on a replay of the Twitter data stream collected during November 26 — a day on which severe weather conditions occurred — the following paragraphs describe in chronological order how an operator could have monitored and analyzed flood related messages during that day.

Preparation - In the beginning of the monitoring phase the operator summons an initial set of emergency related filters trained to detect severe weather effects, fires, damages, etc. To improve the effectiveness of the

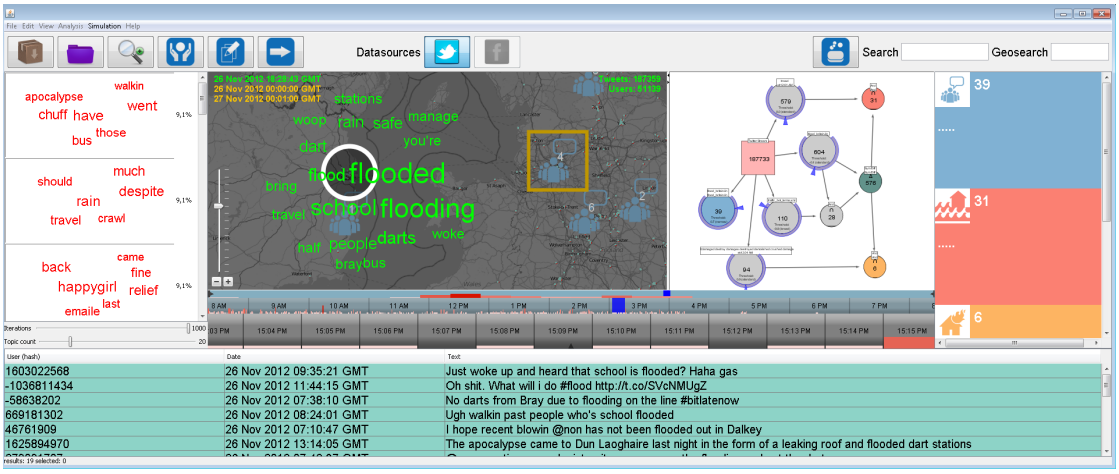


Figure 3.20 — Application of the monitoring environment during the Great Britain and Ireland Floods. The picture shows a more sophisticated filter graph that the operator has already constructed to find high-profile and low-profile flood related messages and separate them from spam and traffic information messages. In the map view we can observe the application of the Content Lens to the flood related filter context, indicating that schools and DART stations were flooded in Dublin.

default filters, the operator combines them with filters that will remove spam and news media, or other messages containing second hand information (e.g. retweets).

01:16 GMT - At this time, the flood classifier detects the first messages related to the flood. It seems that the government-issued flood warnings were successfully disseminated and are discussed by the public (`Flood warning has been issued for #caversham!`). Increasing the filters threshold omits mere repetitions of warnings and reveals a first indication of actual weather impact appearing at Rathdrum near Dublin (`Our yard is flooded!! Go away Rain!!!!!`). The operator continues with two instances of the classifier, one with a strict threshold for high profile messages and one with a medium threshold for monitoring the overall trend.

06:16 GMT - According to a local resident, river banks broke on the River Swake in Richmond. This is the second message detected by the high threshold classifier. Also, more and more messages of a traffic information service begin to appear within the detected messages. As such messages will mostly convey information already known to the operators they can hide them ad hoc with a keyword filter on the words used by the service and a

symmetric difference between the keyword and the flood classifier filter.

08:44 GMT - The situation begins to unfold as more flood indications start to appear all over the map. The aggregated classifier icons provide an impression of the distribution, but their increasing number hinders the further examination of events. Therefore, the operator creates more specific categories, such as road blockage related messages, by combining the classifiers with manual keyword and region filters.

10:21 GMT - Although unrelated to the flood, the default emergency filters instantiated during preparation show that a fire broke out in Oldbury near Birmingham caused by an explosion at a distillery. Several eyewitnesses talk about the incident and report on its severity (`Saw the explosion at the oldbury fire as I was driving past on the motorway...`) which results in a clear peak in the area compared to other parts of the country.

11:07 GMT - The map overview and road blockage classifier icons provide the operator with a good indication that traffic is hindered in southern and middle parts of the UK. Inspecting some of their messages by selecting an icon highlights reports on flooded roads (`Can't believe I drove down a flooded road where I couldn't see the sides or the end. Was like being in a boat.`).

15:41 GMT - Since the operator is already confronted with a very high quantity of flood related messages (about 579 for the default threshold classifier), it is now a good idea to get a general overview of what the people are concerned about in different parts of the country. The analyst thus selects the flood classifier and sets it as the current filter. Based on this filter context, it is now possible to apply visual aggregation tools like the LDA Topic View and the Content Lens in order to find topics connected to the flood. The exploration of the map quickly shows that in Dublin `school` and `dart` are prominent keywords among the flood related messages (see words around the circle in Fig. 3.20). By investigating these messages, the operator can quickly understand that the Sandycove DART (Dublin Area Rapid Transport) Station is flooded and that schools in the areas have been closed in the morning due to the flooding. Using similar means it is possible to see that several people complain about delayed or canceled trains because of the weather conditions in London and that the region of Worcester was severely affected by the flood.

The system helps the operator to keep an overall picture of the ongoing events, thus ensuring situational awareness. Although the operator was able to detect several smaller incidents and flood damages that affected people, it was also possible to recognize that the general situation stayed under control.

3.6 Uncertain Set Definitions

The last section in this chapter shows how the filter/flow metaphor can be applied to the problem of exploring uncertain set membership configurations. Here, every entity of the application domain is present in every node of the graph, but may take, depending on the applied filters in the graph structure, different roles. The graph structure allows for a free exploration of filter combinations with immediate feedback. The scenario for this use case is again taken from the VAST Challenge, this time from 2009, and it is centered on pattern matching in a social network. Due to the vague and imprecise description of the pattern, a plainly automatic approach is difficult, but in combination with human guidance and rule probabilities it can be solved rather quickly. It is thus a model example of the visual analytics methodology.

This section is based on work also presented in:

- H. Bosch, J. Heinrich, C. Müller, B. Höferlin, G. Reina, M. Höferlin, M. Wörner, and S. Koch. Innovative filtering techniques and customized analytic tools. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST '09)*, pages 269–270, 2009

3.6.1 VAST Challenge 2009 Scenario

In the scenario of the VAST Challenge 2009, an embassy has an issue with confidential data leaking to outsiders. Due to the employed security measures, an employee of the embassy is probably involved in this data theft. The challenge is divided into three 'Mini Challenges' focusing on different data types, i.e. badge and computer network traffic, a social network, and video data. Our entry to the challenge [Bosch et al., 2009] contributed to each mini challenge, but this section is focusing on the second mini challenge to which the selection management approach was contributed in collaboration with my colleagues Michael Wörner [2014, chap. 2] and Steffen Koch. In the second mini challenge, the embassy suspects that one of their employees made contact to a criminal network over the fictional social network 'Flitter'. The usernames of the employees are unknown and we therefore have no entry point for matching a network pattern. The network has 6000 members and some details about the structure of the criminal network are suspected and described informally. The associated network is believed to take one of two forms:

> *A.* The employee has about 40 Flitter contacts. Three of these contacts are his "handlers", people in the criminal organization assigned to obtain his cooperation. Each of the handlers probably has between 30 and 40 Flitter contacts and share a common middle man in the organization, who we have code-named Boris. Boris maintains contact with the handlers, but does not allow them to communicate among themselves using Flitter. Boris communicates with one or two others in the organization and no one else. One of these contacts is his likely boss, who we've code-named Fearless Leader. Fearless Leader probably has a broad Flitter network (well over 100 links), including international contacts.
> *B.* The employee has about 40 Flitter contacts. Three of these contacts are his "handlers", people in the organization assigned to obtain his cooperation. Each of the handlers likely has between 30 to 40 Flitter contacts, and each probably has his or her own middle man in the organization, who we've code-named Boris, Morris and Horace. It is probable the middle men will not allow the handlers to communicate among themselves using Flitter. Each of the middle men probably communicate with one or two others in the organization, and no one else. One of the contacts for all of the middle men is the head of the organization, Fearless Leader. Fearless Leader has a broad Flitter network (well over 100 links) including international contacts. [Grinstein et al., 2009]

These descriptions are imprecise by, first, the number of contacts that a role has ('30 to 40') and, second, if a statement is valid at all ('Fearless Leader *probably* has a broad Flitter network'). These imprecisions need to be modeled accordingly.

3.6.2 Domain Model

The informal description of the roles' properties where formalized into a set of fuzzy rules of the form '$\langle role_1 \rangle$ knows [at least] $\langle range \rangle$ $\langle role_2 \rangle$'. Here, a $\langle role \rangle$ can be any of the mentioned roles `employee`, `handler`, `middle man`, `leader` or `contact`. The latter role is a generic description for any user of the Flitter network. The range is modeled as a target zone which qualifies to make the statement valid (e.g. Handlers have 30-40 Flitter contacts) and a gradual fall-off range in which the statement is no longer valid but also not completely off (e.g. 20% below or above the target range). In total 23 rules where created

(see right hand list in Figure 3.21), which can cover most statements of the scenario description, but not all. For instance, we can say that each handler needs to know one middle man, but not if this middle man should be the same for each handler. However, this can be circumvented by introducing a statement that the leader may know only one middle man.

The initial assumption is that every member of the network can take any role. They are therefore a *candidate* for these roles. By applying the rule that an employee needs to know 30-40 other users of the network, everyone having a substantially different number of contacts is excluded from the set of employee candidates. Accounts with a number within the fall-off range, have a decreased certainty of being this role, and accounts with the correct number of contacts maintain a certainty of 1. When additional rules are applied these certainties can be further decreased and everyone below the threshold of 0,5 is excluded as a candidate for the specific role.

Starting with many candidates for each role makes working with exact ranges or maximum values infeasible. We do not yet know who the one correct leader is and cannot exclude every middle man that knows more than one potential leader. Therefore, most rules are modeled as 'knows at least' and only penalize accounts with less than a certain number of contacts. This also means that the combined result of multiple rules is depending on the order in which they were applied. In order to explore the potential rule configurations we integrated the filter/flow approach in our solution.

3.6.3 Exploring Possible Solutions

The exploration of the configuration space and matching networks is done in two views: the *Hypothesis Graph View* and the *Network View*. The first one allows the application of the rules to the role candidate sets to subsequently reduce the potential candidates. The graph structure can be designed freely to explore different combinations and sequences of rules and provides an aggregated overview of the analysis progress. The second view provides a detailed network visualization of all accounts that are still a candidate to at least one role to allow the analyst to inspect the results of rule configurations.

In the hypothesis view, each node shows four candidate sets, one for each role, as colored bars. The different roles are color-coded consistently between the views with purple for `leader`, cyan for `middle man`, green for `handler`, and red for `employee`. Each bar depicts the current distribution of role probabilities as a gradient. Fully colored bars indicate large candidate sets with a high role probability for every account, gradual gradients indicate

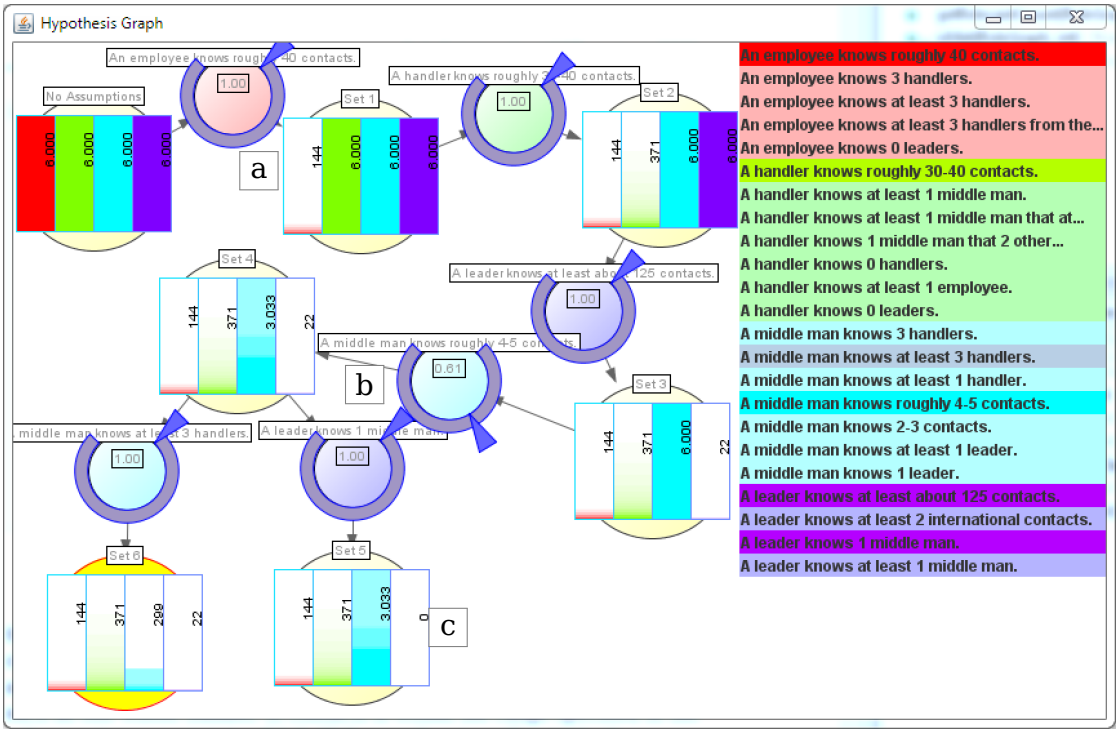


Figure 3.21 — The *Hypothesis Graph View* containing the available rules on the right hand side and the freely constructible sequence of rule applications on the left hand side. The colored bars at each node show the probability distribution for each role at the respective rule sequence.

many different role probabilities throughout the candidate set, and an almost empty bar with a sharp transition between colored and white area indicates few but highly possible candidates. The latter one is the desired outcome for each bar at the end of the analysis.

The available rules for reducing the candidate sets are available at the right hand side of the view. They can be applied to existing nodes by dragging and dropping the rule's name near the target node. This creates a node for the rule, on which the confidence for this rule can be adjusted by a slider widget, and a further node for the new resulting role candidate sets, showing an immediate feedback on the selectivity of the applied rule by comparing the colored bars between the source and result node. In Figure 3.21, a rule application with full confidence can be seen at location (a), where it reduces the candidate set of the employee role from 6000 to 144 candidates. At location (b), a role with reduced confidence was applied reducing the

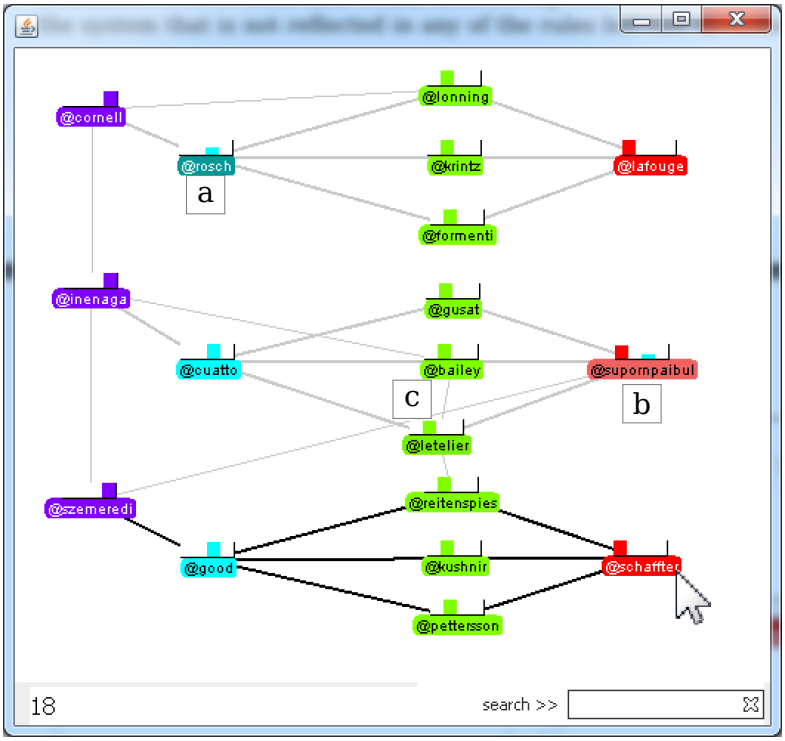


Figure 3.22 — The *Network View* showing the last three matching networks for manual review. From left to right: the roles of `leader`, `middle man`, `handler`, `employee`. Accounts that are a candidate for multiple roles may appear multiple times in the view.

candidates for the middle man role from 6000 to 3033 candidates with a very soft gradient. If the analyst arrives at a point where neither scenario appears to be valid (see zero leader candidates in Figure 3.21c), the analysis can be continued at any other node of the graph by simply creating a new branch and potentially using the 'problematic' rule at a later point when the other candidate sets are already reduced further. To indicate which rule was already used in the graph, they are drawn with increased saturation in the list of available rules. By selecting a result node, the detailed *Network View* of this rule configuration is opened in a new window.

Figure 3.22 shows an advanced state of the analysis, in which three possible structures where already identified by concatenating fuzzy rules. Each account of the network that is part of at least one candidate set is shown as a node in the network view. They are horizontally ordered by their roles and connected to all of their contacts that are also part of the view. If an

account is a member of multiple candidate sets, it is shown multiple times in the network. The nodes are accompanied with the same colored bars as in the graph view, this time depicting the account's probability for each role. Additionally, the node's color is a combination of the role colors weighted according to the remaining probability for the role. This can be seen by the slightly different shades of `@rosch` and `@supornpaibul` at the locations (a) and (b) in Figure 3.22. A candidate for every role would be colored white, and a handler/employee has a combination of green and red, i.e. yellow. Moving the mouse over an account node highlights the potential structures between different roles. Here, the analyst can see that in the middle network the two handlers `@bailey` and `@letelier` are contacts to each other (Figure 3.22c), which is not allowed due to the scenario description.

By invoking a context menu on either node, the analyst can manually exclude this account from a certain role candidate set, which causes a re-evaluation of the rule configuration of the hypothesis graph and renders the whole middle structure invalid. It therefore disappears from the network view after this interaction. This is a good example of the visual analytics feedback cycle, in which human guidance through interactive visualization and automatic approaches enrich each other's capabilities. From the two remaining networks, the upper one has a rather low confidence for the middle man candidate and is only included due to the low confidence that we assigned to the rule in Figure 3.21b. Therefore, the highlighted network is the most probable candidate for Scenario A, and was indeed the correct answer from the VAST Challenge solution.

Chapter 4

Automatic Adaption through Ontology Exploitation

Today's World Wide Web[1] is more than the network of linked hypertext documents for which it was postulated in its early days. It is rather an open data-processing system in which each 'web page' can provide, convert, remix and interactively present information in a multitude of ways. Using their online presence, government organizations publish census and administrative information in structured formats, following Open Data initiatives. Interested citizens can use web-based tools to aggregate, visualize, and relate this data in order to draw insights and produce new information products to be distributed on the web. Enthusiasts help to maintain databases and enhance encyclopedic articles with categorical information. Companies providing content and interaction platforms make their services and data publicly available in order to spur the networking of services and therefore increase their reach.

> [...] *Data Web* refers to the evolution of a mainly document-centric Web toward a more data-oriented Web. In its narrow sense, the term describes pragmatic approaches of the Semantic Web, such as RDF and Linked Data. In a broader sense, it also includes less formal data structures, such as microformats, microdata, tagging, and folksonomies. [`http://www.visualdataweb.org/`]

[1] Here, stipulatively defined by its transfer protocol `HTTP(S)`.

This definition of the term Data Web, nicely emphasizes the data-centric notion while at the same time being less strict on the employed data and presentation formats, compared to Semantic Web or Linked Open Data. It therefore covers a greater portion of the mechanisms in use, e.g. the artifacts of Web 2.0 where participating users produce not only documents but also additional metadata such as tagging folksonomies and product ratings. The web is a collection of data sources and tools to address various tasks. While some are intended for the general audience, others require the user to have a specific domain knowledge. For instance content aggregators and meta-reviewing platforms such as Idealo[2] use the data of many shopping and product review platforms to offer a faceted browsing of the collections of multiple stores based on the products' attributes. Even if common web users could formulate a query for the intellectual property rights search engine Espacenet,[3] they are likely to fail at making sense of the result containing links to classification schemata and legal event codes.

Interacting with the elements of the Data Web is not as straight forward as reading a document. Heim [2012] argues that the key for providing access to the data for the average user is the interactive alignment using visual interfaces. This is reflected by the three key design principles highlighted by the *Visual Data Web*:

> *Minimal Technical Requirements* – [. . .] Ideally, the applications are immediately executable in the Web browser without the need for a local installation.
>
> *Intuitive User Interfaces* – [. . .] Deep knowledge about the underlying technologies should not be necessary for working with the tools.
>
> *Standardized Data Access* – Data access should be as generic as possible and largely based on common Web standards, such as SPARQL and XML. [http://www.visualdataweb.org/]

However, while these design principles provide access to the data, making sense of it requires combining different data sources, analyzing the relations, drawing conclusions, and evaluating results. This is a demanding tasks in itself and additionally requires knowledge from the data's domain.

The EC-project Personalized Environmental Service Configuration and Delivery Orchestration (PESCaDO) has shown that the user's decision process

[2] http://www.idealo.de/

[3] http://worldwide.espacenet.com/

can be supported by general-purpose reasoning [Shearer et al., 2008] using web data sources. This was enabled by the ongoing improvement of Semantic Web technology [Moßgraber and Rospocher, 2012; Rospocher and Serafini, 2012; Motik et al., 2009] and the modeling of the application domains in semantic structures. At the same time the semantic information can be used to personalize all steps of the process from query generation, over data acquisition and result computation. The PESCaDO project [Wanner et al., 2010, 2011, 2012a,b] provides a web-based personalized decision support related to environmental data, a domain which features complex interrelations and has a broad data coverage in the World Wide Web. Its target audience are citizens, public services, and administration in sectors sensitive to the environmental condition, e.g. people planning activities that are exposed to chemical and meteorological weather.

This chapter focuses on the personalization and adaption of the PESCaDO user interface. After an introduction to the PESCaDO system, the remainder of the chapter describes two aspects in detail: Section 4.2 presents a technique to exploit context information and the encoded domain knowledge to improve the process of formulating a request. By automatically deriving explicit relations between domain entities, the input form can be enhanced with better error highlighting and explanation. Section 4.3 explains the personalized orchestration of visual result representation and potential trivialization of domain data. By introducing a generic web-based visualization framework for spatial data, different environmental data sources can be displayed on a map concurrently. The framework shall convey the semantically derived findings in a way suitable to the user profile, being either a domain expert or casual web user.

Parts of this chapter have been published in:

- H. Bosch, D. Thom, and T. Ertl. Das Web als personalisierte Entscheidungsplatform – Die PESCaDO Idee. In *Lecture Notes in Informatics (LNI) – Proc. Informatik 2011: Informatik schafft Communities*, volume P-192, page 256, 2011
- H. Bosch, D. Thom, G.-A. Heinze, S. Wokusch, and T. Ertl. Dynamic ontology supported user interface for personalized decision support. In *Proc. 5th Int'l Conf. Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2012)*, pages 101–107. IARIA, 2012

- S. Vrochidis, H. Bosch, A. Moumtzidou, F. Heimerl, T. Ertl, and Y. Kompatsiaris. An environmental search engine based on interactive visual classification. In *Proc. 1st ACM Int'l WS Multimedia Analysis for Ecological Data (MAED '12)*, pages 49–52. ACM New York, 2012
- L. Wanner, H. Bosch, S. Vrochidis, N. Bouayad-Agha, G. Casamayor, L. Johansson, A. Karppinen, A. Moumtzidou, I. Kompatsiaris, and T. Ertl. Involving the expert in the delivery of environmental information from the web. In B. Page, A. G. Fleischer, J. Göbel, and V. Wohlgemuth, editors, *EnviroInfo*, Berichte aus der Umweltinformatik, pages 561–568. Shaker, 2013

4.1 Environmental Decision Support

Environmental information is one of the most ubiquitous contents on popular websites. Almost every news site and web search portal contains weather forecasts, several national or regional meteorological institutions provide public access to their knowledge, and data from privately-owned weather stations are voluntarily shared and enhanced with forecast models by services like the Weather Underground.[4] With this broad availability, citizens are increasingly aware of the influence of environmental factors on their personal decisions regarding their health and quality of life. However there are problems that hinder the use of the available environmental data for personal decision support:

- *Data Quality* – For the same spatiotemporal extent and the same environmental factor, different providers state different forecasts[5]
- *Data Resolution* – Different providers use different resolutions for publishing their data, which can become problematic when comparing results for rural areas with built-up urban areas.
- *Domain complexity* – The relationships within the environmental domain are quite complex. For instance, precipitation can cause dangerous road conditions on several ways such as black ice. The effect of chemical weather to the personal health is equally uncertain to the average user.

[4] `http://www.wunderground.com`

[5] For instance, comparing the next-day forecasts for wind speed found by a query for'Weather Barcelona' on December, 3rd 2013: first hit 16km/h, second hit 43km/h.

The influence of ozone concentration is related to the physical intensity of the current activity of the person, etc.

- *Individual Situation* – The users may suffer from several health issues that cause individual sensitivities against certain pollen types, air pollutants, or weather situations. These may range from allergies over asthmatic conditions to cardiovascular diseases. For these users, the average thresholds for computing air quality indices may not be appropriate.

Therefore the users of environmental information need support in performing three tasks in order to make an informed decision:

1. Identify the environmental factors that relate to their personal situation.

2. Identify available and trustworthy providers for the needed environmental information and combine their data.

3. Interpret the reported measurements or forecasts for their personal situation.

PESCaDO uses public websites and data sources to offer personalized environmental decision support. Based on the user's profile and query, it infers the relevant environmental factors, aggregates the needed data, draws conclusions by applying the encoded domain knowledge, and presents the result textually as well as graphically. For this purpose, it relies on the evaluation of semantic models to adapt to different problems and situations and not just creating a web mash-up of preselected data providers. The underlying ontology of PESCaDO acts as an abstraction layer between the user's problem definition and the available data and hides the domain inherent complexity from the application and the user. This allows the generation of queries independently of the currently available data sources, while it is still possible to connect to them during the execution of the request. PESCaDO therefore has two main activity phases, data source acquisition and online user sessions, that share a domain ontology composed of knowledge about environmental information, health situations, user activities and profiles, and additional relations linking these areas. In the following, we introduce the PESCaDO architecture by explaining the two activity phases as well as the parts of the ontology that are most relevant to the user interface.

4.1.1 Environmental Search Engine for Data Source Acquisition

During the data source acquisition phase, we try to find potential data sources for environmental information of specific locations and identify the way to access the data. These data sources can be websites with textual information, images of weather maps or graphs, direct links to data encoded in formats like XML or JSON, or Web-APIs. They are summarize under the generic name 'environmental nodes'. Findings are stored with their access parameters in a repository and are linked to the related concepts of the PESCaDO ontology and the geographic location, e.g. wind speed for the city of Helsinki. The discovery of new nodes is supported by a semi-automatic, domain-specific search engine, which will be described in the following. In this case it is configured for the environmental domain but could service others as well. Details on the content extraction can be found in the works of Pianta and Tonelli [2010] and Moumtzidou et al. [2013, 2014].

Domain-specific search engines, also called vertical search engines, are either processing search results of existing multipurpose search engines, or directly crawl the web to index the relevant sites. Our approach is of the former category and combines keyword spicing [Oyama et al., 2001] with interactive result classification [Heimerl et al., 2012]. First a query is constructed and targeted on the environmental domain by adding 'spice' keywords. Because keyword spicing alone produces also false hits, the results are subsequently classified into relevant (containing environmental information) and irrelevant (otherwise) nodes by a SVM model. Related to this approach is the work of Kun Wu and Zhang [2010] who employ visual and textual features to improve the results of image search engines, and Luong et al. [2009] who propose a classification-enhanced focused crawler to find resources for automatic ontology construction. We could show that an interactive training of the employed classifier can significantly improve the performance of the resulting environmental search engine when dealing with live web data [Vrochidis et al., 2012].

Figure 4.1 shows the process of the semi-automatic environmental node discovery. The first step is the selection of geographical location names and environmental factors for which the new nodes shall provide data. The location selection is supported by a map that translates a spatial selection into a list of toponyms[6] that can be used for the text-based web search engines. The second input is one of the environmental factors of the PESCaDO ontology which includes keywords related to the chosen factor into the query. Finally,

[6] This feature uses the GeoNames service for reverse geocoding: `http://www.geonames.org`

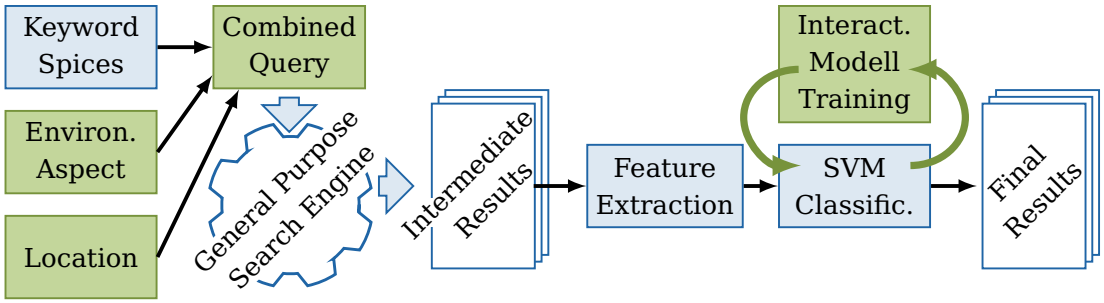


Figure 4.1 — Schema of PESCaDO's environmental data source acquisition. A textual web search query is created from a term describing an environmental aspect, a location name, and spice keywords that are frequently appearing on data sources. The intermediate result list cannot be used directly as it contains too many unrelated sites and is revised using a classification approach. The classifier can be re-trained interactively on the current results when the need arises to improve its performance. Elements denoted in green involve a human actor.

additional spice keywords are included that help in targeting the search to environmental data websites, e.g. 'forecast' or units of measurement. These keywords are maintained manually but can be automatically benchmarked to guide their selection by monitoring the amount of false positives that they produce. The final query string is send to a general purpose web search engine[7] and a preset amount of hits is retrieved for classification.

In order to apply classification, the SVM requires training data and appropriate features for website data. The training data was initially composed by environmental experts that listed one hundred websites as positive examples and added two hundred negative examples comprising random websites and environmental discussion websites without actual data. The imbalance in the training set reflects the broader variety of unrelated websites compared to the related ones. The features are consisting of key-concept terms in a bag-of-word model and are generated by the KX framework [Pianta and Tonelli, 2010], which is also employed during the actual content extraction. Each website of the result list is therefore accessed and presented to KX. Additionally, a thumbnail is generated to help with the visual, interactive evaluation of the classification results.

The interactive classification framework is based on the same principles as

[7] Yahoo! BOSS Search API: `http://developer.yahoo.com/boss/search/`

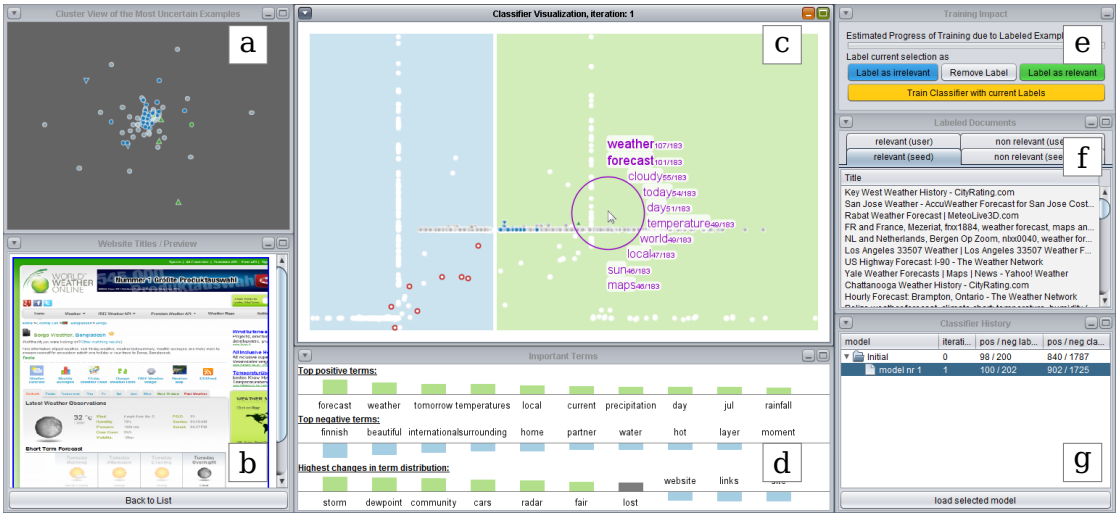


Figure 4.2 — The interactive classifier training desktop for environmental nodes consisting of: (a) a clustering of the most uncertain classification results, (b) thumbnail preview of the currently selected site(s), (c) the visualization of the websites in relation to the hyperplane, (d) important key terms and how the classifier perceives them, (e) training controls for labeling and proceeding to the next visualization/iteration, (f) history of assigned labels, (g) history of classifier configurations for undo operations.

the interactive training for the classification-based filter of *ScatterBlogs2*, described in Section 3.5.2 but is more similar to the version presented in Heimerl et al. [2012]. Because of the different vector model using only key-terms instead of the whole textual content, a radial kernel is applied to allow for a better separability of the websites when using SVM classification.

Its interface is depicted in Figure 4.2. Instead of a map, the central component shows a projection of the websites' feature vectors in relation to a straight vertical white gap as a visual abstraction of the SVM hyperplane, i.e. the decision boundary. It separates the data classified as relevant on the right hand side from the data classified as irrelevant on the left hand side. Each website belonging to the training data or web search results is depicted as a dot in the appropriate region defined by the classification result. Dots in white represent labeled training sites, while dots in gray are the unlabeled sites. The currently selected sites are marked with a darker outline. Each dot's distance to the decision boundary corresponds to the classifier's confidence value for this site, i.e. the most uncertain classification results are located close to the boundary. The vertical layout of the view is based on a principle

component analysis of the training data and attempts to place similar sites in close proximity to each other. If an already labeled site, which is not part of the support vectors, is misclassified by the current classifier, it is marked by a red outline. This can happen if the data is not linearly separable after the kernel transformation and indicates to the users that the classifier's kernel is not parametrized optimally.

In the top left corner, a clustering of the hundred most uncertain classifications results is shown in a layout that considers both dimensions to map the similarity and ignores the classifiers confidence. Labels that are assigned to websites of this set have a high potential to influence the classifier's support vectors and are therefore given more prominence by this second view on the data. The clustering is computed by a bisecting k-means algorithm [Steinbach et al., 2000] and a subsequent projection into 2D using the LSP algorithm [Paulovich et al., 2008]. Because the decision border is not displayed here, the classes of the sites has to be indicated by their color. Especially heterogeneously classified clusters identify suitable regions for detailed inspection since the chance that some of them are classified incorrectly is high.

The thumbnails of currently selected websites are shown in the lower left corner in order to allow the user to quickly judge the relevance of the sites. If the thumbnail is not sufficient, a click opens the default browser and loads the website for a detailed inspection. The bar chart view next to the thumbnail explains how the classifier 'perceives' the websites data by showing the most distinguishing features of the two classes. For this purpose, the features of the support vectors of both classes are summed separately and the difference between them is calculated. The first and second row of bars displays the ten most defining features for the relevant and irrelevant class respectively. The third row shows those features whose scores changed the most compared to the previous training iteration. Each of the bars can be hovered by the mouse which highlights all websites containing the respective feature.

The interactive training system can be used to review the classification of new results in an aggregated visualization and to update the classifier to maintain the performance of the domain specific search engine.

4.1.2 Process of the Online Decision Support Interaction

While large parts of the system work automatically in the background during an end user's online session, most of them are influenced by the users' profiles, interactions, and feedback. The interactive classification training described in the last section is one example for this human influence. While this activity

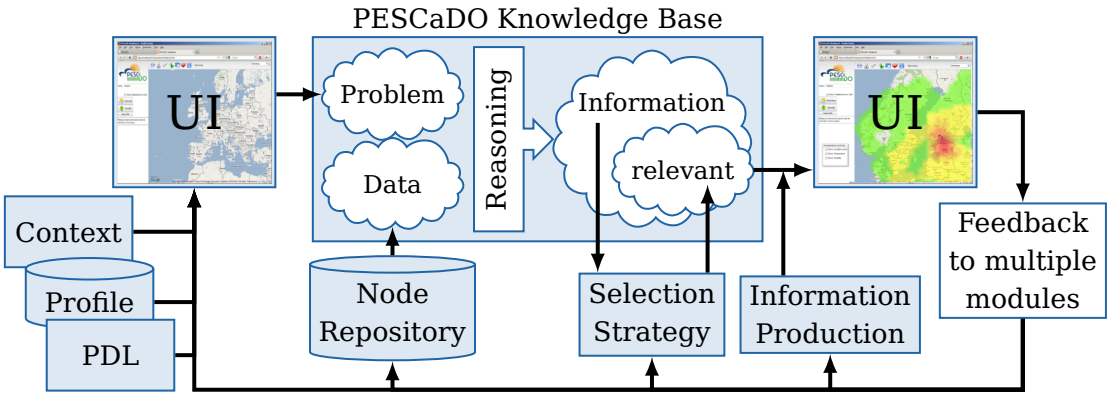


Figure 4.3 — Schematic process of a PESCaDO online user session. The users formulate their request supported by a dialog that draws knowledge from the context, user profile, and the ontology. The problem definition is instantiated in a knowledge base together with the PESCaDO ontology and environmental data. A general purpose reasoner infers new information from which a selection strategy marks the most relevant parts. These are transformed into the graphical and textual answer of the system on which the user can supply feedback.

is performed separately and from domain experts, others relate the end users' interaction with the system during a decision support session. The involved components and the user's influence can be described using the decision support loop shown in Figure 4.3.

Based on her information need, the user formulates a query for the system using an intelligently steered dialog which supports this activity by incorporating the current session context, domain ontology, and her personal profile. This part is detailed in Section 4.2. In order to associate the input with the stored domain knowledge and the environmental data, it is translated into the semantic structures of the PESCaDO ontology. The semantic structures are then aligned with the available environmental information in a knowledge base containing the PESCaDO ontology. This allows a generic Semantic Web reasoner to derive possible implications related to the user's problem. The resulting knowledge is filtered according to a selection strategy that eliminates rudimentary data facts which were created during the reasoning step and would obscure the result presentation. The last step of one iteration of the process is the graphical as well as textual presentation of the resulting insights as an orchestrated and configurable ensemble of visualizations tailored to the

present user and available data. This part is detailed in Section 4.3. Finally, the resulting view can be used as the basis for adjustments and user feedback in order to influence the individual steps of the process if the demands should not be satisfied. This can be done by:

- directly changing the presentation of the current results (information production),
- rating elements of the results (selection strategy),
- exchanging data sources (node repository),
- changing the personal profile (profile),
- or reformulating the query if the problem was misinterpreted.

4.1.3 The PESCaDO Ontology

The PESCaDO ontology is the central element of the system and all other parts connect to its contents. Therefore, it is quite complex and consists of ten modules that are interconnected [Rospocher, 2010, 2014]. Each module covers an aspect of the system:

- *Problem Description Language (PDL)* – classes to model the user's query and profile and link them to environmental aspects, .
- *Diseases* – Database of diseases related to environmental factors structured according to WHO's ICD-10 classification.[8]
- *Geographical Data* – classes to model geographical areas, points, and toponyms.
- *Environmental Data* – classes to model environmental information such as forecasts, measurements, aggregation types, and qualitative ratings.
- *Units of Measurement* – as the name implies and mostly taken from the Sweet Ontology [Raskin and Pan, 2003].
- *Environmental Nodes* – classes to represent data sources by their environmental type, access parameters, data quality, etc.
- *Exceedance Data* – classes to represent the various threshold exceedance types such as one-time exceedance or multiple exceedances over a set time frame.
- *Conclusions* – the warnings and suggestions that the system is able to infer from a problem description and present data.

[8] [World Health Organization, 2012]

- *Text Synthesis Data* – classes used during the generation of the textual result representation.
- *Logico-Semantic Relations* – types to model higher level relations between the content elements of the other modules.

Regarding the user interface, the most relevant modules are the Problem Description Language (PDL) for the query formulation support as well as the environmental data and units of measurement for the visual result presentation. The former is the structure in which every user input has to be translated in order to influence the system's back-end behavior. The PDL contains five main top-level classes for describing a request: `Problem`, `Request`, `Activity`, `User`, and `Task`. The `Problem` class is used as a wrapper to collect all semantic statements that belong to one user request. The `Task` class can be used to schedule the re-execution of known `Problem` descriptions in frequent intervals to allow for a push-notification service.

The other top-level classes are not instantiated directly but have subclasses that build a taxonomy for each concept. Requests shall describe the information need that the user wants to fulfill with the query. This can range from requesting a suggestion for an administrative action (e.g. road maintenance), over reports (e.g. quality-of-live reports), to inquiring warnings (e.g. about potential health issues). Activities are the context in which the request should be answered, covering statements such as attending an outdoor event, activities of varying physical intensity, long term stays, or traveling. The user class divides into different user types and user profiles containing information about their health issues and demographic details. Any of these taxonomies can have further relations to other concepts of the PESCaDO ontology in order to link to additional context information, e.g. a second geographic location for a report, or related environmental information that is needed to answer a request. They are also interrelated and cannot be mixed arbitrarily. Certain user types cannot use certain request types, and not all activities may be used for all requests. All these restrictions and requirements are encoded in the PDL's structure.

4.2 User Input Validation and Support

PESCaDO is an example for a domain-specific software solution with requirements on the user supplied requests to match the special back-end infrastructures. In this case the back-end requirements stem from the em-

ployed domain ontology to which the formulated queries must fit. Otherwise, the user input can lead to an inconsistent ontology, i.e. containing contradicting statements, or miss important information needed to trigger the encoded rules. The complexity is increased by the fact that multiple user types have different information needs leading to various usage scenarios that need to be covered by the user interface.

Nevertheless, in every closed system, potential inconsistencies and required user input can be anticipated to support the users during their query formulation to avoid submitting invalid requests. This is already done for interactive configuration management for mass customization or automated generation of user interfaces. However, even with an ontology as the basis for computation, the relation between the input elements of the user interface may be stated implicitly and cannot easily be inferred due to potentially missing relations. For example, the fact that the user may choose one of multiple options in the user interface can be modeled in various ways in the ontology: the options may be siblings of the same super-class by using OWL constructs, linked by an arbitrarily typed relations to a collecting class representing the choice, or be part of a restriction of an implicit class definition.

The following sections describe how an intelligent user interface can be achieved by accounting for already submitted information, user profiles, and an automatic analysis of ontological relations of classes related to input elements. In this context *intelligent user interfaces* are defined as systems that react individually on user input based on background information. This definition is in line with *intelligent support systems* discussed in the work of Delisle and Moulin [2002] but does not have to be based on machine learning algorithms. They can be grouped into three, not necessarily disjoint, categories inspired by Dryer [1997].

Guides support users by providing additional information for their tasks. This guidance may range from simple hints about the expected date format to full user manuals helping in choosing the correct form. If guides dynamically take into account the available information, they can also be considered 'intelligent'. For instance, the *COACH* system [Selker, 1994] builds an adaptive user model and provides contextual help by commenting on the user's actions. Guides are useful for providing local and not too complex information about the currently focused aspect of the user interaction.

Wizards support users by structuring complex input forms into separate, sequential, and thematically coherent pages. Each page can take the previously provided information into account, e.g., to include or exclude branches of the predefined course of the dialog. In WOLD [Stocq and Vanderdonckt,

2004], a wizard for generating new user interfaces suggests parameter values. Wizards profit from the fact that only few interactive elements are available at any point in time. However, with increasing task complexity it becomes harder to define thematically coherent but independent subtasks to be grouped in a linear structure.

Reactive Systems can actively influence the current dialog. This is often implemented through the use of *agents* [Wooldridge, 2002] and covers a wide area of applications from saving previous input for automatic input completion, to learning the user's behavior to better adapt dialogs to their needs. Examples for this support style can be found in the work of Yang [2009] describing a website that uses expert user input to adapt search results to the requesting user, as well as the system of Lee et al. [2009] which supports tourists in planning a path through cities to meet different sightseeing interests. In *mixed-initiative* systems [Armentano et al., 2006], e.g. in the work of Frank et al. [2001] for trip planning, the active part changes between user and machine in a predefined sequence. While this can also be achieved by using agents, the clearer role definition leads to less user astonishment.

The intelligent user support for personalized query formulation in PESCaDO is a combination of these principles in the form of a guided wizard that reacts intelligently on the user input using an automatically derived rule set. For this purpose, simple logic rules are created based on templates that match certain ontological relations. The new explicit relations can be used for (*a*) effortless fully automatic input validation and error explanation, and (*b*) dividing the input options into coherent subsets for better structuring a wizard dialog where the former pages are mostly independent from inputs on later pages.

4.2.1 Rule Generation

Because each option in the user interface needs to be translated into the PDL structures, an initial mapping of input elements to ontological entities can be obtained from this step. The mapping is stored directly in the web front-end as a property attached to the input elements containing the resource identifier of respective ontological concept. For each pair of these concepts their ontological relation is examined and matched to a set of template relations that encode if the inputs require or exclude each other. Some prototypical templates and their resulting logical rules are listed in Table 4.1.

The first template handles subclass relations. In PESCaDO, `Hiking` is a subclass of the concept `Outdoor Activity`. If users state to undertake an

outdoor activity, they also have to state one and only one of its subclasses. Therefore, a rule is created for each subclass Y such that if the parent class X is selected and none of its siblings $Z_{i..n}$ (subclasses of X) is selected, the subclass Y is a required input. Because this rule exists for each of the siblings, too, they all are required unless one is selected.

The second template matches a class restriction using an object property. Creating a request for warning of potential health issues (X in this case) requires that the users state one of a defined set of a activities ($Z_{1..n}$) while omitting others. This is reflected by first inferring the least common ancestor of the activities, in this case the class `activity` itself and marking it as required. Then, the same schema as above is used to make all Z_i required inputs as long as none of them is selected by the user. The third rule is the complement of the second and states that selecting any of these classes would create an invalid request if X is selected.

The forth template is used when needing exactly one input option, typically referring to the user type. It is translated that if the statement X is selected, Y is a required input and all siblings S_i of Y would create a conflict. This is useful for generating a meaningful explanation why an option cannot currently be selected.

The fifth and sixth template handle cardinality. Here, we introduce a `count` operator to state how this rule should be evaluated in the interface.

Similar templates following the same approach exist for disjoint classes,

Ontology Relation	Resulting Rules
Y subClassOf X	$X \wedge \neg(S_1 \vee ... \vee S_n) \rightarrow Y$
X someValuesFrom $\{Z_{1..n}\}$	$X \rightarrow L$
$\hookrightarrow$ and for each Z_i:	$L \wedge \forall j(j \neq i \Rightarrow \neg Z_j) \rightarrow Z_i$
not(X someValuesFrom $\{Z_{1..n}\}$)	$X \rightarrow \neg Z_i$
X allValuesFrom(Y)	$X \rightarrow Y$
$\hookrightarrow$ and for each $S_i \neq Y$:	$X \rightarrow \neg S_i$
X exactCardinality(1, Y)	$X \rightarrow Y$
X minCardinality(a, Y)	$X \wedge count(Y) < a \rightarrow count(Y) \geq a$

Table 4.1 — Some prototypical rules inferred from ontological relations. Here, S_i is a siblings of Y and L is the least common ancestor of all Z_i. The *count* operator is used to count the occurrences of Y. The ontological relations used in the table are based on OWL structures but are paraphrased to keep the table contents concise

special cases of cardinality, unlisted siblings of `someValuesFrom` relations, etc. Some input options are modeled in the ontology as datatype properties that map to literals instead of concepts, such as `hasStartDateTime`. These are examined in the same fashion to create further rules for date ranges and route definitions.

4.2.2 Intelligent Wizard and Error Explanation

The PESCaDO user interface is a website composed of a Google Maps[9] component and a dialog floating over it (See Figure 4.4). The dialog can be collapsed to a small icon to reveal a larger part of the map which is used to visualize environmental data and perform geospatial selections, such as routing points for travel or areas of interest for weekend activities. The top part of the dialog has four tab buttons (`Home`, `Request`, `Result`, and `Profile`) that exchange the lower part. The `Home` tab is the welcome page and can show currently important information when arriving at the PESCaDO website. The `Request` tab contains several pages that comprise the guided wizard that is detailed below. The `Result` tab holds the textual result of the last submitted request and potentially widgets to control the environmental data visualization. The `Profile`) tab is for user log-in and profile editing.

Of the roughly 640 concepts in the PESCaDO ontology, 26 were relevant for the User Interface (UI) because they have a mapping to input elements. From their relations and properties, 56 rules were generated and stored in an XML file. The rules format allows for an easy evaluation in the user interface by substituting the class names for a Boolean value depending on whether the related variables are filled by the user. If the proposition of the rule is invalid in the current interpretation of the input, a conflict is present. This can be due to either missing a required or containing an excluded input option.

Based on the now explicit relations between input elements, they can be grouped on pages according to their influence. If the most influential input fields are placed at the beginning of the interaction, most of the fields of latter pages of the wizard should already be marked as required or excluded by the time the users arrive there. Pages containing only excluded fields could be skipped while the user processes through the wizard, but are deliberately shown in order to show inexperienced users potentially available fields. This provides them with the opportunity to fill in the fields nevertheless, thus creating an error highlight, and go back to change the input fields that

[9] `https://developers.google.com/maps/documentation/javascript`

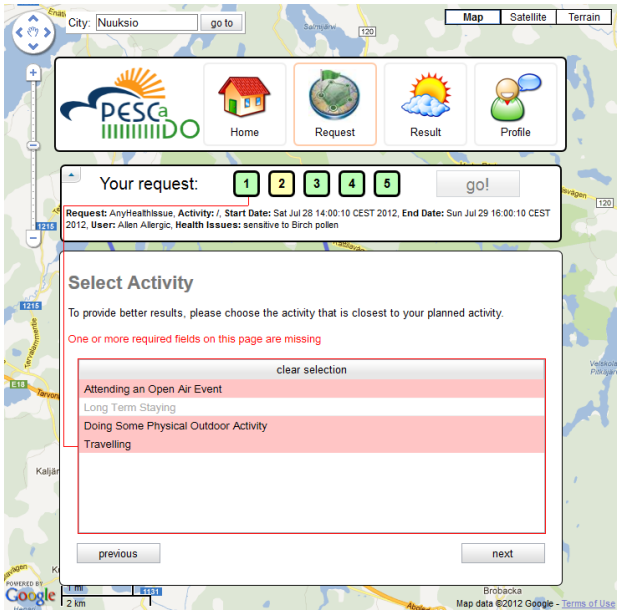


Figure 4.4 — The PESCaDO query generation wizard. The top row of buttons access different parts of the dialog. On the *request* page, four of the wizard's pages have been filled in successfully (green numbers). The current page (yellow number) has an error because an input on the first page (red line) requires the user to fill in an activity. A summary of the user-supplied information is given immediately below the page numbers. In the background, a Google Map environment is available for region selection and result visualizations.

are marked as causing the conflict. The overall progress can be seen by a numbered list of pages in the upper area of the dialog. Here, successfully submitted (i.e. conflict-free) pages are marked with a green color, the current page is marked with yellow, and pages with errors are marked with red (see Figure 4.4). The user can navigate freely to any page of the wizard by clicking on a page number.

Inconsistencies in the input are pointed out to the user in three ways at the same time, by highlighting, linking, and descriptive texts. Every input field's widget implements its own highlighting methods. This way, even non-standard input methods like a route selection can be highlighted in an appropriate way. The highlighting of forbidden fields or unfilled but required fields follows the common convention to change the color of the fields' borders and backgrounds to gray or red respectively, and show an error message.

The error message contains an explanatory text of the rule's meaning that was created during the rule generation phase and includes the involved input names and their relation. Additionally, a red line connects conflicting inputs in order to facilitate the tracing of errors. This cue-based view coordination is similar to the context-preserving visual links of Steinberger et al. [2011]. If one of the conflicting widgets is situated on a different page of the wizard, the red line leads to the appropriate page number at the top of the dialog. The rules are evaluated if the mouse cursor hovers over a navigation control to go to another page of the wizard. This way, the user is not irritated by a changed appearance of the form after every editing step but can correct errors on the current page before proceeding to the next page. Additionally, the rules of the currently edited input element are evaluated if it currently is in a 'conflicting' state to immediately see when an error is resolved.

4.3 System Output Personalization

The response of the system is composed of the environmental data that was deemed relevant to the problem, a natural language description of the situation, as well as potential guidelines that refer to the planned activity and the user's profile. This result is personalized in several ways over the course of the generation process. Here, the term 'personalization' is used in the sense of adjusting the content selection and its presentation to enable the user to interpret the response quickly and completely, potentially trivializing the involved domain concepts into more abstract features. For this adaption the system considers (*a*) information about the user, (*b*) the information need of the user, i.e. the query, (*c*) available data, (*d*) conclusions derived by the ontological back-end, i.e. the result, and (*e*) relevance feedback. Of course, the back-end processing is already personalized as it involves the first three aspects of this list. This section focuses on how the computed output is personalized during its presentation. Referring to the steps in Figure 4.3, the following stages are directly personalized during output generation.

The knowledge base contains a broad variety of information after the reasoning step and depending on the aspects (*a*) to (*e*), not all of it is relevant to the user under each circumstance, e.g. the detailed ozone concentrations (*c*) can be omitted if they are too high in favor for the resulting threshold exceedance warning (*d*) that conveys the same information in an condensed form, except in the case when the user type (*a*) indicates that the recipient is an environmental expert that might be interested in the actual values. After

the content elements are selected for presentation, their mode of presentation is defined as either generated text, a visualization, or both. PESCaDO is capable of generating coherent, high quality natural language from the content of the knowledge base using intermediate communication models for discourse structuring [Bouayad-Agha et al., 2012]. A web-based environmental data visualization component was developed that employs different visualization techniques to show the environmental situation embedded in the base map of the same interface in which the query was generated, optionally to or accompanying the textual result description. This approach exploits the inherent geospatial nature of the data and at the same time presents the contextual data in the vicinity of the region of interest. The employed visualization techniques are designed to be parametrizable to allow for personalization of individual views, and be stackable for the concurrent display of different environmental information types to allow for a personal orchestration of data sources. If multiple environmental factors have to be displayed simultaneously, the system has to decide which factor will be mapped to which visual attribute (shape, size, color, position) without delimiting the interpretability of the other factors. This need arises from the goal of PESCaDO to orchestrate the usage of different data types and different data providers into a single decision support system.

4.3.1 Content and Mode Selection

The personalization of content selection means the configuration of the strategies used to select the most important subset of the produced information within the knowledge base (cf. Figure 4.3) from which the result is generated. This selection strategy is based on machine learning to allow a fully automatic selection of content elements in unforeseen configuration as well as to be able to influence the strategy with user feedback. Therefore, the training of the selection strategies was separated into two phases. First, the content selection service requires an initial strategy that is modeled using input from environmental experts. Second, the initial strategy can be employed in online user sessions to produce results for collecting user feedback for additional training. The personalization of the mode selection, responsible to assign a presentation mode to each selected content element, can follow the same two phase approach.

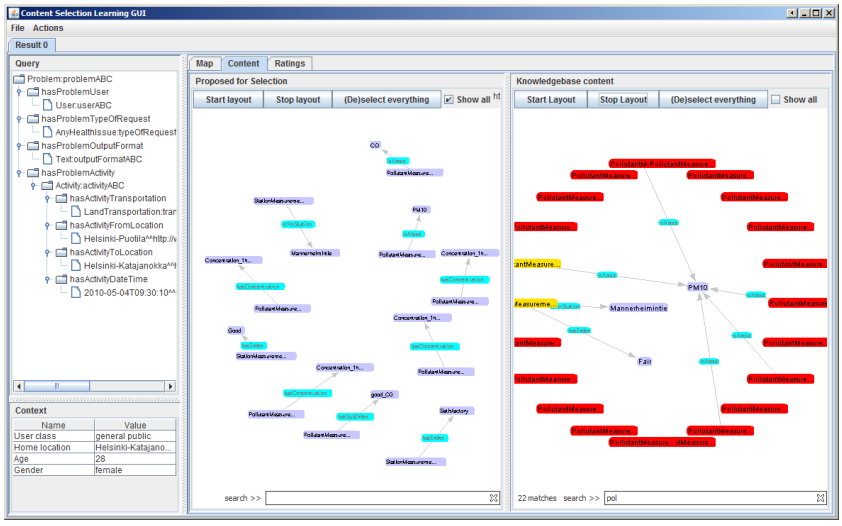


Figure 4.5 — User interface for the initial, offline reinforcement learning of the content selection strategy.

Interactive Content Selection

The setup for obtaining the initial strategy starts with generating a training data set consisting of a sparse sampling of the PESCaDO query parameter space, i.e. the options defined in the PDL, and their respective results under different data sets. The training of the selection strategy is based on reinforcement learning [Sutton and Barto, 1998] and models the selection task as a Markov Decision Process (MDP) [Bellman, 1957]. It can start with a random selection of content elements and present this choice to the expert for review. Using the interface depicted in Figure 4.5 the expert can assign weights to the presented content elements and include additional content from the knowledge base. After submitting the weights, the system incorporates the feedback in the selection strategy and presents an updated selection for review. This approach is iterated several times for each sample point of the training data.

The user interface for the expert review displays the query and user profile as the current context in a tree structure (in the left part) and the currently selected content elements and the contents of the knowledge base as graph structures (middle and right parts, respectively). The graph display allows for navigating the contents by textual searching and node adjacency. Each relation can be assigned a numeric weight for the computation of the reinforcement reward.

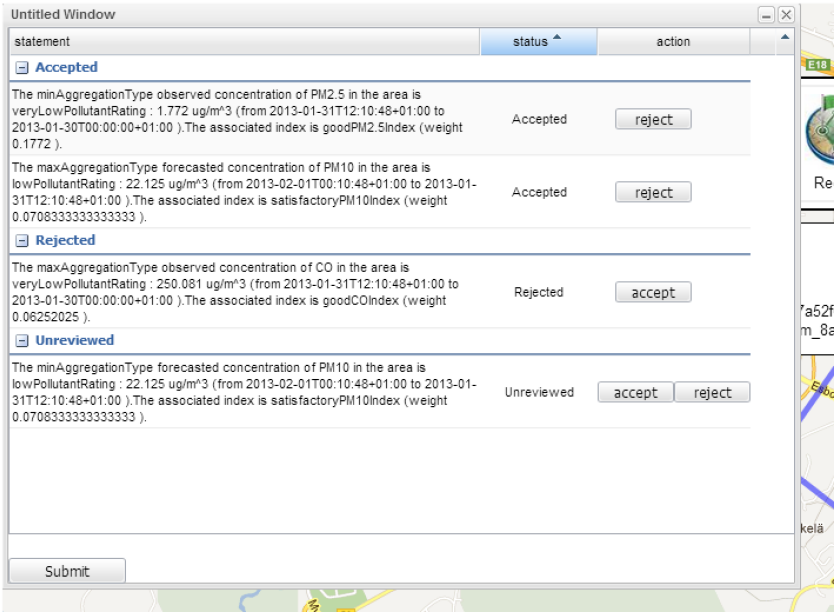


Figure 4.6 — User interface for the continuous, online reinforcement learning of the content selection strategy.

The online reinforcement learning is also embedded into the normal user interface as an additional dialog that is situated between the submission of the query and the presentation of the final result (see Figure 4.6). Because the generation of high quality natural language texts is time consuming and displaying the raw content elements from the knowledge base is infeasible for end users, a middle way is chosen by showing representative statements during the online rating of content elements. These statements are a mixture of stored text fragments and data that conveys the principle meaning of the content element, but is not optimized for good grammar and coherence. The system first presents the content elements that it deems most important and the users can accept or reject each statement as their feedback. The system incorporates the feedback and provides additional elements for review. This process is iterated until a final summary is presented which also contains statements that were originally discarded by the content selection strategy, but are offered for manual inclusion by the user. After the summary is accepted, the normal workflow continues and an elaborate natural language answer is generated in combination with the environmental data visualizations.

Mode Selection

The mode selection component receives the ranked content elements from the content selection services and assigns the ‘visual mode’ to those elements that have a high rank and can be directly linked to an available data source. This way, the most important findings for which visualizable data is available will be shown graphically. Additionally, the mode selection decides on the mapping of environmental data type to visualization technique to present a setting that has no overlapping use of visual channels. For instance avoiding that two environmental factors are displayed as heatmaps concurrently, which would make the interpretation of values impossible. The initial strategy of the mode selection component was crafted manually using a two-dimensional matrix of all possible combinations of environmental factors. Based on the experience gained from a user study of the visualization techniques (see Section 5.7), this matrix was filled by assigning a mapping from environmental factor to visualization type for each cell. If the content selection weights denote more than two factors as important, an additional visualization technique is chosen by iterating through a ranked list of visualization types for the factor until reaching a conflict free configuration. When users change this standard mapping during their session (see Visualization Manager in Section 4.3.3), this information can be tracked and taken as input for a later update of the default mapping either manually or with machine learning approaches.

4.3.2 Visualization Data Model

In order to be able to adjust the parameters and mappings dynamically, every visualization technique has to work on the same generic data model. Of course, not every technique is suitable for visualizing a certain data type, e.g. wind direction vectors cannot meaningfully be visualized with a bar chart display, but the implemented software interface to the data is the same for all data types. It comprises *Data Source* components to access the data in a standardized way, and *Data Objects* that contain the actual data which will be used by the visualization techniques.

Data Sources

Environmental information can be provided by the web resources and the data retrieval service in various types, resolutions, dimensionalities and with different degrees of uncertainty. The PESCaDO fusion service [Wanner et al., 2011] tries to unify some of these characteristics, but the variability is still

too high to be handled by the visualization techniques directly. Therefore, the Data Sources are used as a mediator that translates and unifies the data into a generic fixed resolution data format that offers the same access methods for every visualization technique and environmental data type. For instance, a view generating a temperature heatmap overlay needs to compute a color value for every pixel of the overlay, whereas the data retrieval or fusion service might only supply one value for the geospatial area of a city. The data source offers the functionality to extend this value to the screen resolution by either returning the same value for every requested location that lies within the boundary of the city, or by interpolating values between cities.

Each data source provides metadata about its environmental aspect in order to allow the visualization to adapt itself to the specific attributes. These metadata contain the name of the aspect, its unit of measurement, the specific data subtype of data that will be returned, and the thresholds for categorizing continuous data. They can be overwritten on a per-user level so that the color coding could be personalized to account for specific sensitivities of a user.

The communication between the Data Sources and the visualization techniques is modeled in an asynchronous way, which allows for placing the functionality of the Data Sources either in the server or the client component. A server-side Data Source can benefit from the increased performance of optimized Java code execution but also has to deliver higher resolution data over the network to the users. Client-side Data Sources can decrease the server's computation load and network traffic, but can also decrease the UI's responsiveness on low performance user devices. The PESCaDO prototype utilizes client side Data Sources.

Data Objects

Every data request to a Data Source is answered with an object of the abstract *Data* class. Different subtypes of this class model the different environmental data types. The two main data types are *simple data* and *complex data*. Simple data is further divided into having only a single value (ozone concentration) or an interval of possible values (e.g. min/max temperature). Complex data can be an array of similar data types (e.g. total air quality as an array of simple pollutant concentrations) or be arbitrarily composed (e.g. wind data as combination of single valued wind strength and a wind direction interval). Due to the importance of uncertainty in PESCaDO, each data object has an uncertainty score between 0 and 1 assigned from the environmental node repository and the fusion service.

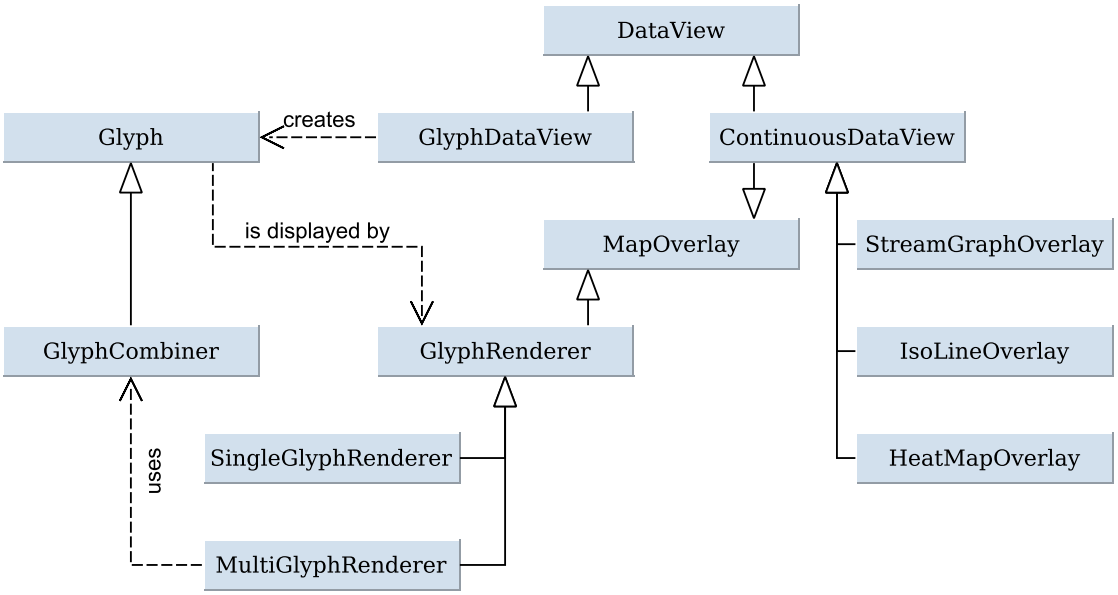


Figure 4.7 — Excerpt of the PESCaDO visualization framework classes. `DataView` is the class connecting to the Data Sources and provides data handling functionality. `MapOverlay` is the interface to the Google Maps framework and is used for displaying the views at the correct geographic coordinates.

4.3.3 Visualization Techniques

The visualization framework provides a set of web-based visualization techniques that are capable of representing environmental data on a base map. Some of the techniques are tailored to depict a specific data type, e.g. particle flow for wind data, but most are suitable for visualizing multiple types. Figure 4.7 shows an excerpt of the framework's structure. The two main visualization types are `ContinuousDataView`, showing continuous data (like heatmaps, isolines, particle flow, and diagrams), and `GlyphDataView`, showing data at sample points using glyphs (like weather icons, bars, and labels).

The first type always covers an area and is realized by a `MapOverlay`. The second type can create one or many glyphs to show data (a) at single locations, (b) at repeated locations along a route, (c) in a regular grid of locations in an area, or (d) in a separate view outside of the map. Each glyph is realized by its own `MapOverlay` via a `GlyphRenderer`. Additionally, glyphs can be grouped to display complex data values with multiple glyphs per location. The following sections will discuss each available technique and provide an example visualization. For illustrative purposes, the examples are generated

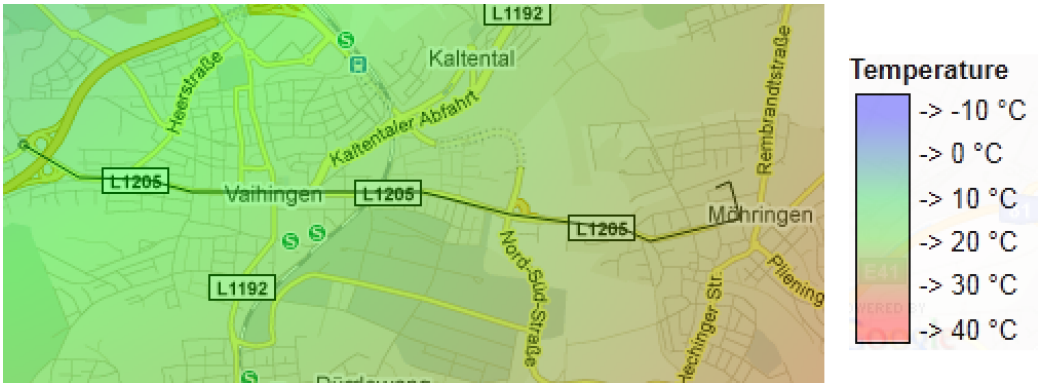


Figure 4.8 — A Heatmap of artificially generated temperature values and the according legend.

based on a fabricated data set and not with real measurements. A method for illustrating data uncertainty is defined for each visualization technique. They are inspired by or taken from the works of MacEachren [1992], Wittenbrink et al. [1996], Olston and Mackinlay [2002], and Hengl and Toomanian [2006]

Continuous Data Visualization

The spatial extent of these visualization types cover an area of interest and provide information about every covered point. Because both spatial dimensions of the screen are therefore utilized for displaying the data for one point in time, there is no room for representing the change of data over time with approaches such as small multiples [Tufte, 2001, chap. 8]. If multiple time steps are relevant (e.g. throughout one day or a weekend) the users could obtain an overview by browsing through the time frames either animated automatically or controlled manually using a slider control.

Heatmaps are a commonly known visualization technique that uses color to depict data in a two dimensional domain. Especially in displaying temperature values, they are frequently used for weather reports in mass media. In the recent years they have gained popularity also in the display of other data sets such as sports statistics.[10] An example from the PESCaDO prototype can be seen in Figure 4.8. The heatmap is drawn semi-transparently over the base map, which eliminates the need to redraw orientation guides like state borders or city names. The conversion from data values to color values

[10] http://resources.fifa.com/mm/document/tournament/competition/02/40/51/24/64_0713_ger-arg_playersheatmap.pdf

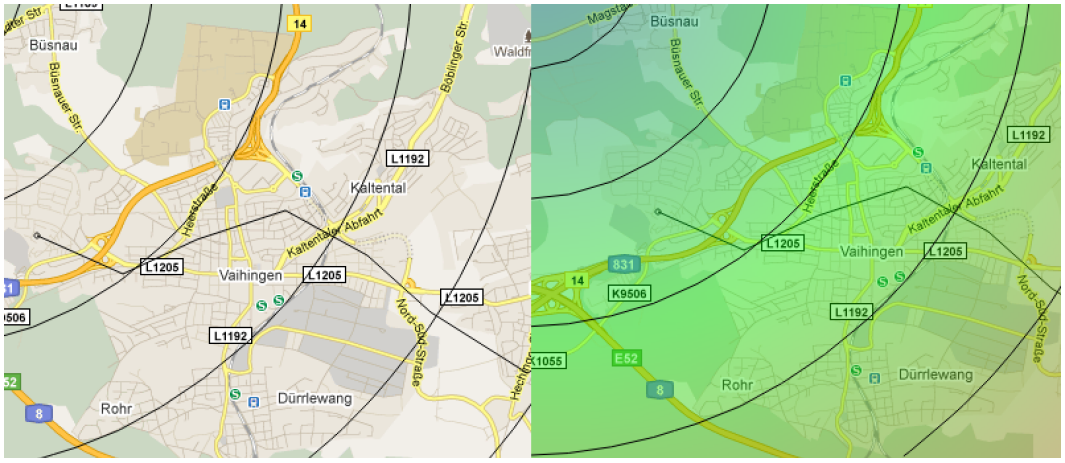


Figure 4.9 — Isolines of artificially generated data. The right image depicts the isolines overlaid with a heatmap of the same data.

('mapping') is based on the thresholds of the data source's meta information and can therefore be personalized to the current season and to each user, individually. This is beneficial for users that are visually impaired or are sensitive to certain environmental factors. The mapping is done for each point in the data grid the resulting image is subsequently scaled to match the current zoom level of the map.[11] Heatmaps are suitable for showing an overview and are mainly applicable to spatially dense data grids. For sparse data points, the continuous display resulting from interpolation can give a false impression of data availability. This can be countered by increasing the translucency of the heatmap if the data uncertainty is high. Data intervals can also be shown in a heatmap by using animation. However, as reading exact data values from color hues is already not optimal [Mackinlay, 1986], it becomes even harder when animation is applied. Also, the semi-transparent color composition can delimit the interpretability of the data if the background uses vivid colors. Here, desaturating the base map is an option.

Isolines depict points on the map where the visualized data equals a predefined value. For continuous data, these points create lines. Isolines are a well-known metaphor for representing atmospheric pressure as isobars or altitude as contour lines in topographic maps. They are created from a

[11] Performing the color mapping prior to interpolation to screen resolution is referred to as 'pre-shading' approach. If the data is smooth and its resolution sufficient, this approach is feasible. Otherwise, high frequencies in the data would cause an incorrect color interpolation deviating from the gradient of the color map.

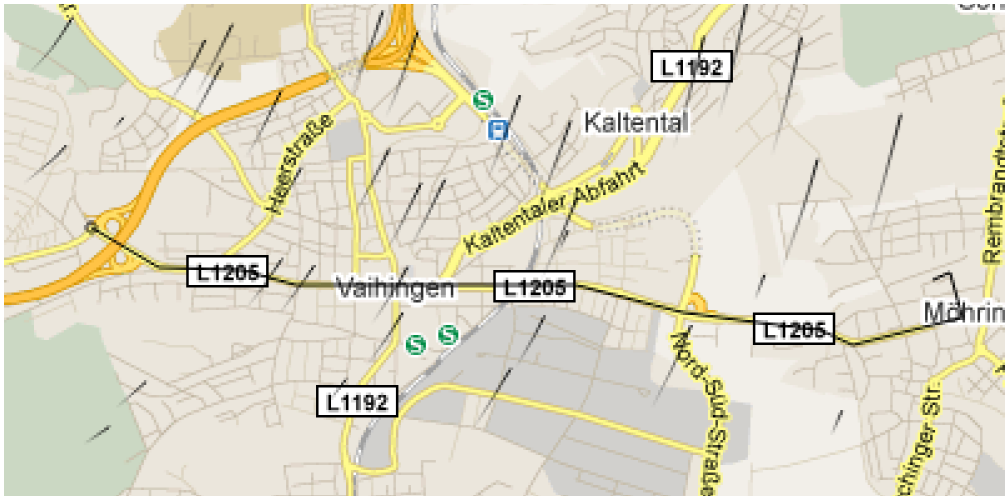


Figure 4.10 — Wind particle display showing strong wind from south west.

regular data grid with a variation of the marching square algorithm [Lorensen and Cline, 1987; Luo and Pan, 2009]. The lines can be drawn blurred to depict uncertainty, whereas there is no intuitive way to show interval data. Figure 4.9 shows an isoline display from the prototype. The advantage of isolines is that they occlude only limited screen space and do not use color. Therefore they can be used in combination with other, color-based data visualizations and different base maps.

Particle flow is used as a special visualization technique for wind data (c.f. Figure 4.10). Similar to isolines, it occludes little screen space and does not employ color variations. The view is generated by seeding particles at random map locations and let the wind vector field (strength and direction) virtually advect them. This creates a pathline for each particle that can be drawn on the map with an increased transparency at the beginning of the path to indicate the direction of the wind. The particles can be displayed as a static image in which only the length of the path denotes wind strength, or it can be animated by adding and removing additional path segments which creates a stronger sensation of the wind velocity.

Line Charts are a special case of continuous data view. Their display of data is continuous but their spatial coverage extends on one dimension only, i.e. along a polyline as a dynamic baseline for the chart. The line chart maps the data values to the orthogonal distance between cart line and baseline (c.f. Figure 4.11). This baseline can be given by a route that the user wants to travel, e.g. a cycling tour. Here, each point along the route can be assigned

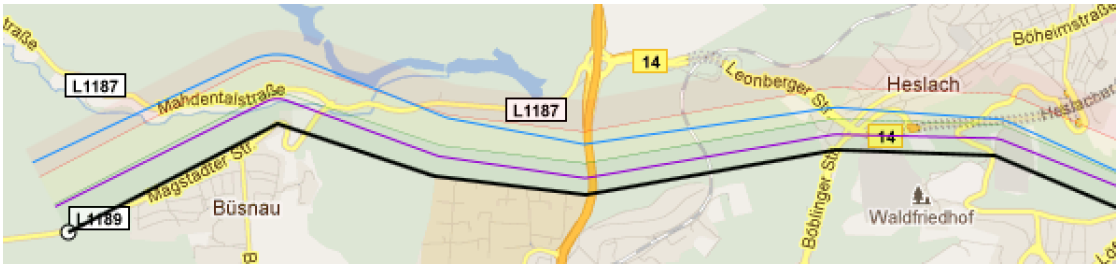


Figure 4.11 — A line chart display of artificially generated ozone and PM concentration along a user-selected route. The data are normalized according to three different qualitative indices (red, yellow, green).

a point in time based on the start and end time of the travel activity. This can be used to give an overview of the data both along the spatial as well as temporal domain in a natural way: the data displayed at a point in space is the forecast for the point in time at which the user will arrive at this location. The scale of the graph is either denoted quantitatively by labeling the vertical axis with the unit of measurement or qualitatively by introducing colored bands for different categories, e.g. *good*, *medium*, *bad* air quality bands, and normalizing the data values into these categories. The latter approach allows the combination of different environmental aspects into the same graph if they can be normalized into equivalent categories. The meta information of the source is again used for the normalization thresholds which allows for user level personalization of the graph. The graph line can also be extended to a band to show interval data and it can be drawn blurred, transparent or dashed to depict the uncertainty of the information.

Glyph based Visualization

Glyphs denote data at singular locations from one data object. However, they can be spatially extended by repeating them along a route or in a regular grid within an area, thus, mimicking the behavior of the continuous data views. Additionally, they can be moved to a separate view outside of the map while showing data from, e.g., the current location of the mouse cursor on the map. The repetition of glyphs has the potential advantage that each glyph in the view can depict a different point in time, thereby eliminating the need for exploring a time range manually. Same with the line chart, an intuitive mapping from geocoordinate to time is needed for this. The prototype allows for creating a route with timestamps for each waypoint. This simulates the

Figure 4.12 — Wind arrows depicting strength (size), direction interval (arc and arrow direction), and data uncertainty (blur).

actual travel/hike and shows the data of the time on which the users will be at the given location.

Labels are a very common and direct way to convey information to users. The data value with its unit of measurement is placed at the specified geographical location. When sampling an area by multiple labels, one has to avoid overlapping with other labels or lines from other visualizations. Therefore, the number of labels that can be used for the sampling of an area or route correlates with the label size. In PESCaDO the user can change the label size which causes a change in the sampling rate.

Weather Icons are generic glyphs that map the data values that would otherwise be displayed as labels to individual icons. This kind of data display is established very well in the environmental data domain. An example of the PESCaDO prototype are wind arrows (c.f. Figure 4.12), in which the average wind direction is mapped to the rotation of the arrow and the possible directional interval is mapped to an arc in front the arrow [Wittenbrink et al., 1996]. The magnitude is mapped to either the glyph size or the speed of a rolling animation. Uncertainty is mapped to the intensity of a Gaussian blur.

Bars behave similar to line charts, but can also be used for single locations because they do not need to connect at least two data points to draw a line. They map data values to their height, which makes them suited for almost any environmental measurement. They can denote also negative values by extending below the baseline, intervals by adding error margins, and uncertainty by blur, color variations, or opacity. Figure 4.13 shows how the orientation of the glyph can be changed according to the principle direction of the baseline to avoid overlapping and at the same time provide stable sampling rates.

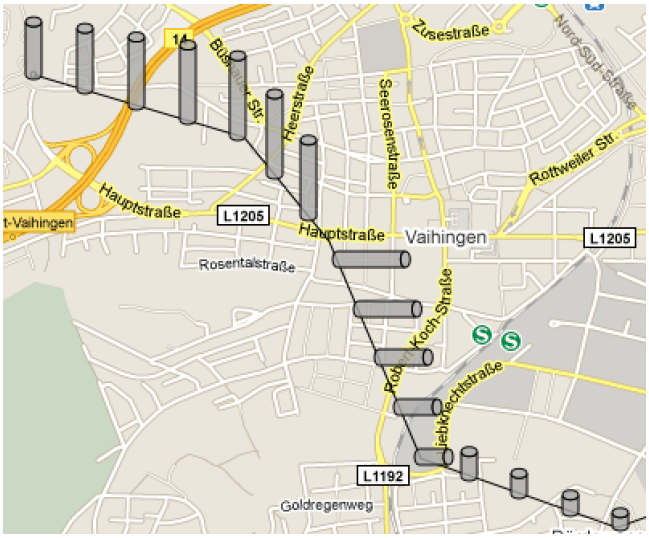


Figure 4.13 — Bars along a user-selected route depicting artificially generated air quality data. The orientation of the bars is adjusted to allow for a constant sampling rate and non-overlapping bars.

Visualization Manager

All visualization types and available data types are registered at the Visualization Manager component. Some of them supply additional control widgets for integration into the user interface. Among the default control widgets are slider controls for the adjustment of the visualized point in time and the icon size. Individual visualizations can add their legends as custom control widget to the map interface. Also, the visualization manager has its own control interface that allows the disabling and enabling of visualizations and controlling the mapping from data type to visualization technique for further personalization. Figure 4.14 shows a combination of different environmental data types using multiple visualization techniques in one interface. The user acceptance and interpretability of single and combined visualizations was evaluated in an online user study (see Section 5.7). Table 4.2 summarizes and compares the mentioned visualization techniques for their suitability to the PESCaDO problem domain.

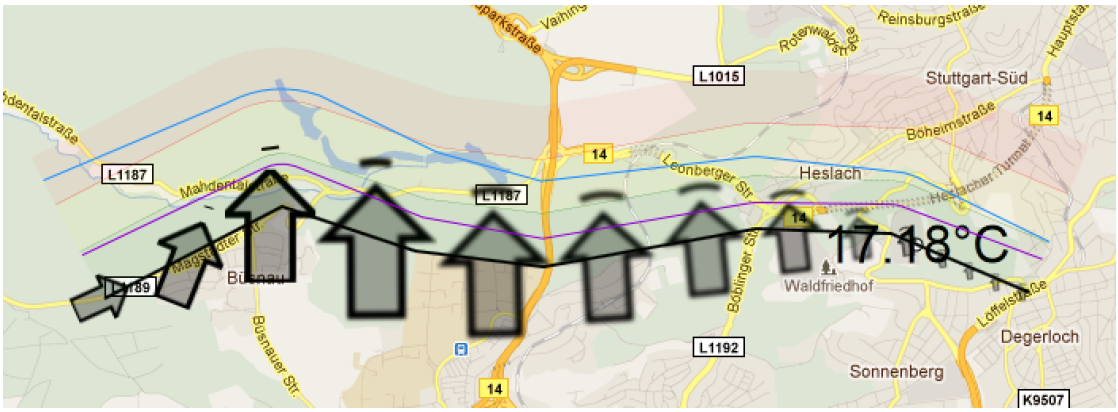


Figure 4.14 — A combination of glyphs, labels, and line charts; overlaid along the user-selected route.

Continuous Data Views	
Heatmap	Isolines
👍 continuous and sparse data 👍 good for overview 👍 uncertainty ⇒ transparency 👎 time series need interaction / animation 👎 interval data is problematic 👎 color usage problematic for combination	👍 good for overview 👍 little occlusion 👍 uncertainty ⇒ transparency / blur 👎 only for continuous data 👎 time series need interaction/animation 👎 interval data is problematic
Particle Flow	Line Chart
👍 little occlusion 👍 uncertainty ⇒ transparency 👍 interval data mapped to jitter during animation 👎 single purpose (wind data) 👎 time series need interaction 👎 interpretation can be problematic	👍 continuous and sparse data 👍 whole time series in one view 👍 good for overviewand details 👍 intervals mapped to bands 👍 uncertainty ⇒ blur 👎 interpretation of normalized data requires learning • little occlusion but color bands

Table 4.2 — Comparison of the visualization techniques for their suitability of the environmental decision support visualization.
👍 indicates an advantage,
👎 indicates a disadvantage, and
• indicates a neutral statement.

Glyph Data Views	
Labels	Icons
👍 suited for all data types 👍 used as area overlay, single locations or separate view 👍 good for details 👎 but not for overview • questionable use of exact data values for end users • combinations may reduce readability	👍 suited for all data types 👍 used as area overlay, single locations or separate view 👍 good for overviewand details 👎 usually single purpose icons necessary (e.g. wind arrows) 👎 interpretability depends on how well the icons are known
Bar Charts	
👍 suited for many data types 👍 little occlusion 👍 good for overviewand details 👎 height comparison problematic if used at multiple locations 👎 interpretation of exact data values problematic	

Table 4.2 — *(continued)*

Chapter 5

Results and Discussion

This chapter presents the evaluations, measurements and results of the presented prototypes and approaches. Depending on the subject, this ranges from online user studies and performance measurements over challenge reviews to expert evaluations and discussions. Using different evaluation procedures was necessary, because of the different accessible audience for the individual prototypes[1] and the problem of evaluating complex visual analytics systems.

5.1 Evaluating Visual Analytics Systems

Scientific research fields such as Natural Language Processing (NLP) and Information Retrieval (IR) often have evaluation tool sets for each of their major tasks. Usually, two basic ways exist for evaluating a new approach in these fields: (*a*) testing it against a known data set with a 'gold standard' or 'ground truth' result, i.e. the expected results have been manually created and the approach's results can be compared automatically against them, or (*b*) comparing the own results within a targeted application domain against other approaches from the literature. This requires a list of reoccurring tasks within the field in order to have appropriate test data and compatible approaches

[1] The PESCaDO related prototypes are the only ones with a web front-end and thus can be tracked and used by many users in parallel.

to compare against. To stay with the example of NLP, these tasks could be named entity recognition, part-of-speech tagging, text summarization, etc.

Visualization, especially interactive visualization and visual analytics, are interdisciplinary research fields involving concepts from computer graphics, Human-Computer-Interaction (HCI), psychology, data mining, and machine learning. Potentially, each of these fields influences the outcome of an interactive system creating a vast parameter space for designing one's solution. Similarly, the employed measurements for results differ among the fields and range from the quality and count of gained insights, over precision and recall, to task completion times. Individual approaches are hard to compare because one may not dominate the other over the entire field of measurement options. Success in one category may be hard to attribute to a certain design decision. Additionally, visualization is an applied science and a supportive task which also draws its research questions from the application domains. Resulting research prototypes are therefore often novel in the sense that no appropriate baseline approach is available to compare against.

Naturally, visualization techniques and interactive systems cannot be evaluated without a human in the loop. Repeated experiments will most likely lead to different results, even with the same participants due to learning effects. Saraiya et al. [2006] summarize their findings from a longitudinal study with two bioinformaticians using multiple visualization tools:

> From the discussion, it is clear that the choice of visualization methods used to analyze the data is based on the subjects' domain knowledge. Discovering an appropriate visual representation and procedure to interpret the data could be considered procedural insight. This is usually a nontrivial task, and requires trial-and-error attempts with many combinations. The subjects reported that in the future they will be able to analyze a similar data set in a relatively shorter time. Such use of learned domain knowledge is very difficult to reproduce in short-term experiments.

Even small irregularities in the usability may lead to a devastating evaluation result because the research prototype cannot compete with optimized commercial products in all involved facets at once. In the HCI domain, approaches are dismantled into clearly defined and limited subtasks to allow for studies with many experiments in order to compensate the variance of individual user results. However, visual analytics approaches depend on the interplay of several mechanisms that cannot be regarded in isolation and widespread user studies are infeasible due to the task and tool complexity. Most of these

issues regarding the evaluation of visual analytics approaches can also be found in a list of common problems compiled by Koch:

- duration of tasks
- diversity of domains
- availability of domain experts
- lack of suitable (large enough) test data
- lack of ground truth data and gold standard data
- lack of comparable VA approaches
- lack of VA approaches' maturity
- lack of suitable evaluation criteria [Koch, 2012, chap. 6]

As a result, novel approaches for evaluating interactive visualization techniques are an ongoing research topic and the BELIV[2] workshop provides an overview of recent works. Greenberg and Buxton [2008] argue that usability studies can hinder the innovation if applied too early in the development cycle, and that other evaluation approaches, fitting the research prototype, may have to be chosen instead. They also provide several examples how cultures adopt new technologies that would be considered not usable during the time of their invention. This is related to the nested validation model by Tamara Munzner [2009]. It consists of four levels going from the domain problem characterization down to the algorithm design. Similar to the V-model of the software development life cycle, each level has a corresponding validation task and errors in more abstract levels propagate down to the algorithm design. Here, informal usability studies are stated as a way to validate data abstractions such as the visual encoding, but they would not be appropriate to validate the correct understanding of the domain problem or the algorithm complexity.

A different point of view is presented by Plaisant et al. [2008] and Scholtz et al. [2012], recapitulating over several years of InfoVis Contests and VAST Challenges. Inspired by the success of TREC [Voorhees, 2002], they designed competitions to promote the research fields of information visualization and visual analytics, by supplying data sets, scenarios, tasks, and ground truth results. These challenges are provided in a benchmark repository and allow researchers to mimic realistic analysis processes based on the provided tasks which are an incentive to perform a long-term analysis using their own software. The benchmark repository is also a common ground to compare the

[2] Beyond Time And Errors: Novel Evaluation Methods For Visualization: `http://beliv.cs.univie.ac.at`

effectiveness of own approaches against other contest entries or published uses of the data sets. Additionally, the VAST Challenge entries are reviewed by peer researchers as well as analysts to provide feedback from both points of view. The selection management applications that related to challenge entries will be presented using the feedback of the reviewers next to our own insights from the analysis process.

Arias-Hernandez et al. [2011] propose *Pair Analytics*, an approach loosely based on the agile software development concept of *pair programming*. Two users with different expertise form a team to solve a task using visual analytics software. One participant is the dedicated visual analytics expert (VAE), the other is the subject matter expert (SME). From the perspective of capturing cognitive processes during the study, the paired approach is beneficial because the participants naturally articulate domain knowledge, findings, and strategies in order to function as a team, instead of having to 'think-aloud' artificially for the sake of the study. From the perspective of evaluating visual analytics approaches, one can overcome the usability problems of early research prototypes by having a VAE who is familiar with the software and can mitigate usability problems while the SME can still evaluate the task effectiveness of the approach. Of course, in such a setup, it has to be assured that the VAE does not steer the analysis and is restricted to implementing the requests of the SME. Such an approach was taken to evaluate the *ScatterBlogs2* system with crisis response domain experts.

5.2 VAST Challenge Feedback

Two of the presented scenarios are based on VAST Challenge entries and this section will discuss the outcome and insights chronologically.

5.2.1 VAST Challenge 2009

The application of the filter/flow graph to uncertain set-membership problems such as matching roles of a network pattern was developed for the VAST Challenge 2009 Mini Challenge 2 (see Section 3.6). In 2009, the tool was part of both the individual mini challenge entry as well as the grand challenge entry drawing conclusions from each mini challenge. Overall, five reviewers provided feedback to the hypothesis graph display. Two of them in the context of the grand challenge and three of them for the mini challenge. The reviewers provided free comments to whatever parts they deemed relevant

but also had to provide numeric scores for some explicitly mentioned aspects, i.e. explanation of the solution, analytic process, visualization, interactions, novelty, and overall satisfaction. The average score for each aspect varied between 6 and 7 with 10 being the highest possible score. This scores were contrasted with rather enthusiastic comments and the entry was given awards for being an “innovative analytic tool” and an “excellent example of analytic tradecraft.”

All reviewers noted that the methodology of the approach and its presentation and explanation was flawless and highlighted the interactivity of the solution. Most (4/5) of them stated that the application of rules by dragging and dropping them in the hypothesis graph was very intuitive. One reviewer disagreed completely and described the graph as being “totally opaque and counterintuitive.” As the interactions with the prototype were described in the submission using text as well as video, it is possible that this reviewer did not have access to the video component showing the animated interaction with the graph. He/she would therefore be left with only the screenshot of the graph in its final configuration that was included in the text, without having seen its iterative creation. Nevertheless, this reviewer also considered the methodology sound and the graph valuable if usability improvements would be made.

The following observations have been stated by one or two reviewers. The free exploration using the workspace provides for an easy to learn tool. This may be due to the fact that wrong decision simply lead to dead-ends in the graph, but do not break the analysis as one can continue to add rules at any prior node on the path. This decision to leave dead-ends visible was also mentioned as a good way to support an understanding of the thought process that lead to a conclusion. The same reviewer also deemed the resulting ‘process diagram’ clear enough to explain the analysis to others. This remark was an impulse to include more sophisticated reporting functionality in later prototypes. Direct manipulation by dragging nodes and adjusting confidence at the nodes was recognized as very useful and good way to integrate the fuzzy rules. Here, the instant feedback through changing levels in the result node’s bars was also highlighted as an easy way to identify very selective filters.

Next to the overall positive feedback, two questions were raised. One was concerning the interactive creation of the fuzzy rules and can be answered with the fact that these rules were hard-coded, but it would not be much effort to create new rules by filling in the placeholders with roles and number ranges. The second question regarded the rather simple structure of the

sought-after network, forming a chain between the roles of the scenario, and if more complex subgraphs could also be identified with the proposed approach. While this question is of course valid, the applicability of the approach would actually be increased if more relations between the roles would exist as these would lead to additional rules that could be exploited to reduce candidate sets. With the current chain-like structure, each role has at most three relations that can be used: two for the neighboring roles and one to the generic `contact` role. Much more problematic would be the omission of statements that allow the initialization of the candidate sets using the number of contacts that a member of a role might have (e.g. employee roughly 40, leader well over 100). At the start of the analysis every account is a candidate for each role and any statement about the relations between the roles would be useless without the initialization.

5.2.2 VAST Challenge 2011

In 2011, mini challenge 1 of the VAST Challenge was concerned with geospatial social media messages and selection management was an essential part of our entry (see Section 3.1). For this entry, two reviewers provided feedback and, again, were allowed to include free comments as well as rate the clarity of explanation, the accuracy of answers provided for the scenarios tasks, visualizations, interactions, novelty, and overall rating. In most categories, the reviewers unanimously rated the submission ‘excellent’ and the given answers as ‘accurate’ or ‘very accurate’, with the exception of the novelty criteria which was rated with ‘moderate’ or ‘significant novelty.’ Due to the length limitation of both the text and video part of the submission, the semi-automatic report generation could not be presented prominently enough. It was only shown during the last seconds of the video. This might explain why no comments on this important aspect were included.

Besides rating these categories, the reviewers highlighted several aspects of the submission. One reviewer particularly liked the way of eliminating the possibility of person-to-person transmission, which was heavily relying on the selection management component to constitute and compare the related user and message sets. This was particularly important as many disease-related blog messages were posted by people other than the patients themselves which obstructs approaches based purely on the textual content. Due to the artificial nature of the data set, identifying re-occurring text fragments would have also been an option to distinguish the sets. However, this would not be the case for a real-world scenario as can be seen by the necessity to include

more complex message classification and filter orchestration approaches in *ScatterBlogs2*.

Access to the raw data is often necessary to validate the impression that an analyst gets when being confronted with aggregated and preprocessed data. It was stated in the reviews that the combination of the extracted keyword cloud with being able to drill-down to the related messages makes the analysis very straight forward. This is an example of the high interactivity of the prototype in which every visualized element could be used to cause a selection of related messages. Accordingly, a reviewer stated that the free navigation and exploration of both spatial and temporal domain is well supported.

The presentation was stated as perfectly representing what decision makers need during a crisis. This was also due to the manner in which time and geospatial elements could be incorporated into hypotheses and their validation using the selection management approach. The overall submission was awarded for its “unique integration of tag clouds in geo-spatial visualizations”.

5.3 PatViz Insight Reintegration

The *PatViz* system was evaluated in three different ways to show the suitability of the design decisions. (*a*) A questionnaire was used to evaluate the visual metaphors employed in the interactive and visual query editor. While this component is not directly involved in generating insights about a patent document collection, an effective query structure visualization is required for integrating the obtained insides into the analysis cycle. (*b*) During a ‘think-aloud’ study, the whole analysis process including the result set exploration and insight integration was examined with domain experts. (*c*) Over the course of the *PATExpert* project, the prototype was presented several times to audience of patent specialists and peer researchers.

A questionnaire was send to fifteen people containing depictions of different metaphors used in the visual query editor and some example queries in textual and visual form. The metaphors described the Boolean combination of query fragments by branching or by a sequential concatenation. Two alternatives were presented either using branching for AND and concatenation for OR operations or vice versa. All participants were familiar with Boolean operators as the group was composed of patent search specialists and computer science students. Most evaluators preferred the branching for both Boolean operators alike, while a third of them preferred a sequential order

for the AND. However, none of them had difficulties to interpret any metaphor correctly in the example queries. The invariance of the two metaphors is probably due to the fact that for the computer science participants both are just n-ary operators. More importantly, the test persons were all able to correctly recognize operator scopes. This is already a benefit over text-based query representations and is essential for working with more complex real-world queries and being able to place integrated insights at the correct position.

For the think-aloud study the participants required knowledge about the patent search process and the domain of the patent document sets. Due to the indexing effort, the patent document database of *PATExpert* was limited to two domains, 'optical recording' and 'machine tools'. Therefore, the participants had to be chosen from the project consortium because these were the only available experts with the required expertise. During one analysis session, the participants were asked to 'think-aloud' while carrying out the same analysis tasks they are performing in their daily work. It became quickly apparent that the practitioners were most familiar with form-based queries resulting in result lists with some additional annotations and aggregations. Therefore, a MCV environment such as *PatViz* required an initial training phase. Starting with a subset of the views consisting of the tag cloud, charts, and world map, the more sophisticated views could be included over the course of the analysis session. Providing a wide variety of views on the same document set therefore was beneficial to scale the complexity of application to the user's abilities and training level. In the end, the interlinked, interactive views were one of the most appreciated system's properties. The other one was the iterative refinement of queries and patent sets. Each submitted query results in an new set of views allowing to go back and forth within one result set (using the selection management component) as well as in the overall process (going back to an old result set and its views).

5.4 ScatterBlogs2

The *ScatterBlogs2* system was evaluated using questionnaires as well as a 'pair analytics' approach with several domain experts. These evaluations are limited to the real-time monitoring environment. The task applicability of the filter creation and the real-time scalability will also be discussed. Results on the filter creation using interactive classifier training can be found in the evaluation of the environmental search engine (Section 5.5), because the it employs the same approach.

5.4.1 Questionnaire Feedback on Monitoring

A questionnaire was composed of various statements to which the participants could state their level of agreement using a five point Likert scale [Likert, 1932] and provide comments. It was concerned with the requirements for real-time situational awareness, the system's usability, and its application to disaster management scenarios. The questionnaire was presented to a disaster management expert (DME) from the German Federal Office of Civil Protection and Disaster Assistance (BBK) as well as to a usability consultant (UC). Both were introduced to the capabilities of the system by a twenty minute presentation and a subsequent questions and answers round. The following paragraphs summarize their judgment.

Both experts agreed with the overall description of the domain's requirements, especially on the importance of having a real-time monitoring system during disastrous events for creating situational awareness. They also agreed with the high importance of temporal, spatial, and keyword filtering, and the combination of these methods. While the UC was undecided on the role of historic data for event filters, the DME strongly agreed that the quality of the filters should be evaluated on past events. Their opinion also differed on how long old information should be kept visible. Here, the DME favored a user-controlled setting on how long filter hits and unrelated messages should be visible, while the UC voted for removing unrelated messages after 30 minutes. Additionally to the provided questions, the UC also commented that reporting functionality with snapshots and exports are important for this task.

Both experts felt that the presentation was not sufficient to rate the usability of the prototype, but while the DME skipped this part of the questionnaire, the UC provided an informal estimate. She highlighted that the system's components complement each other very well and that the spatial display of the information contributes to gaining insights into the current situation. She doubted that one could use the system and the filter orchestration graph efficiently after the short presentation. Even when knowing what each part of the system does, choosing the right functionality to fulfill a task, especially under the time constraints during a crisis, requires training.

The feedback on the system's suitability for monitoring events as part of disaster management was rather homogeneous, with one exception. While the DME felt confident to use the system in addition to other tasks, the UC disagreed in that point, stating that the system would require the whole attention of the user. Both agreed that a good overview of the incoming information is given, situational awareness is provided, and that the system would

fulfill an existing information need for disaster management. The display of temporal patterns was rated as not sufficient. Indeed, there currently is no aggregated view of the filter hits. This could be addressed by including the tag icons or their color into the timeline histogram. Finally, the DME mentioned that she would like to employ the system for disaster management tasks.

5.4.2 Pair Analytics Feedback on Monitoring

ScatterBlogs2 was presented to employees of four electric power companies and three regional crisis management authorities in separate pair analytics sessions. Both groups are familiar with control room environments that aggregate many information sources about either a power grid or crisis situations. For the evaluation, a data set of a recorded flood event from 2013 was analyzed in cooperation with the domain experts. Their opinions on the prototype were recorded throughout the analysis and during an open discussion session afterwards. Again, the participants noted that they would like to spend more time with the system and solve some additional tasks to be able to judge the system properly. Both user groups see the approach as auxiliary source of information which provides a detailed additional view on a situation. This is valuable as the control room analyst is rather detached from the situation in the field. While the energy domain experts (EDE) see this foremost as a way to search for reasons of problems in the power grid, the DMEs saw a problem with very emotional messages that could have demoralizing effects during a crisis. However, they considered this information source as being faster than their traditional reporting channels and also valuable for local administration for coordinating the help of supportive citizens.

The desired outcome of an application of social media monitoring would be the assessment of the current public opinion, mitigation efforts, and communication channels according to the DMEs. The EDEs have less need for overview and context assessment and are more interested in the situation at a specific location. This is due to the fact that the measures to indicate problems in the power grid already exist, but not for finding the cause. Therefore, the term cloud and time overviews were not deemed important, and it was considered problematic to depend on the social media users to provide the needed information. However, the user group was open for new visual analytics approaches to incorporate into the control room. Finally, the domain experts asked for incorporating other social media sources, adding reporting functionality, and using the customary symbols of the related domains.

5.4.3 Filter Creation and Application

The motivation for using filter methods beyond plain keyword lists and metadata restrictions is generalization and customizability by adapting thresholds. The experience with the presented prototypes has shown that the performance of such filter methods, with respect to the perceived accuracy for detecting certain events and message types, is rather different. For some information needs, it is possible to create good classifiers and statistically motivated keyword lists, while in other situations an acceptable performance is hard to achieve. The same observation applies to the application of well-established methods such as spatiotemporal restriction and direct keyword filtering. All of the mentioned techniques are useful on their own, but typically a combination of them is beneficial. In general, most monitoring tasks aim at a high recall, meaning that it is important not to miss an important message. Achieving this, however, often requires a trade-off regarding precision. As a consequence *ScatterBlogs2* provides the capability to train, combine, and configure classifiers and filters to let analysts decide on this trade-off based on domain knowledge and the current situation.

Nevertheless, the creation of a good filter combination requires some expertise and can lead to unintended effects if not done properly. Care should be taken, for example, if keyword filters are applied before machine learning based methods. The distinct cut-off of the former ones can invalidate the generalizability of the latter ones because the data stream is already too focused. However, in times of data abundance such a strategy can be needed nevertheless to be able to use computationally intensive filter types on the reduced data stream. Building good filter sequences can therefore be seen as a creative act, which requires testing different filter combinations. Accordingly, it is important to provide a facility such as the filter/flow orchestration that enables analysts to create and test filter combinations on their own and within the application environment.

5.4.4 Scalability and Performance

The development of the *ScatterBlogs2* approach focused specifically on scalability aspects. These aspects include reducing the costs of filter creation, fast and user-steered filter application, as well as real-time monitoring. In addition, the separation of filter creation and their application allows the specialization of users on different domains and tasks, such as classifier training or situation monitoring. Once created, generic filters can be used in an large

number of monitoring sessions and scenarios.

In order to make filter processing and orchestration scalable, all stages of the user-defined filter-pipeline are parallelized. Here, the benefits vary depending on the complexity of the filters to be evaluated. Support Vector Machines with a linear kernel can be evaluated very fast on short documents such as Twitter messages, limiting the benefit of parallelization due to the management overhead. The most critical points in the pipeline are those filters working directly on all incoming messages of the graph root, because subsequent nodes will receive considerably less input data. Table 5.1 contains performance evaluations of single filter types as well as a whole graph structure. The graph is composed of three linear SVM classifiers and one keyword metric filter, which directly work on the whole input data. Their output is joined by an OR/Union node and ends in a geographic filter. The performance was measured on 3,9 million messages using a machine with forty physical cores. The geographic filter node was reached by 31 thousand messages.

It can be seen from these figures that the employed techniques are fast enough to have several filter instances running on the global stream of georeferenced Twitter messages. A computer with one core could evaluate multiple linear SVMs or keyword-based filters on the daily average of 10 million Tweets. The only exception is the string kernel based SVM classifier, which should only be used on already reduced streams. This can be achieved, e.g., by constraining the incoming stream to a narrow location of interest. The presented approach for monitoring messages differs from many others, since it does not restrict the Twitter stream based on keyword filtering, but processes the complete available stream of geolocated messages. Using a content filter, e.g. by using Twitter's filter stream API, or querying messages

Filter type	Time/Message 1 Thread	Time/Message 40 Thread
Linear Kernel SVM	19 µs	6 µs
String Kernel SVM	400 ms	87 ms
Keyword Metric	20 µs	6 µs
Whole Graph	–	73 µs

Table 5.1 — Average time needed for evaluating a filter on one microblog message when evaluating them sequentially or in parallel. The *Whole Graph* is composed of multiple filters and due to the inherent parallelism of the graph structure, no value is given for one thread.

according to a specific key word list, eliminates the chance of finding related messages with different wording, which might be detectable with generalizing machine learning approaches such as classification.

5.5 Environmental Search Engine

The interactive training of classifiers by using visual depictions of the data and its relation to the current model has several benefits. Primarily, the visualization is used to assess the classifier's performance and thus provides insights into the capabilities and drawbacks of the resulting classifier and builds trust into the final product. At the same time it is an incentive to apply the active learning methodology and combines the labeling of influential instances with labeling larger sets at once. It was evaluated by comparing it against automatic baselines, basic active learning using uncertainty sampling, as well as a hybrid approach using active learning and model visualization at the same time [Heimerl et al., 2012]. It has been shown that the fully user-steered approach can perform equally well as the active learning based approach when comparing the F_1-score evolution over the assigned labels, but has the aforementioned benefits of trust building. Manually choosing the labeling candidates and assigning labels based on aggregated information, however, is more vulnerable against improper usage if the mechanisms of the tool set are not clear to the user.

With the basic methodology proven, an experiment with expert users (i.e. with knowledge about classification tasks and environmental aspects) was conducted to demonstrate its applicability for optimizing the environmental search engine classification task. The initial classification model (M_0) was constructed using a training set defined by environmental experts, which comprised 99 positive and 200 negative website examples. The variability of web resources requires a frequent assessment and retraining of the classification task based on the observed results. The visual and interactive environment provides the necessary functionality to achieve this.

5.5.1 Evaluation Setup

Due to the need to familiarize with the visualization tool and the nature of environmental data sources, the user study was limited to six users. An increased data set containing 2329 websites with gold labels (1692 irrelevant and 637 relevant) was constructed using eight different queries to a

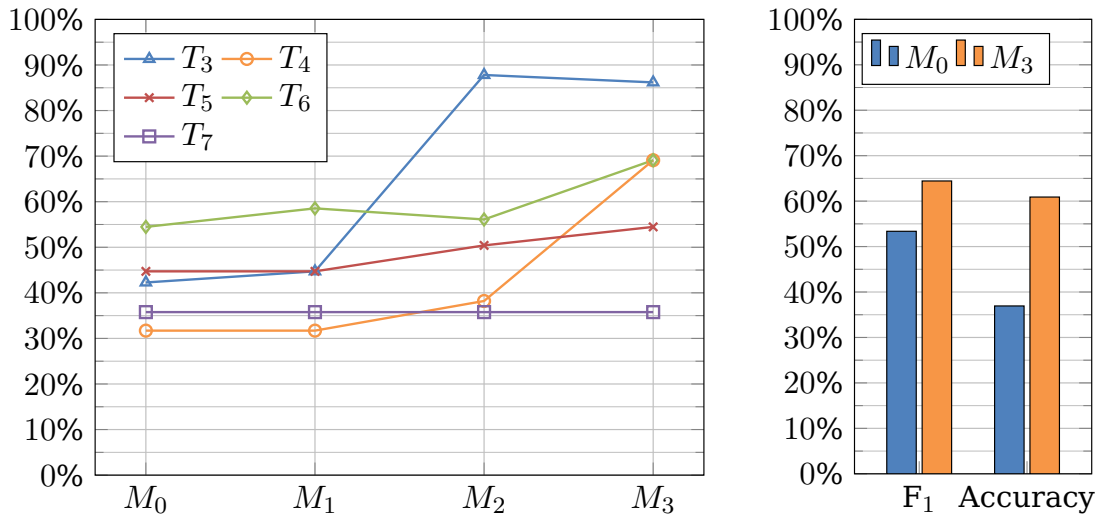
general purpose search engine. In order to rule out overfitting effects during performance testing, a cross-validation inspired experiment setup was used. The data was divided into eight sets $T_{i\in\{1..8\}}$ derived from different indicative queries (e.g. T_1: `weather` + `Helsinki`).

The test procedure was set up as follows: Each participant $(P_{j\in\{1..6\}})$ worked on an individual sequence of three consecutive data sets (T_j, T_{j+1}, T_{j+2}). Initially, the participant were provided with the visualized result of M_0 for T_j and performed the interactive training to receive a new classifier model M_1^j. The resulting model was then applied to the second data set T_{j+1} and another phase of interactive labeling generated the model M_2^j. The same approach was repeated once more with the data set T_{j+2} to obtain the final model M_3^j. The final model can then be tested against the remaining sets that were not used for interactive training. The test users had as much time to familiarize themselves with the tool as they wanted and could abort their participation in the evaluation at any time. After the initial tutorial phase, the users had 5 minutes to perform the interactive labeling for each model creation (i.e. 15 minutes in total for each user). The setup for all participants and data sets is shown in Table 5.2.

For instance, P_1 initially is confronted with a visualization of M_0's classification for the result set T_1 and evolves the model into M_1^1 by adding new labels to its training data. He/she is then presented the second result set T_2 and further improves the model to M_2^1. Respectively, M_3^1 is generated in the context of T_3. This final model is evaluated against every set which have not been used in its training ($T_{4...8}$).

		Data sets (D)							
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
Participants (P)	P_1	M_0	M_1^1	M_2^1	M_3^1				
	P_2		M_0	M_1^2	M_2^2	M_3^2			
	P_3			M_0	M_1^3	M_2^3	M_3^3		
	P_4				M_0	M_1^4	M_2^4	M_3^4	
	P_5					M_0	M_1^5	M_2^5	M_3^5
	P_6			M_3^6			M_0	M_1^6	M_2^6

Table 5.2 — Overview of the models used for evaluating interactive training for website classification.



(a) Accuracy Evolution

(b) Average Scores

Figure 5.1 — (a) Classifier accuracy fluctuation for T_3 to T_7 over the evolution of the trained classifiers through the models M_0 to M_3^j. (b) F_1-score and accuracy of the initial Model M_0 and the average over all unused test sets for the final Model M_3.

5.5.2 Results

After the experiments had been carried out, the system performance before and after the interactive classification training was compared. Figure 5.1a demonstrate how the classification performance for specific data sets ($T_{3...7}$) improved after the interaction by different users. These data sets have a full sequence of models (M_0 - M_3) that were created by successive users, as can be seen in the columns of Table 5.2. It should be noted that the M_0 results are available for all test sets. It can be observed that in almost all cases the results improved after the interactive labeling was performed. However, it is interesting to note that in the case of T_7 there is no improvement. The reason may be that this data set is rather different from the others due to the employed query for creating it. Additional examples provided by the users from other data sets are not very relevant to the websites included in T_7 and therefore the classifier performance remained the same. This would also explain the slight decrease in performance from M_2 to M_3 for T_3 which is the only test set was also trained on T_7 (training sequence of P_6).

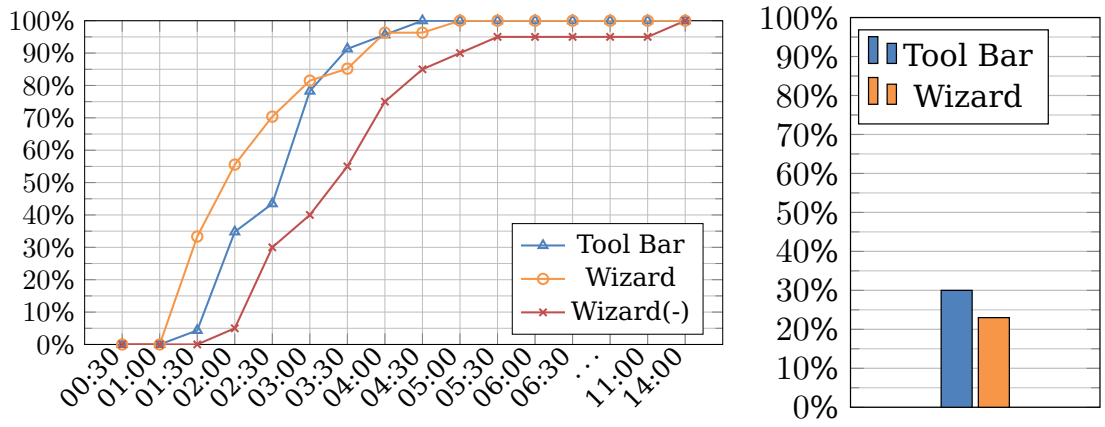
Figure 5.1b shows how the average F_1-score and accuracy increased

between M_0 and M_3. Here, M_0 was evaluated against the whole data set while each M_3 was tested using only data sets that were not involved in its creation, e.g. $T_{4...8}$ for M_3^1. On average the results improve when more interactions are taken into account. Specifically, the F_1-score is improved by around 20,7%, while the average classifier accuracy is increased by 65%.

5.6 Context-supported Query Wizard

The ontology-supported query generation framework was evaluated in a comparative, web-based user study. Two different tasks (plan a hike and get air quality information) could be solved with three alternative versions of the user interface. Each participant was given two of these six task/UI combinations in random order to avoid having learning effects in the aggregated results. Here, the selection was assured to cover both tasks and two different UI versions in each session. Besides the UI presented in Section 4.2, a previous version, as well as a version without error highlighting were used for comparison. The older version did not highlight errors dynamically but employed hard-coded error checks prior to sending the user query to the system. It was not organized as a wizard dialog but as selection of input widgets that could be accessed from a tool bar. During the interaction with the system, the time and correctness of results were measured and questionnaires on the user satisfaction and UI preference were presented after each task. Overall, 56 participants completed the evaluation.

The UI version without error highlighting was rendered almost unusable by the fact that 38% of the participants aborted the task when faced with this version. The difference between discontinuation rates of the other two versions, which both contained explanations of errors, were negligible (new UI: 4%, old UI: 5,7%). Figure 5.2 indicates that the presented approach is slightly faster to use and less error prone than the tool-bar based version with hard-coded error checking. The number of wrong queries that were submitted by the users is quite high. This is also due to an informal task description that had to be mapped to the available input options. Here, any query that differed from a defined ‘ground truth’, even slightly, was considered as wrong. Seventeen participants stated that they liked the step-by-step sequence of the wizard. Consistently, most users clearly preferred the wizard-based UI of the proposed approach. A significant speed-up from the first to the second task was observed, which means that familiarizing with the scenario has a large influence on the completion time.



(a) Task Completion Times

(b) Error Rates

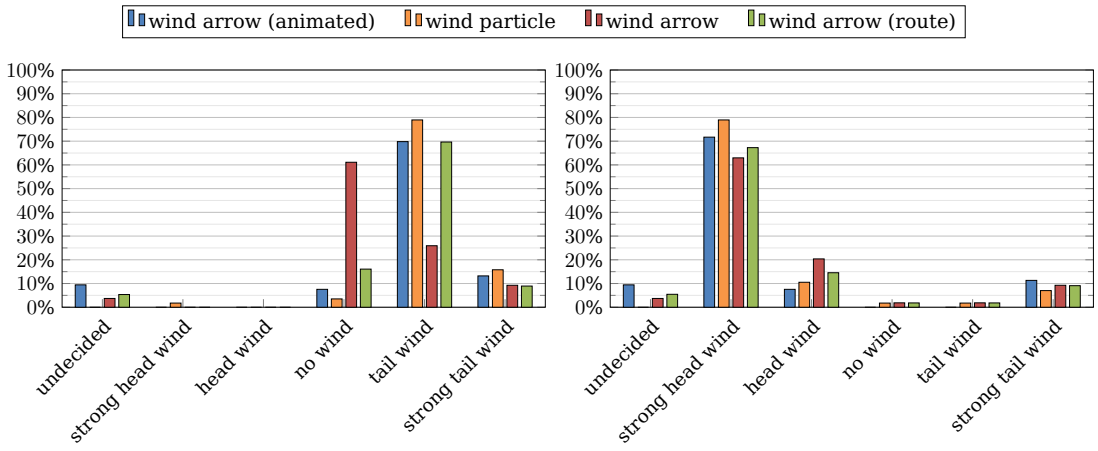
Figure 5.2 — (a) Percentage of users that have successfully submitted a correct query after the stated time. Wizard(-) is the wizard dialog without error highlighting. (b) Percentage of queries that differed from the gold label.

These results show that error highlighting is essential and that the wizard-based approach performs well. The chosen design for error highlights is useful and leads to slightly faster query creation and increased user satisfaction. This could be achieved with other measures as well, but it shows that the rules extracted from the ontology as well as the automatically created error explanations were indeed the needed information to fulfill these tasks efficiently. With further improvements on semantic technologies, this approach may become more prevalent in online decision support systems in general.

5.7 Orchestration of Environmental Visualizations

The visualization framework for environmental data was evaluated in a web-based user study, similar to the evaluation of the query generation interface. In total, 55 participants completed the study. The participants were asked to solve several information gathering tasks and to rate the systems performance after each task in short questionnaires. During the tests the correctness of the answers and the average task completion time were measured.

In a first phase the users were presented with individual visualization types in order to familiarize themselves with the framework and interpretation of the visualizations. This phase consisted of temperature heatmaps,



(a) Segment 1 (b) Segment 2

Figure 5.3 — Participant consensus on interpreted wind data based on different visualization types for two different data sets. For the first segment, most users reported that the visualization was showing moderate tail wind, for the second segment strong head wind.

air-quality line charts, and wind particles and arrows in an area layout and along a route. Here, most visualization types performed as expected. Two thirds of the participants correctly identified the temperature (within a ±3 K tolerance) using the heatmap, despite the problem of deducing values from color hues. The normalized line chart requires a training period but could then be interpreted well.[3] While some visualizations accidentally lacked important details such as a legend, they still produced consistent results. For instance, each of the different wind visualization types provided the same subjective impression of wind strength (see Figure 5.3), even when applying different ways to show the uncertainty of the data (blurring and direction arcs). An animated variation of the heatmap, showing a temperature range by fluctuating between the values, took significantly longer to interpret than its non-animated counterpart.

For the subsequent tasks, the users were shown temperature, wind and air quality data in different combinations (e.g. temperature as heatmap, wind as particle flow, and air quality as bar charts) and they were asked if the depicted situation would influence their plan for a bicycle ride within that time frame and map area. From the questionnaire answers and the correctness of the

[3] Average agreement to this statement was four on a five point Likert scale with a standard deviation of 0,75.

results, the simultaneous display of these data types could be mastered in a coherent fashion that helps the users to get a holistic picture of the situation. Again, the expected attributes of the visualization types were confirmed. Wind particle and isolines do not hinder the interpretability of other visualizations. Heatmaps and normalized line charts with colored bands conflict in their use of colors. The participants liked the additional display of actual values in a separate side panel. However, only displaying these values was not favored.

Overall, the participants found the presented methods useful (rated 4.04 of 5 with a standard deviation of 0,6) and could apply them to solve the presented tasks. Extending glyphs to areal visualization was reported to create clutter but the route based displays were highlighted for their combination of the spatial and temporal domain which makes interaction and animation unnecessary to see the data for the whole time range.

Finally, the whole PESCaDO system was evaluated by an environmental expert user panel consisting of seven participants. The majority of users deemed the provided information comprehensible and useful and they stated that they would use this kind of service. They found the interface suitable for decision support and pointed out that showing the actual data is important to them. The *particle flow* was not considered to be a good visualization for wind data by some participants. Here, it has to be noted that during the evaluation the particle flow depicted a weak northerly wind which could be easily misinterpreted as falling raindrops. Adding wind specific behavior to the particles, such as small random changes in the direction, could help in avoiding such errors.

5.8 General Considerations

In this thesis, several approaches and prototypes were presented. For each presented application area of the filter/flow selection management approach, it added analysis capabilities that allowed to perform analysis step, feedback loops, and other visual analytics functions that would otherwise be complicated or impossible to achieve using the domain specific tools alone. Certainly, each of these capabilities could have been achieved with other approaches using different interaction designs, too. Most of the related work uses similar graph-based filter tool sets or provide methods for provenance recording. However, these are either (*a*) stand-alone applications or (*b*) highly integrated into existing analysis frameworks. The former category is not adapted to a specific domain and thus often comes either with no visualization capabilities

or visual programming environments to configure generic visualizations methods such as scatter plots. The latter category, highly integrated approaches, bind the semantic of user interaction closely to the application and its domain. The provenance recording and analysis functions are thus harder to apply to other applications and represent isolated approaches that are introduced individually into each framework.

A 'plug-in' visual analytics component such as the presented filter/flow approach can utilize the visualization and selection features of its host information visualization environment. At the same time it can encapsulate functionality to be introduced into multiple environments for increasing learning effects. In addition, explicitly storing intermediate analysis products in the graph structure becomes an integrated operation. It serves the analysis process as well as provenance recording, instead of being an additional action required for producing the needed provenance information for report generation. As with many things, mitigating the conflict of interests between tool generality and customization for domain-specific needs is a balancing act. In each of the presented prototypes, minor adaptions to the domain's entities or problems have being introduced to the general approach, such as adding social media users a second entity type for the VC'11 scenario. This allowed a more intuitive usage of the component for filter creation and combination.

In general, the effort invested into an analysis has to be put in relation to the severity of the decision that it should support. If the consequences of day-to-day decisions are low, the analysis supporting them has to be efficient and available to users untrained with analysis systems. Here, the ontology-exploitation approach and the PESCaDO project in which it is embedded provide a way to facilitate the usage of domain knowledge by lay users. Systems like PESCaDO with a completely ontology-based back-end are currently rarely available to normal web users. More often, Semantic Technology is used internally for coordinating media,[4] service and supply chain management [Haller et al., 2005] and implementing agent and recommender systems [Hussein et al., 2014]. Interactive applications utilize semantic technology often only as a back-end for generic linked-open-data browsing. As the evaluation of the query generation UI has shown, other approaches for supporting the user to formulate a serviceable request exist, e.g. in the most basic way by hard-coding the required rules as has been done for the comparative user study. However, the automatic extraction of the rules provides an efficient way to exploit the domain ontology for this

[4] http://www.w3.org/2001/sw/sweo/public/UseCases/BBC/

task and it could be shown that the resulting rule set was suitable for this task. The rule evaluation during user interaction and its error highlighting can be transferred to other application areas, independently of the usage of semantic technology.

The indirection presented for visualizing georeferenced environmental data is a way to coordinate and personalize the information display of several sources. In this case, personalization can also mean the trivialization of domain-specific data such as pollutant concentrations that are not interpretable by the lay user without additional effort. The presented schema used normalization, transformation into an ordinal index, and color palettes, each potentially influenced by the user profile. The personalization and orchestration of the visualizations can be supported by machine learning approaches. Here, machine learning is not used as part of an visual analytics approach because the visualizations techniques are the data instances instead of visualizing the learning model results. Nevertheless, the respective feedback loops are introduced through supervised learning.

The evaluation studies have shown that the presented approaches have an initial barrier for being used by untrained participants, hindering their application for ‘casual analytics’. In the challenges for visual analytics, Thomas and Kielman [2009] postulate the need for ‘walk-up usable’ interfaces and an immediate use of technology without training. While being a legitimate goal, the definition of walk-up-usable is certainly depending on the common user interface designs currently in use throughout the market. Several techniques nowadays considered usable where novel concepts breaking with the user’s expectation prior to their widespread adoption, e.g. multi-touch swiping gestures for panning and zooming. Pousman et al. [2007] propose the term *casual information visualization* for “the use of computer mediated tools to depict personally meaningful information in visual ways that support everyday users in both everyday work and non-work situations”. They define common characteristics that establish the ‘casual’ nature and list edge cases of information visualization as examples: *Ambient visualizations* have at most minimal interaction capabilities and use a high abstraction to create awareness of some data source, e.g. a light bulb that changes its color based on the weather forecast for the next day. *Social visualization* depicts personal data such as photo collections, email archives, or one’s social network to create insights that are social or reflective in nature. For instance they can depict collaborative bookmark collections using tag clouds or the social network using node-link diagrams. *Artistic visualization* are data-driven depictions that may question established guidelines to provoke an emotional response through curiosity,

puzzlement or even frustration. While providing an umbrella term for such cases and less task-driven approaches for an enlarged user population, each of the examples are isolated applications with fundamentally different visual mappings and metaphors.

Interaction techniques, such as tool bars, undo histories, and multi-document environments that spread over multiple applications become conventions and benefit from learning effects that can be transferred between applications. It would be far fledged to compare the presented graph-based approach with such predominant examples, but similar to faceted browsing the general idea might be considered for appropriate tasks. Despite its name, casual analytics can also target expert users, knowledgeable in their domain and expert in their tools, but laymen to visual analytics, nevertheless. Providing a generic plug-in component facilitates the integration of such functionality in the domain-specific software solutions. On the opposite end are users with an analysis need but no expertise with the specific domains. Here, the ontology exploitation approaches are a promising way to ease the interaction. As said before, systems fully based on semantic technology are rare, but especially in the domain gaining interest from the general public, the health sector, several semantic models are already well established as can be seen by, e.g., the IDC-10 ontology.[5] Finally, the analysis and provenance effort has to be justified by the impact of the decision for which it is invested. If the analysis effort can be lowered and the provenance can be supplied semi-automatically, then lower-impact decisions can be supported by with interactive analysis processes as well, making visual analytics more ‘casual’.

[5] `http://purl.bioontology.org/ontology/ICD10`

Chapter 6

Outlook

The prototypes presented in this thesis were or could be applied to real-world tasks, but they are not of production quality. This would require a much more rigorous software development as well as eliminating usability obstacles. They have been developed for research purposes, for a heterogeneous environment of tool sets, and with a flexible infrastructure. Therefore, they will not be directly adopted by software that is currently in daily use of many people. However, I do think that similar approaches will, first, be integrated into software solutions for domain-related analysis, and will then transpire into mass market products.

Generally, visual analytics will retain strong research basis as many of the stated challenges and questions are still hardly addressed. Having started as a way to deal with the inability to explore and integrate the vast intelligence data that should have prevented the September 11 attacks, the target domains for visual analytics have shifted to other areas such as economy, social science, or digital humanities. With an increasing uptake of visual analytics methods in industry, even more research questions will surface. However, care has to be taken to avoid applying the methodology to problems that show no need for visual analysis and can be solved automatically with little effort. Otherwise, visual analytics could be misused as a way to rely on the analyst to take care of any problem that might arise. In any case, the stakeholders in each application domain have to be included into a successive transition process that will not replace existing solutions, but augment them with free

exploration capabilities and human-guided automation. A solution inspired by plug-ins can facilitate this.

With an increased prevalence of visual analytics approaches in the field, the need for suitable provenance support will rise, too. A decision process, especially in domains with severe financial or societal consequences such as patent search and crisis management, requires documentation to defend against accusations of being arbitrary. This has to include the quality of the data source and therefore the presented semi-automated reporting facility is only one part of a larger effort to record a decision's provenance. With the human tightly integrated into the analysis process, modeling cognitive processes will play an important role for researching accountability.

Using social media as a source of information is currently very popular in visual analytics research. The vast amount of data, the uncertain quality, and its unstructured content exhibit many interesting problems. Especially Twitter is used, because of its public availability and the reduced data privacy concerns due to its broadcasting nature intended for public use. Social media thus functions as a ubiquitous sensor network. Practitioners already use these sensors for opinion mining during major events as well as monitoring brand values and the impact of advertising campaigns. Observed individually, little trust can be assigned to social media messages and even aggregated information has to be used with care due to the fast spreading of rumors. However, *ScatterBlogs2* has shown that using primarily geospatial data can dissect the global chatter from local events. Still, human interpretation is essential to adjust the filters and exclude false information. Crisis management will not rely purely on this type of channel but use it as an additional source of information to obtain eyewitness reports.

Casual user decision support systems based on ontologies such as PESCaDO are just becoming feasible with the recent advances of semantic technology. Thus, the applicability of the presented approach for rule extraction has not yet been tested in other domains due to the lack of comparable systems. However, personalizing the end users experience is an established goal of usability, also listed in the standard for dialog design (EN ISO 9241-10). While the standard focused on the explicit personalization of the interaction through settings, content providers implicitly personalize the content based on the users' interaction histories. The personalization of visualization is an evident further step. Especially in the environmental and health related domain, personal sensitivities are important, and these domains experience increased attention by the casual user, as can be seen by the broad availability of activity trackers and health-monitoring wearables.

Bibliography

F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, and K. Tao. Twitcident: fighting fire with information from social web streams. In *Proc. 21st Int'l Conf. Companion on World Wide Web (WWW '12 Companion)*, pages 305–308. ACM New York, 2012. (Cited on page 60.)

C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: an implementation and evaluation. In *Proc. SIGCHI Conf. Human factors in computing systems (CHI '92)*, pages 619–626. ACM New York, 1992. (Cited on page 26.)

G. Andrienko, N. Andrienko, H. Bosch, T. Ertl, G. Fuchs, P. Jankowski, and D. Thom. Thematic patterns in georeferenced tweets through space-time visual analytics. *Computing in Science & Engineering*, 15(3):72–82, 2013. (Cited on page 4.)

D. Archambault, T. Munzner, and D. Auber. GrouseFlocks: Steerable exploration of graph hierarchy space. *IEEE Trans. Vis. Comput. Graphics*, 14(4): 900–913, 2008. (Cited on page 18.)

R. Arias-Hernandez, L. Kaastra, T. Green, and B. Fisher. Pair Analytics: Capturing reasoning processes in collaborative visual analytics. In *Proc. 44th Hawaii Int'l Conf. System Sciences (HICSS '11)*, pages 1–10, Jan 2011. (Cited on page 126.)

M. Armentano, D. Godoy, and A. Amandi. Personal assistants: Direct manipulation vs. mixed initiative interfaces. *Int'l J. Human-Computer Studies*, 64 (1):27–35, Jan. 2006. (Cited on page 102.)

P. Baudisch and R. Rosenholtz. Halo: a technique for visualizing off-screen objects. In *Proc. SIGCHI Conf. Human factors in computing systems*, pages 481–488. ACM, 2003. (Cited on page 18.)

R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29 (2):127–142, 1987. (Cited on page 17.)

R. Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957. (Cited on page 108.)

E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and Magic Lenses: The see-through interface. In *Proc 20th Ann. Conf Computer*

Graphics and Interactive Techniques (SIGGRAPH 93), pages 73–80. ACM New York, 1993. (Cited on page 18.)

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Machine Learning Research*, 3:993–1022, 2003. (Cited on page 75.)

H. Bosch, J. Heinrich, C. Müller, B. Höferlin, G. Reina, M. Höferlin, M. Wörner, and S. Koch. Innovative filtering techniques and customized analytic tools. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST '09)*, pages 269–270, 2009. (Cited on pages 4 and 82.)

H. Bosch, D. Thom, and T. Ertl. Das Web als personalisierte Entscheidungsplattform – Die PESCaDO Idee. In *Lecture Notes in Informatics (LNI) – Proc. Informatik 2011: Informatik schafft Communities*, volume P-192, page 256, 2011a. (Cited on page 4.)

H. Bosch, D. Thom, M. Wörner, S. Koch, E. Puttmann, D. Jackle, and T. Ertl. ScatterBlogs: Geo-spatial document analysis. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST '11)*, pages 309–310. IEEE Computer Society, 2011b. (Cited on pages 4 and 29.)

H. Bosch, D. Thom, G.-A. Heinze, S. Wokusch, and T. Ertl. Dynamic ontology supported user interface for personalized decision support. In *Proc. 5th Int'l Conf. Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2012)*, pages 101–107. IARIA, 2012. (Cited on page 4.)

H. Bosch, D. Thom, F. Heimerl, E. Püttmann, S. Koch, R. Krüger, M. Wörner, and T. Ertl. ScatterBlogs2: Real-time monitoring of microblog messages through user-guided filtering. *IEEE Trans. Vis. Comput. Graphics*, 19(12): 2022–2031, 2013. (Cited on page 4.)

N. Bouayad-Agha, G. Casamayor, S. Mille, M. Rospocher, H. Saggion, L. Serafini, and L. Wanner. From Ontology to NL: Generation of multilingual user-oriented environmental reports. In G. Bouma, A. Ittoo, E. Métais, and H. Wortmann, editors, *Natural Language Processing and Information Systems*, volume 7337 of *Lecture Notes in Computer Science*, pages 216–221. Springer Berlin / Heidelberg, 2012. (Cited on page 107.)

N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu. Whisper: Tracing the spatiotemporal process of information diffusion in real time. *IEEE Trans. Vis. Comput. Graphics*, 18(12):2649–2658, 2012. (Cited on page 60.)

S. Card, J. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1999. (Cited on pages 15 and 16.)

T. Catarci, M. F. Costabile, S. Levialdi, and C. Batini. Visual query systems for databases: A survey. *J. Visual Languages & Computing*, 8(2):215–260, 1997. (Cited on page 27.)

J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST '12)*, pages 143–152. IEEE Computer Society, 2012. (Cited on pages 4 and 75.)

S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *ACM Sigmod record*, 26(1):65–74, 1997. (Cited on page 20.)

C. Chew and G. Eysenbach. Pandemics in the age of Twitter: Content analysis of tweets during the 2009 H1N1 outbreak. *PLoS One*, 5(11), 2010. (Cited on page 60.)

E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proc. IEEE Symp. Information Vizualization (INFOVIS 2000)*, pages 69–75. IEEE Computer Society, 2000. (Cited on page 16.)

A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, Jan. 2008. (Cited on page 17.)

J. Codina, E. Pianta, S. Vrochidis, and S. Papadopoulos. Integration of semantic, metadata and image search engines with a text search engine for patent retrieval. In *Proc. WS on Semantic Search (SemSearch 2008) at the 5th European Semantic Web Conf. (ESWC '08)*, pages 14–28. CEUR-WS.org, June 2008. (Cited on page 40.)

D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proc. 15th Ann. Int'l ACM SIGIR Conf. Research and development in information retrieval*, pages 318–329. ACM, 1992. (Cited on page 13.)

M. Delest, T. Munzner, D. Auber, and J.-P. Domenger. Exploring InfoVis publication history with Tulip. In M. O. Ward and T. Munzner, editors, *Proc.*

10th IEEE Symp. Information Visualization (INFOVIS '04), pages r10–r10. IEEE Computer Society, 2004. (Cited on page 18.)

M. Delest, A. Don, and J. Benois-Pineau. DAG-based visual interfaces for navigation in indexed video content. *Multimedia Tools and Applications*, 31 (1):51–72, 2006. (Cited on page 18.)

S. Delisle and B. Moulin. User interfaces and help systems: from helplessness to intelligent assistance. *Artificial Intelligence Review*, 18(2):117–157, Oct. 2002. (Cited on page 101.)

W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou. LeadLine: Interactive visual analysis of text data through event identification and exploration. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST '12)*, pages 93–102. IEEE Computer Society, 2012. (Cited on page 60.)

D. C. Dryer. Wizards, guides, and beyond: rational and empirical methods for selecting optimal intelligent user interface agents. In *Proc. 2nd Int'l Conf. Intelligent User Interfaces (IUI '97)*, pages 265–268. ACM New York, 1997. (Cited on page 101.)

DUDEN. *DUDEN Das Herkunftswörterbuch: Etymologie der deutschen Sprache*. Bibliographisches Institut & F.A. Brockhaus AG, Mannheim, 3rd edition, 2001. (Cited on page 38.)

N. Elmqvist, J. Stasko, and P. Tsigas. DataMeadow: A visual canvas for analysis of large-scale multivariate data. *Information Visualization*, 7(1):18–33, 2008. (Cited on page 27.)

M. Frank, M. Muslea, J. Oh, S. Minton, and C. Knoblock. An intelligent user interface for mixed-initiative multi-source travel planning. In *Proc. 6th Int'l Conf. Intelligent User Interfaces (IUI '01)*, pages 85–86. ACM New York, 2001. (Cited on page 102.)

G. W. Furnas. Generalized fisheye views. In *Proc. SIGCHI Conf. Human factors in computing systems (CHI '86)*, pages 16–23, New York, 1986. ACM Press. (Cited on page 18.)

J. Gantz and D. Reinsel. Extracting value from chaos. White Paper, IDC iView, 2011. [Online]. Available: `https://www.acadiacorporation.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf`. (Cited on page 1.)

M. Giereth. *An Architecture for Visual Patent Analysis*. PhD thesis, Universität Stuttgart, 2012. (Cited on page 43.)

M. Giereth, H. Bosch, and T. Ertl. A 3d treemap approach for analyzing the classificatory distribution in patent portfolios. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST '08)*, pages 189–190, oct. 2008a. (Cited on pages 3 and 42.)

M. Giereth, S. Koch, H. Bosch, and T. Ertl. Visual patent retrieval. In *Internationales Rechtsinformatik Symposion (IRIS'08)*, pages 569–574, 2008b. (Cited on page 3.)

M. Giereth, M. Wörner, H. Bosch, P. Baier, and T. Ertl. Utilization of semantic annotations in interactive user interfaces for large documents. In H.-G. Hegering, A. Lehmann, H. J. Ohlbach, and C. Scheideler, editors, *GI Jahrestagung (2)*, volume 134 of *LNI*, pages 706–711. GI, 2008c. (Cited on page 5.)

D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, 2009. (Cited on pages 10 and 11.)

S. Greenberg and B. Buxton. Usability evaluation considered harmful (some of the time). In *Proc. SIGCHI Conf. Human factors in computing systems (CHI '08)*, pages 111–120. ACM New York, 2008. (Cited on page 125.)

G. Grinstein, C. Plaisant, J. Scholtz, and M. Whiting. Visual Analytics Benchmark Repository: VAST Challenge 2009 MC 2, 2009. [Online]. Available: `http://hcil2.cs.umd.edu/newvarepository/VAST%20Challenge%202009/challenges/MC2%20-%20Social%20Network%20and%20Geospatial/`. (Cited on page 83.)

G. Grinstein, K. Cook, P. Havig, K. Liggett, B. Nebesh, M. Whiting, K. Whitley, and S. Konecni. Visual Analytics Benchmark Repository: VAST Challenge 2011 MC 3, 2011. [Online]. Available: `http://hcil2.cs.umd.edu/newvarepository/VAST%20Challenge%202011/challenges/MC1%20-%20Characterization%20of%20an%20Epidemic%20Spread/`. (Cited on pages 29 and 30.)

F. Haag, S. Lohmann, and T. Ertl. Evaluating the readability of extended filter/flow graphs. In *Proc. 2013 Graphics Interface Conference*, volume 2013,

pages 33–36, Toronto, Canada, 2013. Canadian Information Processing Society. (Cited on page 19.)

F. Haag, S. Lohmann, S. Bold, and T. Ertl. Visual sparql querying based on extended filter/flow graphs. In *Proc. 12th Int'l Working Conf. Advanced Visual Interfaces (AVI '14)*, volume 2014, pages 305–312. ACM New York, 2014. (Cited on page 19.)

A. Haller, E. Cimpian, A. Mocan, E. Oren, and C. Bussler. Wsmx-a semantic service-oriented architecture. In *Proc. IEEE Int'l Conf. Web Services (ICWS 2005)*, pages 321–328. IEEE, 2005. (Cited on page 142.)

T. Hansaki, B. Shizuki, K. Misue, and J. Tanaka. FindFlow: visual interface for information search based on intermediate results. In *Proc. 2006 Asia-Pacific Symp. Information Visualisation (APVIS)*, volume 60, pages 147–152. Australian Computer Society, Inc., 2006. (Cited on page 19.)

P. Heim. *Interaktive Angleichung als Modell für die Mensch-Computer-Interaktion im Semantic Web*. PhD thesis, Universität Stuttgart, 2012. (Cited on page 90.)

P. Heim, J. Ziegler, and S. Lohmann. gFacet: A browser for the web of data. In *Proc. Int'l WS Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW'08)*, volume 417, pages 49–58. Citeseer, 2008. (Cited on page 20.)

F. Heimerl, S. Koch, H. Bosch, and T. Ertl. Visual classifier training for text document retrieval. *IEEE Trans. Vis. Comput. Graphics*, 18(12):2839–2848, December 2012. (Cited on pages 4, 71, 94, 96, and 135.)

T. Hengl and N. Toomanian. Maps are not what they seem: representing uncertainty in soil-property maps. In M. Caetano and M. Painho, editors, *Proc. 7th Inter. Symp. Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, pages 805–813, 2006. (Cited on page 113.)

T. Heverin and L. Zach. Microblogging for crisis communication: Examination of Twitter use in response to a 2009 violent crisis in seattle-tacoma, washington area. In *Proc. 7th Int'l ISCRAM Conf.*, 2010. (Cited on page 60.)

P. Hitzler, M. Krötzsch, S. Rudolph, and Y. Sure. *Semantic Web: Grundlagen*. Springer Berlin / Heidelberg, 2008. (Cited on page 20.)

D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. Vis. Comput. Graphics*, 12(5):741–748, Sept.-Oct. 2006. (Cited on page 42.)

A. Hughes and L. Palen. Twitter adoption and use in mass convergence and emergency events. *Int'l J. Emergency Management*, 6(3):248–260, 2009. (Cited on page 60.)

T. Hussein, T. Linder, W. Gaulke, and J. Ziegler. Hybreed: A software framework for developing context-aware hybrid recommender systems. *User Modeling and User-Adapted Interaction*, 24(1-2):121–174, 2014. (Cited on page 142.)

Impact Forecasting. November 2012 global catastrophe recap. Aon Benfield Annual Global Climate and Catastrophe Reports, 2012. [Online]. Available: `http://thoughtleadership.aonbenfield.com/Documents/201212_if_monthly_cat_recap_november.pdf`. (Cited on page 79.)

D. Jäckle, H. Bosch, D. Thom, R. Krüger, D. Keim, and T. Ertl. Visual analysis of social media data in emergency situations by aggregating annotated user movements. In T. Comes, F. Fiedrich, S. Fortier, J. Geldermann, and T. Müller, editors, *Proc. 10th Int'l Conf. Information Systems for Crisis Response and Management (ISCRAM)*, volume 2013. ISCRAM, 2013. (Cited on page 5.)

B. J. Jansena, D. L. Bootha, and A. Spinkb. Determining the informational, navigational, and transactional intent of web queries. *Information Processing & Management*, 44(3):1251 – 1266, 2008. (Cited on page 11.)

W. Javed and N. Elmqvist. ExPlates: spatializing interactive analysis to scaffold visual exploration. *Computer Graphics Forum*, 32(3pt4):441–450, 2013. (Cited on pages 16 and 19.)

S. Jones, S. McInnes, and M. Staveley. A graphical user interface for boolean query specification. *Int'l J. Digital Libraries*, 2(2):207–223, 1999. (Cited on page 27.)

M. Judex and J. Zisgen. Nutzung von Volunteered Geographic Information (VGI) und moderner Technologien zur Verbesserung des Lagebildes. Web 2.0 und Social Media in Katastrophenschutz und Hochwassermanagement in Heidelberg, 2013. [Online]. Available: `http://kats20.leiner-wolff.de/vortraege-3/`. (Cited on page 58.)

D. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In S. J. Simoff, M. H. Böhlen, and A. Mazeika, editors, *Visual Data Mining*, volume 4404 of *Lecture Notes in Computer Science*, pages 76–90. Springer Berlin / Heidelberg, 2008. (Cited on page 16.)

D. A. Keim. Information visualization and visual data mining. *IEEE Trans. Vis. Comput. Graphics*, 7(1):100–107, 2002. (Cited on pages 7 and 14.)

S. Koch. *Visual Search and Analysis of Documents in the Intellectual Property Domain*. PhD thesis, Universität Stuttgart, 2012. (Cited on pages 15, 43, 57, and 125.)

S. Koch and H. Bosch. From static textual display of patents to graphical interactions. In M. Lupu, K. Mayer, J. Tait, A. J. Trippe, and W. B. Croft, editors, *Current Challenges in Patent Information Retrieval*, volume 29 of *The Kluwer Int'l Series on Information Retrieval*, pages 217–235. Springer Berlin / Heidelberg, 2011. (Cited on page 3.)

S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative integration of visual insights during patent search and analysis. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST '09)*, pages 203–210, Oct. 2009. (Cited on page 4.)

S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative integration of visual insights during scalable patent search and analysis. *IEEE Trans. Vis. Comput. Graphics*, 17(5):557–569, 2011. (Cited on pages 4 and 40.)

R. Krüger, H. Bosch, S. Koch, C. Müller, G. Reina, D. Thom, and T. Ertl. HIVEBEAT - a highly interactive visualization environment for broad-scale exploratory analysis and tracing. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST '12)*, pages 277–278, 2012a. (Cited on page 5.)

R. Krüger, S. Lohmann, D. Thom, H. Bosch, and T. Ertl. Using social media content in the visual analysis of movement data. In *Proc. 2nd WS Interactive Visual Text Analytics at VisWeek*, 2012b. (Cited on page 5.)

R. Krüger, H. Bosch, D. Thom, E. Püttmann, Q. Han, S. Koch, F. Heimerl, and T. Ertl. Prolix - visual prediction analysis for box office success. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST '13)*, 2013a. (Cited on page 5.)

R. Krüger, D. Thom, M. Wörner, H. Bosch, and T. Ertl. TrajectoryLenses - a set-based filtering and exploration technique for long-term trajectory data. In *Computer Graphics Forum (Proc. Eurographics Conf. Visualization)*, 2013b. (Cited on page 5.)

R. Z. Kun Wu, Hai Jin and Q. Zhang. A vertical search engine based on visual and textual features. In *Proc. Edutainment '10, Changchum, China, August 16-18, 2010*, pages 476–485. Springer-Verlag, 2010. (Cited on page 94.)

C.-S. Lee, Y.-C. Chang, and M.-H. Wang. Ontological recommendation multi-agent for tainan city travel. *Expert Systems with Applications*, 36(3):6740–6753, Apr. 2009. (Cited on page 102.)

R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22:1–55, 1932. (Cited on page 131.)

Z. Liu, B. Jiang, and J. Heer. imMens: Real-time visual querying of big data. *Computer Graphics Forum*, 32(3pt4):421–430, 2013. (Cited on page 17.)

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. J. C. H. Watkins. Text classification using string kernels. *J. Machine Learning Research*, 2: 419–444, 2002. (Cited on page 72.)

S. Lohmann, J. Ziegler, and L. Tetzlaff. Comparison of tag cloud layouts: Task-related performance and visual exploration. In T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. Prates, and M. Winckler, editors, *Proc. 12th Int'l Conf. Human-Computer Interaction (INTERACT '09)*, volume 5726 of *Lecture Notes in Computer Science*, pages 392–404. Springer Berlin / Heidelberg, 2009. (Cited on pages 43 and 75.)

W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, July 1987. (Cited on page 115.)

X. Luo and Z. Pan. Isoline plotting method of discrete geophysical and geochemical data. In *Proc. 2nd Int'l Symp. Knowledge Acquisition and Modeling (KAM '09)*, volume 1, pages 414–417. IEEE Computer Society, 12 2009. (Cited on page 115.)

H. P. Luong, S. Gauch, and Q. Wang. Ontology-based focused crawling. In *Proc. Int'l Conf. on Information, Process, and Knowledge Management*, pages 123–128, 2009. (Cited on page 94.)

A. MacEachren, A. Jaiswal, A. Robinson, S. Pezanowski, A. Savelyev, P. Mitra, X. Zhang, and J. Blanford. Senseplace2: Geotwitter analytics support for situational awareness. In *Proc. IEEE Conf. Visual Analytics Science and Technology (VAST '11)*, pages 181–190, oct. 2011. (Cited on page 60.)

A. M. MacEachren. Visualizing uncertain information. *Cartographic Perspective*, 1992(13):10–19, 1992. (Cited on page 113.)

J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, 1986. (Cited on page 114.)

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, 2008. (Cited on page 12.)

A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proc. SIGCHI Conf. Human factors in computing systems (CHI '11)*, pages 227–236. ACM New York, 2011. (Cited on page 60.)

M. Mendoza, B. Poblete, and C. Castillo. Twitter under crisis: Can we trust what we RT? In *Proc. 1st WS Social Media Analytics*, pages 71–79. ACM, 2010. (Cited on page 60.)

K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *J. Visual Languages & Computing*, 6(2):183–210, Jun 1995. (Cited on page 42.)

J. Moßgraber and M. Rospocher. Ontology management in a service-oriented architecture: Architecture of a knowledge base access service. In A. Hameurlain, A. M. Tjoa, and R. Wagner, editors, *DEXA Workshops*, pages 289–293. IEEE Computer Society, 2012. (Cited on page 91.)

B. Motik, R. Shearer, and I. Horrocks. Hypertableau reasoning for description logics. *J. Artif. Intell. Res. (JAIR)*, 36:165–228, 2009. (Cited on page 91.)

A. Moumtzidou, S. Vrochidis, E. Chatzilari, and I. Kompatsiaris. Discovery of environmental resources based on heatmap recognition. In *Proc. 20th IEEE Int'l Conf. Image Processing (ICIP '13)*, pages 1486–1490, Sept 2013. (Cited on page 94.)

A. Moumtzidou, V. Epitropou, S. Vrochidis, K. Karatzas, S. Voth, A. Bassoukos, J. Moßgraber, A. Karppinen, J. Kukkonen, and I. Kompatsiaris. A model for environmental data extraction from multimedia and its evaluation against

various chemical weather forecasting datasets. *Ecological Informatics*, 23 (0):69 – 82, 2014, Special Issue on Multimedia in Ecology and Environment. (Cited on page 94.)

T. Munzner. A nested model for visualization design and validation. *IEEE Trans. Vis. Comput. Graphics*, 15(6):921–928, 2009. (Cited on page 125.)

T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans. Graph.*, 22(3):453–462, July 2003. (Cited on page 18.)

I. Nonaka and H. Takeuchi. *The Knowledge-creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, 1995. (Cited on page 52.)

C. Olston and J. D. Mackinlay. Visualizing data with bounded uncertainty. In *Proc. IEEE Symp. Information Visualization (INFOVIS '02)*. IEEE Computer Society, 2002. (Cited on page 113.)

S. Oyama, T. Kokubo, T. Ishida, and T. Yamada. Keyword spices: A new method for building domain-specific web search engines. In *Proc. 17th Int'l Joint Conf. Artificial Intelligence (IJCAI'01)*, pages 1457–1463, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. (Cited on page 94.)

A. Panagiotidis, H. Bosch, S. Koch, and T. Ertl. EdgeAnalyzer: Exploratory analysis through advanced edge interaction. In *Proc. 44th Hawaii Int'l Conf. System Sciences (HICSS '11)*, volume 44, pages 1–10 pages. IEEE Computer Society, 2011. (Cited on page 5.)

E. Pariser. *The Filter Bubble: What The Internet Is Hiding From You*. The Penguin Press, 2011. (Cited on page 2.)

F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Trans. Vis. Comput. Graphics*, 14(3):564–575, 2008. (Cited on page 97.)

E. Pianta and S. Tonelli. KX: A flexible system for keyphrase extraction. In *Proc. of SemEval 2010*, 2010. (Cited on pages 94 and 95.)

P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc.*

Int'l Conf. Intelligence Analysis, pages 2–4, 2005. (Cited on pages 8, 9, 11, and 23.)

C. Plaisant, J.-D. Fekete, and G. Grinstein. Promoting insight-based evaluation of visualizations: From contest to benchmark repository. *IEEE Trans. Vis. Comput. Graphics*, 14(1):120–134, jan.-feb. 2008. (Cited on page 125.)

Z. Pousman, J. Stasko, and M. Mateas. Casual information visualization: Depictions of data in everyday life. *IEEE Trans. Vis. Comput. Graphics*, 13 (6):1145–1152, 2007. (Cited on page 143.)

R. Raskin and M. Pan. Semantic web for earth and environmental terminology (SWEET). In *Proc. WS Semantic Web Technologies for Searching and Retrieving Scientific Data*, 2003. (Cited on page 99.)

J. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proc. 5th Int'l Conf. Coordinated and Multiple Views in Exploratory Visualization, 2007. (CMV '07)*, pages 61–71, July 2007. (Cited on page 16.)

M. Rospocher. PESCaDO ontology documentation. Technical Report, Fundacione Bruno Kessler, 2010. [Online]. Available: `http://www.pescado-project.eu/Pages/Pdfs-pages/PESCaDO_Ontology_Documentation_2.0.pdf`. (Cited on page 99.)

M. Rospocher. An ontology for personalized environmental decision support. In *Proc. 8th Int'l Conf. Formal Ontology in Information Systems (FOIS 2014)*, volume 267 of *Frontiers in Artificial Intelligence and Applications*, pages 421–426. IOS Press, 2014. (Cited on page 99.)

M. Rospocher and L. Serafini. An ontological framework for decision support. In H. Takeda, Y. Qu, R. Mizoguchi, and Y. Kitamura, editors, *JIST*, volume 7774 of *Lecture Notes in Computer Science*, pages 239–254. Springer, 2012. (Cited on page 91.)

T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. 19th Int'l Conf. World wide web (WWW '10)*, pages 851–860. ACM New York, 2010. (Cited on page 60.)

P. Saraiya, C. North, V. Lam, and K. Duca. An insight-based longitudinal study of visual analytics. *IEEE Trans. Vis. Comput. Graphics*, 12(6):1511–1522, 2006. (Cited on page 124.)

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002. (Cited on page 72.)

J. Scholtz, M. A. Whiting, C. Plaisant, and G. Grinstein. A reflection on seven years of the VAST challenge. In *Proc. 2012 WS Beyond Time and Errors - Novel Evaluation Methods for Visualization (BELIV '12)*, pages 13:1–13:8. ACM New York, 2012. (Cited on pages 29 and 125.)

T. Selker. COACH: a teaching agent that learns. *Communications of the ACM*, 37(7):92–99, July 1994. (Cited on page 101.)

B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012. (Cited on page 70.)

R. Shearer, B. Motik, and I. Horrocks. HermiT: A highly-efficient owl reasoner. In C. Dolbear, A. Ruttenberg, and U. Sattler, editors, *OWLED*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008. (Cited on page 91.)

B. Shneiderman. The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology*, 1(3):237–256, 1982. (Cited on page 26.)

B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992. (Cited on page 42.)

B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. Visual Languages*, pages 336–343, Sep 1996. (Cited on page 14.)

B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In *Proc. IEEE Symp. Information Visualization (INFOVIS '01)*, pages 73–78. IEEE Computer Society, 2001. (Cited on page 42.)

A. Spoerri. Infocrystal: A visual tool for information retrieval. In *Proc. IEEE Conf. Visualization (Visualization'93)*, pages 150–157, 1993. (Cited on page 27.)

M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *Proc. KDD Workshop on Text Mining*, 2000. (Cited on page 97.)

M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg. Context-preserving visual links. *IEEE Trans. Vis. Comput. Graphics*, 17(12):2249–2258, Dec 2011. (Cited on page 106.)

J. Stocq and J. Vanderdonckt. WOLD: a mixed-initiative wizard for producing multi-platform user interfaces. In *Proc. 9th Int'l Conf. Intelligent User Interfaces*, pages 331–333. ACM New York, 2004. (Cited on page 101.)

R. S. Sutton and A. G. Barto. *Introduction to reinforcement learning*. MIT Press, 1998, online version: `http://webdocs.cs.ualberta.ca/~sutton/book/ebook/the-book.html`. (Cited on page 108.)

D. Thom, H. Bosch, and T. Ertl. Inverse document density: A smooth measure for location-dependent term irregularities. In *Proc. 24th Int'l Conf. Computational Linguistics (COLING 2012)*, pages 2603–2618, 2012a. (Cited on pages 4, 63, and 74.)

D. Thom, H. Bosch, S. Koch, M. Wörner, and T. Ertl. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. In *Proc. IEEE Pacific Visualization Symp. (PacificVis)*, pages 41–48. IEEE Computer Society, 2012b. (Cited on pages 4, 34, and 74.)

D. Thom, H. Bosch, R. Krüger, and T. Ertl. Using large scale aggregated knowledge for social media location discovery. In *Proc. 47th Hawaii Int'l Conf. System Sciences (HICSS)*, volume 47, pages 1464–1473, 2014. (Cited on page 4.)

J. Thomas and J. Kielman. Challenges for visual analytics. *Information Visualization*, 8(4):309–314, 2009. (Cited on pages 8 and 143.)

J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005. (Cited on page 8.)

M. Tobiasz, P. Isenberg, and S. Carpendale. Lark: Coordinating co-located collaboration with information visualization. *IEEE Trans. Vis. Comput. Graphics*, 15(6):1065–1072, nov.-dec. 2009. (Cited on page 16.)

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Machine Learning Research*, 2:45–66, 2002. (Cited on page 70.)

E. R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, USA, 2nd edition, 2001. (Cited on page 113.)

V. Vapnik. *Statistical learning theory*. Wiley, 1998. (Cited on page 69.)

E. M. Voorhees. The philosophy of information retrieval evaluation. In *Revised Papers from the 2nd WS Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems (CLEF '01)*, pages 355–370. Springer London, 2002. (Cited on page 125.)

S. Vrochidis, H. Bosch, A. Moumtzidou, F. Heimerl, T. Ertl, and Y. Kompatsiaris. An environmental search engine based on interactive visual classification. In *Proc. 1st ACM Int'l WS Multimedia Analysis for Ecological Data (MAED '12)*, pages 49–52. ACM New York, 2012. (Cited on pages 4 and 94.)

L. Wanner, R. Baeza-Yates, S. Brügmann, J. Codina, B. Diallo, E. Escorsa, M. Giereth, Y. Kompatsiaris, S. Papadopoulos, E. Pianta, G. Piella, I. Puhlmann, G. Rao, M. Rotard, P. Schoester, L. Serafini, and V. Zervaki. Towards content-oriented patent document processing. *World Patent Information*, 30(1):21–33, 2007. (Cited on page 40.)

L. Wanner, H. Bosch, N. Bouayad-Agha, U. Bügel, G. Casamayor, T. Ertl, A. Karppinen, Y. Kompatsiaris, T. Koskentalo, S. Mille, J. Moßgraber, A. Moumtzidou, M. Myllynen, E. Pianta, M. Rospocher, H. Saggion, L. Serafini, V. Tarvainen, S. Tonelli, T. Usländer, and S. Vrochidis. Service-based infrastructure for user-oriented environmental information delivery. In *Proc. 2010 ENVIP WS at EnviroInfo2010*, 2010. (Cited on pages 4 and 91.)

L. Wanner, S. Vrochidis, S. Tonelli, J. Moßgraber, H. Bosch, A. Karppinen, M. Myllynen, M. Rospocher, N. Bouayad-Agha, U. Bügel, G. Casamayor, T. Ertl, I. Kompatsiaris, T. Koskentalo, S. Mille, A. Moumtzidou, E. Pianta, H. Saggion, L. Serafini, and V. Tarvainen. Building an environmental information system for personalized content delivery. In *Proc. 9th IFIP WG 5.11 Int'l Symposium on Environmental Software Systems - Frameworks of eEnvironment (ISESS 2011), Brno, Czech Republic, June 27-29, 2011*, volume 359 of *IFIP Advances in Information and Communication Technology*, pages 169–176. Springer, 2011. (Cited on pages 4, 91, and 110.)

L. Wanner, M. Rospocher, S. Vrochidis, H. Bosch, N. Bouayad-Agha, U. Bügel, G. Casamayor, T. Ertl, D. Hilbring, A. Karppinen, Y. Kompatsiaris, T. Koskentalo, S. Mille, J. Moßgraber, A. Moumtzidou, M. Myllynen, E. Pianta, H. Saggion, L. Serafini, V. Tarvainen, and S. Tonelli. Personalized environmental

service configuration and delivery orchestration: The PESCaDO demonstrator. In *Proc. Extended Semantic Web Conference*. Springer, 2012a. (Cited on pages 4 and 91.)

L. Wanner, S. Vrochidis, M. Rospocher, J. Moßgraber, H. Bosch, A. Karppinen, M. Myllynen, S. Tonelli, N. Bouayad-Agha, U. Bügel, G. Casamayor, T. Ertl, D. Hilbring, K. Karatzas, Y. Kompatsiaris, T. Koskentalo, S. Mille, A. Moumtzidou, E. Pianta, H. Saggion, L. Serafini, and V. Tarvainen. Personalized environmental service orchestration for quality life improvement. In *Proc. 3rd Intelligent Systems for Quality of Life information Services Workshop*, volume 382, pages 351–360. Springer Berlin / Heidelberg, 2012b. (Cited on pages 4 and 91.)

L. Wanner, H. Bosch, S. Vrochidis, N. Bouayad-Agha, G. Casamayor, L. Johansson, A. Karppinen, A. Moumtzidou, I. Kompatsiaris, and T. Ertl. Involving the expert in the delivery of environmental information from the web. In B. Page, A. G. Fleischer, J. Göbel, and V. Wohlgemuth, editors, *EnviroInfo*, Berichte aus der Umweltinformatik, pages 561–568. Shaker, 2013. (Cited on page 4.)

C. Weaver. Building highly-coordinated visualizations in improvise. In *Proc. IEEE Symp. Information Visualization (INFOVIS '04)*, pages 159–166. IEEE Computer Society, 2004. (Cited on page 16.)

WIPO. World intellectual property indicators - 2013 edition - highlights. Technical report, World Intellectual Property Organization, 2013. (Cited on page 39.)

N. Wirth. *Systematic Programming: An Introduction*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1973. (Cited on page 41.)

C. Wittenbrink, A. Pang, and S. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Trans. Vis. Comput. Graphics*, 2(3):266–279, Sept. 1996. (Cited on pages 113 and 117.)

P. C. Wong and J. Thomas. Visual analytics. *IEEE Computer Graphics and Applications*, 24(5):20–21, sept.-oct. 2004. (Cited on page 7.)

M. Wooldridge. Intelligent agents: The key concepts. In *Proc. 9th ECCAI-ACAI/EASSS 2001, AEMAS 2001, HoloMAS 2001 on Multi-Agent-Systems and Applications II-Selected Revised Papers*, pages 3–43. Springer London, 2002. (Cited on page 102.)

World Health Organization. *The international statistical classification of diseases and health related problems, ICD-10*, volume 1. World Health Organization, 2010 ed. edition, 2012. (Cited on page 99.)

M. Wörner. *Visual Analytics for Production and Transportation Systems*. PhD thesis, Universität Stuttgart, 2014. (Cited on page 82.)

M. Wörner and T. Ertl. SmoothScroll: A multi-scale, multi-layer slider. *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, 274:142–154, 2013. (Cited on page 75.)

S.-Y. Yang. Developing of an ontological interface agent with template-based linguistic processing technique for faq services. *Expert Systems with Applications*, 36(2):4049–4060, Mar. 2009. (Cited on page 102.)

D. Young and B. Shneiderman. A graphical filter/flow representation of boolean queries: A prototype implementation and evaluation. *J. American Society of Information Science*, 44(6):327–339, 1993. (Cited on pages 18, 26, and 47.)

W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Proc. 33rd European Conf. Advances in Information Retrieval (ECIR'11)*, pages 338–349. Springer Berlin / Heidelberg, 2011. (Cited on page 60.)