FAKULTÄT FÜR !NFORMATIK

Linking Science Together: How Networking Can Support Research – a Peer-to-Peer Approach

Diplomarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Informatik

eingereicht von

Andreas Ammer
Matrikelnummer 9825497

an der:
Fakultät für Informatik der Technischen Universität Wien

Betreuung:
Betreuer: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Dipl.-Inf. Raphael Fuchs

Wien, 26. 10. 2008

(Unterschrift Verfasser/in) (Unterschrift Betreuer/in)

Linking Science Together: How Networking Can Support Research – a Peer-to-Peer Approach

Masterthesis

in

Computer Science

carried out by

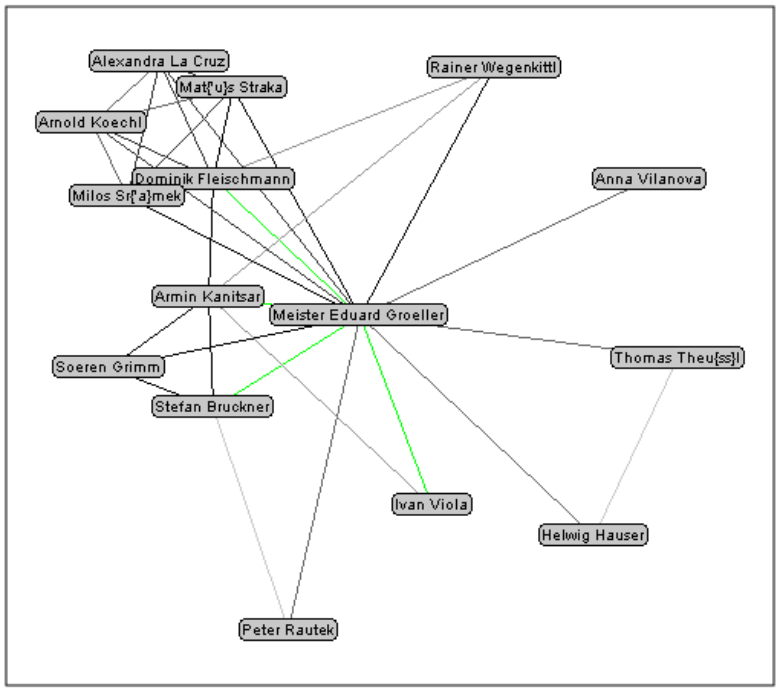
Andreas Ammer

at

VRVIS and Institute of Computer Graphics and Algorithms

Supervisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Assist: Dipl.-Inf. Raphael Fuchs



Abstract

Since no development or invention just appears out of nowhere, the search for former and related work is an important part of the whole research process. During this part of the research – by finding, filtering and connecting the knowledge found in publications of other researchers – insight in a research area is acquired. This insight, if represented and stored in an appropriate way, can be seen as new generated knowledge and could be reused by other researchers as well.

This work therefore proposes a system of interlinked nodes, in which nodes, attributes, links and groupings can be shared over a peer-to-peer network. This allows researchers to store and share their publications, as well as the knowledge they have gained during their research by attributing, linking and grouping the publications useful to other researchers in a similar area.

LinkVis, the system I propose, uses methods from the field of visualization and peer-to-peer networking to provide a platform, which is not only capable of visualizing and sharing this research knowledge, but also any other data, which can be modeled as a network of interlinked nodes.

Zusammenfassung

Da Erfindungen und Entwicklungen nicht einfach aus dem Nichts entstehen, ist das Suchen nach früherer und verwandter Arbeit ein wichtiger Teil des gesamten Forschungsprozesses. Während dieses Teils des Forschens – durch Finden, Filtern und Verknüpfen des Wissens, das in Publikationen anderer Forscher gefunden werden kann – kann ein Einblick in ein Forschungsgebiet gewonnen werden. Dieser Einblick und die in diesem Schritt gewonnenen Einsichten, können, wenn sie in geeigneter Weise dargestellt und aufbewahrt werden, als neues Wissen angesehen und von anderen Forschern weiterverwendet werden.

Diese Arbeit schlägt hierfür ein System verknüpfter Knoten vor, in dem Knoten, Attribute, Verknüpfungen und Gruppierungen über ein Peer-to-Peer-Netzwerk gegenseitig ausgetauscht werden können. Dies ermöglicht Forscher sowohl ihre Publikationen, als auch das Wissen, das sie während der Recherche gewonnen haben – in Form von Attributen, Verknüpfungen und Gruppierungen der Publikationen – zu speichern und anderen zur Verfügung zu stellen.

LinkVis, das von mir vorgeschlagene System, verwendet Methoden aus den Bereichen Visualisierung und Peer-to-Peer-Netzwerke um eine Plattform zur Verfügung zu stellen, die nicht nur Visualisierung und Austausch dieses Wissens ermöglicht, sondern auch für jede andere Datenbasis, die als Netzwerk verknüpfter Knoten modelierbar ist, verwendet werden kann.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In the first section of this chapter I will describe the motivation, which lead to the development of this work. The second section introduces LinkVis, a tool for knowledge sharing, with its features of visualization and networking.

1.1 It's a Small World or The Importance of Collaboration

Since the famous small world experiment conducted 1967 by Stanley Milgram in which he examined the average path length of social networks in the United States of America, and which comes up with the surprising result that any two people are separated by only six people in average, networks and especially social networks gained a lot of interest.

An example, where this small world phenomenon is illustrated is the so-called Erdös-number. Paul Erdös (1913-1996) was one of the most important mathematicans of the last century, whose work consists of approximately 1500 publications with about 500 coauthors. The Erdös-number calculates as followed: Erdös has an Erdös-number of 0, if someone has a common publication with Erdös he has an Erdös-number of 1, if someone has collaborated with a co-author of Erdös, but not with Erdös himself, he has an Erdös-number of 2, and so on. If someone has neither published with Erdös nor with any co-author with a finite Erdös-number, he has an infinite Erdös-number. According to the Erdös Number Project [9] the huge majority of all authors (270000 in their data set), who have a finite Erdös-number, have a number lower than eight. Again the median is about six. This "six-degrees-of-separation", as it is called, seems to be an inherent attribute of such small world networks. A small part of the Erdös-number graph can be seen in Figure 1.1.

Inherently networks with small world properties have a good routing performance (since the average path length between any two nodes is relatively low) and are therefore also quite useful for peer-to-peer networks [53].

Another interesting aspect the Erdös Number Project enlights, is that the collaboration in mathematical research increased significantly over the last decades. While at the middle of the last century most publications were by a single author, today more than half of all publications are collaborations [48]. There is no doubt, that this is also valid for other research areas beside mathematics.

Therefore I conclude collaboration is becoming more important, and it would be very beneficial to support research collaborations with the adequate tools.

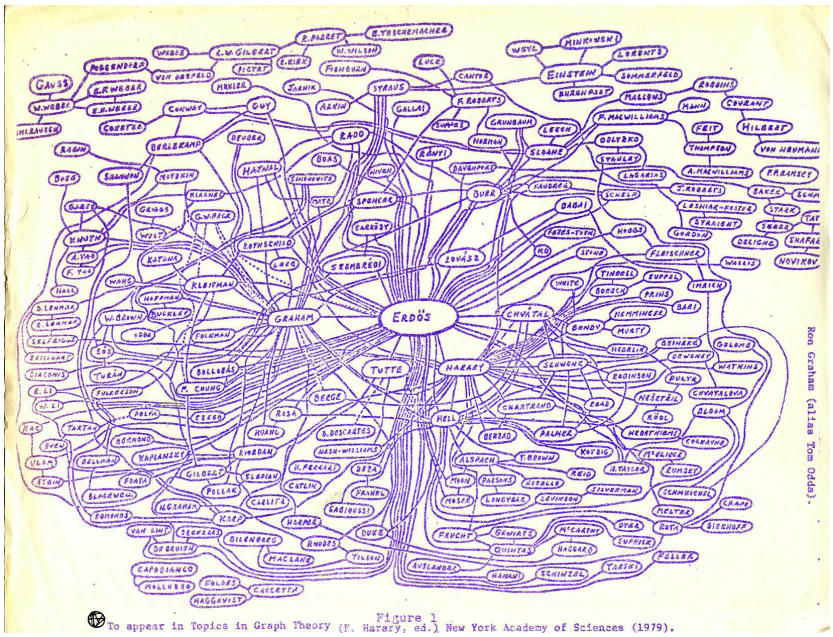


Fig. 1.1: A small part of the Erdös-number graph [9].

The system I propose aims to support collaborative research at the beginning – the scientific literature research – with such a tool, allowing to share publications and the knowledge gained during the literature research process.

1.2 LinkVis – a Tool for Sharing Knowledge

Scientific literature research is usually the basis for further research and insight. If the researcher is provided with appropriate tools to keep track of the insight he gained during this first step, this so generated knowledge may be quite useful to other researchers as well. By annotating, connecting and grouping the publications, which were useful or insightful, in a way that the "mental map" of the researcher is preserved, this knowledge can be shared.

These "mental maps" are easiest built with appropriate pictures, since the eye is the most prominent sense of the average human – hence visualization. To share these maps it makes sense to avoid any centralization for a couple of reasons. First, knowledge is per se distributed among mankind, second, something like censoring is much more difficult to appear at decentralized systems and third, it is not necessary to provide the system with one or more central server(s), what makes the system more stable and much less cost expensive. Therefore a peer-to-peer approach is the most natural approach for sharing these "mental maps".

LinkVis allows researchers to sort their publication pools (the articles they

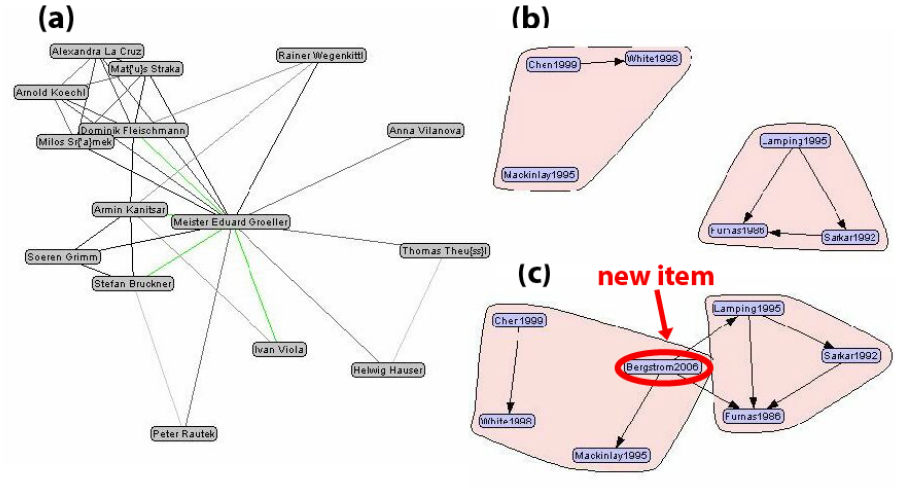


Fig. 1.2: (a) linking between authors, (b) grouping of articles, (c) a new paper is added by someone and the user is informed.

have read in the process of scientific literature research) by attributing, linking and grouping articles in a way, that seems useful to them, and furthermore share these attributes, links and groups among each other. For other researchers in this area this information may be very profitable, reducing the time necessary to identify important publications or finding less obvious connections.

LinkVis may also be used as a collaborative tool between researchers, in that way that it offers a common publication pool, which can be annotated, linked and grouped by the collaborators.

Furthermore it is possible to use LinkVis for any other data beside of scientific literature, where its features of visualizing and sharing may be useful, as long as the data can be modeled as a network of interlinked nodes.

2. STATE OF THE ART OF SCIENTIFIC LITERATURE RESEARCH TOOLS

This chapter discusses related work and gives an overview of the current state of the art considering scientific literature research. The first section starts with the data all scientific literature research tools have to deal with – the data and metadata concerning publications. The second section describes the developments concerning scientific publication in the World Wide Web. After that a review of current peer-to-peer technologies is given, with a section dedicated to bibliographic peer-to-peer applications. The last section deals with visualization, again with a focus on the bibliographic area.

2.1 Considering the Data of Scientific Documents and Bibliographic Descriptions

The first part briefly describes important aspects of data and metadata concerning scientific documents. In the second section I will discuss the relationships between them.

2.1.1 Data and Metadata

Today a community agreed standard declares the format in which scientific articles are distributed: this format being the pdf format[1], postscript or less common HTML. While both dominating formats have their strengths in layouting and especially printing (so called electronic paper[2]), they also bear a disadvantage in full text analysis. Since Postscript and in less extent pdf are actually full programming languages for page layouting, the extraction of the actual text content can be rather difficult (i.e., the letters of a single word could be set at different positions of the file).

For the metadata describing the publications popular standards include mainly BibTeX(from the LaTeXuniversum) [61] and proprietary formats (i.e., Endnote [8]). For an example see Figure 2.1.

Offering a separate metadata file reduces the disadvantages of pdf and postscript, but, due to the lack of full text analysis capabilities, the metadata often has to be generated manually. This often leads to differences between metadata describing the same paper.

[1] The pdf format is a variant of postscript and was introduced by Adobe [2].

[2] Electronic papers (also called e-prints) use the common physical paper page layout.

```
@inproceedings{ shneiderman96eyes,
    author = "Ben Shneiderman",
    title = "The Eyes Have It: {A} Task by Data Type Taxonomy for Information Visualizations",
    booktitle="IEEE Visual Languages",
    number = "UMCP-CSD CS-TR-3665",
    address = "College Park, Maryland 20742, U.S.A.",
    year = "1996",
    pages = "336-343",
    url = "citeseer.ist.psu.edu/shneiderman96eyes.html" }

@article{shneiderman:eht,
  title={{The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations}},
  author={Shneiderman, B.}
}
```

Fig. 2.1: Two differing metadata files (BibTeX) describing the same data.

With the Dublin Core Metadata Element Set [7] an international standard for describing any digital resource is given. The fifteen (optional) tags include i.e., title, subject, format, language, date, identifier, creator and so on and are general enough to describe nearly every resource. With RDF (Resource Description Framework) [17] a model is at hand to process metadata. RDF uses triples of resources, attributes and values to relate specific properties to resources. Uniform Resource Identifiers (URIs) [24] can be used to eliminate ambiguities and to guarantee an unique definition of a resource, a tag or an attribute[3]. Further ontologies can be used, in which concepts of a specific domain are modelled. An ontology consists of a taxonomy – a definition of classes and relations among them – and of a set of inference rules. If different ontologies are used to describe the very same entity, a relationship can be defined how to translate between the ontologies. With these given technologies data can be interpreted and used in a meaningful way by computers. An ontology in which research communities and relevant related concepts are modelled is described by Sure et al. [82] (Semantic Web for Research Communities ontology, SWRC).

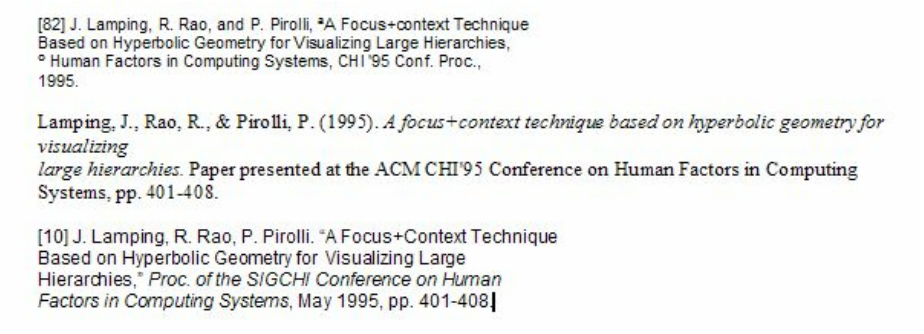
In "The Semantic Web" [25] Berners-Lee et al. describe the powerful new possibilities emerging of the new form of web content which is meaningful to computers, and emphasize the use of metadata. In the chapter dedicated to metadata in "Peer-To-Peer Harnessing the Power of Disruptive Technologies" [41] Rael Dornfest and Dan Brickley also stress the use of metadata in peer-to-peer applications.

2.1.2 Properties of Scientific Documents and the Relations Between Them

Scientific articles often refer to previous and related work. These relations, called citations or references[4] link the variety of articles together. The average article

[3] A unique definition can be achieved by including an Universally Unique Identifier (UUID) [64], which uses both time and location (i.e., the host address) when and where the resource was created as identification.

[4] In the literature there is some ambiguity in the usage of this vocabulary.


[82] J. Lamping, R. Rao, and P. Pirolli, ªA Focus+context Technique
Based on Hyperbolic Geometry for Visualizing Large Hierarchies,
° Human Factors in Computing Systems, CHI '95 Conf. Proc.,
1995.

Lamping, J., Rao, R., & Pirolli, P. (1995). *A focus+context technique based on hyperbolic geometry for visualizing*
large hierarchies. Paper presented at the ACM CHI'95 Conference on Human Factors in Computing Systems, pp. 401-408.

[10] J. Lamping, R. Rao, P. Pirolli. "A Focus+Context Technique
Based on Hyperbolic Geometry for Visualizing Large
Hierarchies," *Proc. of the SIGCHI Conference on Human*
Factors in Computing Systems, May 1995, pp. 401-408

Fig. 2.2: Different citations referring to the same article.

cites about 10 papers [23], but unsurprisingly the actual number of citations varies greatly from paper to paper. While the great majority of papers cite less then 20 - 30 articles, papers with a citation count of up to 300 can also be found. This is easily explained when the nature of the paper is considered; overview articles which offer a solid introduction to a whole research domain are naturally referencing a greater number of papers than high focused ones.

Kim and Whitehead Jr. [54] describe some other interesting properties of these references. They propose a method to find related papers by using the hyperlinked citation relationship between papers. Outgoing from an input paper the algorithm follows the references of the paper, so building a "convex hull" of the source paper reference network. An interesting aspect of this method is that the algorithm doesn't terminate even on a relative high iteration step, but that the number of unique references[5] converges to about 4 per each new added paper.

Citations also have the problem of inconsistency similar to the inconsistency problem with document metadata (partly caused by the inconsistent document metadata as it seems); this inconsistency complicates the automatic linkage between articles (see Figure 2.2).

To handle the problem to decide which citations correspond to the same publication Lawrence et al. [62] propose a word and phrase matching algorithm which they also use for CiteSeer [46] (for a more detailed description see next chapter). Identity uncertainty in the context of citation matching is also considered by Pasula et al. [73], who apply a heuristic approach to this problem.

Backward reference data (article B cites article A, article A is cited by article B) can also be from interest. How often an article is cited can indicate the impact a paper has on the scientific community.

Relationships between the articles concerning the specific topic or the domain can be derived by keywords and domain categories respectively. These keywords or domain categories can be provided manually as metadata or derived from full

[5] References, which did not appear before.

text analysis.

Other relationships between articles can be established by linking articles by the same author together and so on. To complete the list of link types between articles, I cite Börner et al. [27], who cite White and McCain [83]:

> "We use certain technical terms such as intercitation, interdocument, coassignment, co-classification, co-citation, and co-word. The prefix 'inter-' implies relationships between documents [or units]. The prefix 'co-' implies joint occurrences within a single document [or unit]. Thus, intercitation data for journals are counts of the times that any journal cites any other journal, as well as itself, in a matrix. (The citations appear in articles, of course.) The converse is the number of times any journal is cited by any other journal. The same sort of matrix can be formed with authors replacing journals. Interdocument similarity can be measured by counting indicators of content that two different documents have in common, such as descriptors or references to other writings (the latter is known as bibliographic coupling strength). Co-assignment means the assignment of two indexing terms to the same document by an indexer (the terms themselves might be called co-terms, co-descriptors, or co-classifications). Co-citation occurs when any two works appear in the references of a third work. The authors of the two co-cited works are co-cited authors. If the co-cited works appeared in two different journals, the latter are co-cited journals. Co-words are words that appear together in some piece of natural language, such as a title or abstract. Bother 'inter-' and 'co-' relationships are explicit and potentially countable by computer. Thus, both might yield raw data for visualization of literatures."

Summarizing, it can be concluded that relationships between publications are an interesting and important aspect, which can help in exploring, analyzing and visualizing the scientific literatures. A good understanding of these relations may be beneficial for further research.

2.2 World Wide Web

The first part of this section deals with the state of the art concerning publishing of scientific work on the World Wide Web. The problem of identifying relevant data is discussed in the second part.

2.2.1 Scientific Archives

Many scientific archives are nowadays partly or as a whole available on the World Wide Web. While many of these archives – mainly the sites of publishers and journals (i.e., ACM [1]) – offer only restricted access, i.e., by the opportunity to purchase articles or by a membership in a scientific organization, others grant

unrestricted access to them. The latter are often connected to universities or departments of universities or directly to the author. While most large archives include a functionality to search and browse in the data set, others depend on the search of generalized search engines. A widely used example for such a search engine is Google [10] which offers with `http://scholar.google.com` a specific search interface for scientific articles (but no articles themselves).

As mentioned, much scientific content is available on the World Wide Web by now, but the idea Cameron [36] describes – the idea of an universal, freely available, internet-based bibliographic and citation database which links every scholarly work ever written – is still not yet achieved.

One step in this direction is the CiteSeer project by Lawrence et al. [46, 6]. Citeseer is an autonomous citation indexing system which indexes academic literature in electronic format. E-prints, like pdf or postscript files, are acquired by web search and identified as scientific articles by heuristics (i.e., scientific articles usually contain a bibliographic section). These articles are parsed to find citations and useful data like author(s), abstract, introduction, citations, citation context and full text is extracted. This data is indexed and stored. To identify citations to the same article a normalization and matching algorithm is used [62]. To find similar articles Lawrence et al. apply three methods: word vectors, string distance and citations.

The generated index can be queried by users per web interface. CiteSeer presents the results context- and query-sensitive, so the summary contains sample context of the query terms. For each article a list of references, back references and similar articles is provided along with abstract, BibTeX-key and the full text [63].

Another scientific archive which gives open access to e-prints in physics, mathematics, computer science, quantitative biology and statistics is the arXiv-repository [3]. ArXiv is located at the Cornell University and is considered to be the most important online archive by many mathematicans.

CiteULike [5] (developed by Richard Cameron) is a free web service, which allows users to store, organize and share the scientific papers they are reading. CiteULike extracts the citation data automatically and allows the users to attribute tags to the papers. A grouping functionality is included, these groups can be shared among users. It is also possible to search for papers with a given tag.

Another interesting project is the Open Archives Initiative (OAI) presented by Lagoze and Van de Sompel [59, 14]. The OAI aims to provide an open architecture framework which allows the efficient dissemination of content in an electronic way. The core of the OAI is the Open Archives Initiative Protocol for Harvesting Metadata (OAI-PHM), which should enable both data providers and service providers to exchange metadata of scholary publications. Thereby the focus is laid on interoperability. The OAI-PHM is based on the Dublin Core Metadata Element Set, and has reached release version 2, which is in contrast to the previous version not experimental but a stable production version. An important application developed out of the OAI is the search engine OAIster [13]. Today the OAI Protocol for Harvesting Metadata is supported by a number of

archives, including i.e., the former mentioned CiteSeer.

2.2.2 Search – Similarities and Differences

A question which arises with the pure mass of available content, is how to find the most appropriate data to a specific query. Looking on the World Wide Web with its network of websites connected by hyperlinks, a similarity between the World Wide Web itself and the scientific archives with its articles connected by citations (and other references) can be noticed (in its very beginning the Web was designed for the exchange and linking of scientific articles). Kleinberg [55] addresses the problem of finding the most appropriate data to a specific query in the World Wide Web by analyzing the underlying link structure and identifying web pages as "authoritative" pages and hub pages. A web page which is linked to by many other pages, is considered as an authoritative information source, a page which links to many authoritative pages relevant for a specific topic is called a hub. For Kleinberg the web consists of a network of interlinked hubs and authoritative sites. This analogy may be used when looking at the article network. Kleinbergs algorithm was further improved by Bharat and Henzinger [26] by adding content analysis to the connectivity analysis.

Another method for measuring the importance (relevance) of web pages for a specific topic was developed by Page et al. [72]. The method called PageRank calculates rankings by distributing and propagating rank through links. The project PageRank gave birth to the search engine Google, which uses among others the PageRank method to define the relevance of a page.

The task of identifying relevant web pages in the World Wide Web is definitely sharing some similarities with the identification of relevant publications in an article network. However, how well these methods of the web may work in this area, has still to be examined.

Considering searching, there are two main differences between the World Wide Web and the article network:

- Common metadata of publications: Attributes of articles are well-known. Every publication has a title and one or several authors, usual attributes include year and journal of publication, and less common but useful attributes include topic, domain and a list of keywords. Because of the (intended) diversity of the web, the World Wide Web lacks such common descriptions. Nevertheless these attributes may be used to improve the searching process.

- Different link types: In the World Wide Web, there is just one type of connection – the hyperlink – but in the article network, there exist at least two different connection types: co-authorship and citation-linking. Generally, there could be any number of link types; this has to be considered when designing the search functionality.

The best approach here may be adapting the given search methods of the web, so that they consider and exploit the features of the article network.

2.3 Peer-to-Peer Networking

This section is giving an overview of existing peer-to-peer technologies, starting with the three classic models for peer-to-peer networking. Then the network used most today, namely BitTorrent, is described. After that the general peer-to-peer platform JXTA is introduced. Peer-to-peer networks which include semantics and bibliographic peer-to-peer applications are described in the last part of this chapter. The models used by BitTorrent, JXTA and semantic peer-to-peer networks differ from the three classic models for peer-to-peer networks, this will be examined in the respective subsections.

2.3.1 Introduction

Since Napster and Gnutella appeared at the beginning of the 21th century and started a new way of using computers, the term peer-to-peer has become one of the most discussed terms in computer science. File sharing has not only revolutionized the way of consuming media, but also added some challenges to the traditional media producers (i.e., the music and movie business). Beside of legal and economic issues, the concept of peer-to-peer is quite astonishing. In the peer-to-peer model every node of the network (called peer) acts as a server as well as a client, so home computers are uplifted from their used state of pure consumers to full participating actors in the network. The potential of these millions of home computers on the internet in both CPU cycles and memory space could be used for example for computing time expansive tasks (grid-computing like SETI@home [18]), collaborative work spaces, anonymous communications, file sharing applications and so on. Peer-to-peer therefore opens a quite large field of opportunities.

There are three common peer-to-peer models, which will be described in the next sections.

2.3.2 Centralized Directory Model

In the centralized directory model the participating peers connect to a central server and publish information about the content they offer. This information is stored in a central index, which can be queried by search requests. The central server responds to these requests with the peer information matching the query. The content delivery is done directly between two peers.

Since the central server has to manage information of all peers and all published information as well as handle all search requests, a scaling limit may occur. The example of Napster, which made this model popular, however showed that it scales rather well.

2.3.3 Flooded Requests Model

The flooded requests model is a fully decentralized, unstructured and pure peer-to-peer approach. No content information is published, instead search requests are broadcasted to each known peer. If they are unable to answer the request,

they broadcast it to their known peers. This request forwarding is usually limited by a number of hops (forwards), after which the requests are discarded. Therefore it can not be guaranteed that requests for existing data will be answered positively. Flooding also requires a great amount of network bandwidth, which leads to the bad scaling performance of this model.

The flooded requests model was introduced by Gnutella. The topology graph and the generated network traffic of the Gnutella network was evaluated in a case study by Ripeanu [75], who came to the conclusion that the future growth of the Gnutella network is necessarily depending on efficient network usage. Ritter [76] also proved mathematically that Gnutella can't scale.

Attempts to improve the scalability by adding structure to the network lead finally to the development of distributed hash tables. Another approach to improve scalability and performance without adding additional structure to the net tries to optimize the search algorithms.

Yang and Garcia-Molina [84] present therefore three techniques for efficient search in unstructured peer-to-peer networks, which are iterative deepening, directed breadth-first search and local indices.

Iterative deepening just initiates multiple breadth-first searches with successively larger depth limits. Since the peer count increases exponentially with the depth of the search, much traffic can be spared, if the query can be satisfied at a lower depth.

Directed breath-first search uses heuristics to send search requests not to all known peers, but only to nodes which had the best past search performance.

In the local indices technique each peer keeps an index over the data of each peer within some hops of itself. So by querying one peer, the data collections of multiple peers can be used.

Despite their simplicity, these three methods reduce the bandwidth cost of searches to about 40 percent of the unoptimized flooding method.

2.3.4 Document Routing Model – Distributed Hash Tables (DHT)

The document routing model is also fully decentralized, but in comparison to the flooded requests model, it is a structured one, meaning that it establishes an overlay network over the physical network. This overlayed structure allows to route requests in a directed way and therefore increases the performance and scalability of the system. The document routing model attributes documents (values) as well as peers with a key, which identify them uniquely. If a document (value) is inserted by a peer, it forwards the document to the peer, which has the most similar key to the document (value) key. This is repeated until the most similar peer is the current peer. Usually each peer on the route also keeps a local copy of the document (value).

A request is handled in a similar way: it is forwarded to the peer, which is most similar to the requested key. If the document (value) is found, the response is forwarded back to the initiator. Peers on the route usually keep a copy of the document (value) in their local cache again. Figure 2.3 and Figure 2.4 show how routing works with distributed hash tables (DHTs) in the case of the Tapestry

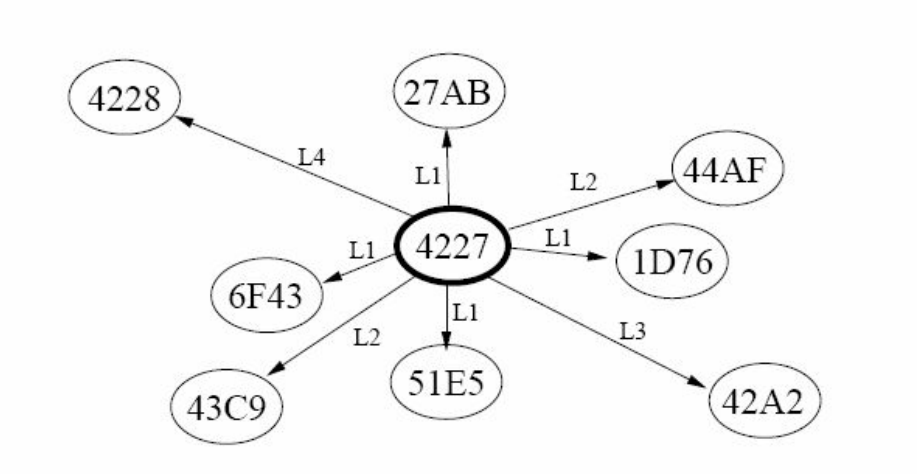


Fig. 2.3: Tapestry routing mesh from the perspective of a single node. Outgoing neighbor links point to nodes with a common matching prefix. L1-links match zero digits, L2-links match the first digit, L3-links match the first two digits and so on. Together, these links form the local routing table (Zhao et al. [85]).

example. Since the key of the requested document (value) has to be known to retrieve the document (value), the implementation of a search algorithm is much more sophisticated as in the straight forward flooded request model.

Algorithms implementing the document routing model, may differ in calculating the similarity between keys, routing table sizes, peer selection strategies, caching and other optimization methods.

Besides Freenet by Clarke et al. [40], which uses the document routing model to generate a fully anonymous publication network, and Oceanstore by Kubiatowicz et al. [58], which aims at a global-scale architecture for persistent storage, there are several more algorithms using this model.

These algorithms are quite similar in their goals, to reduce the number of peer-to-peer hops that must be taken to locate a document and to reduce the amount of routing information that must be kept at each peer.

The most common algorithms based on distributed hash tables are CAN by Ratnasamy et al. [74], Chord by Stoica et al. [81], Pastry by Rowstron and Druschel [77] and Tapestry by Zhao et al. [85].

Milojicic et al. [69] offer not only a good general introduction, overview and classification of peer-to-peer computing but also a comparison of these algorithms.

Further DHT algorithm developments include Kademlia by Maymounkov and Mazieres [68], which uses the XOR-metric as similarity distance of keys, and also introduces an optimization based on the inverse relation of node failures to node up-time and P-Grid by Aberer et al. [19, 20], which generates virtual

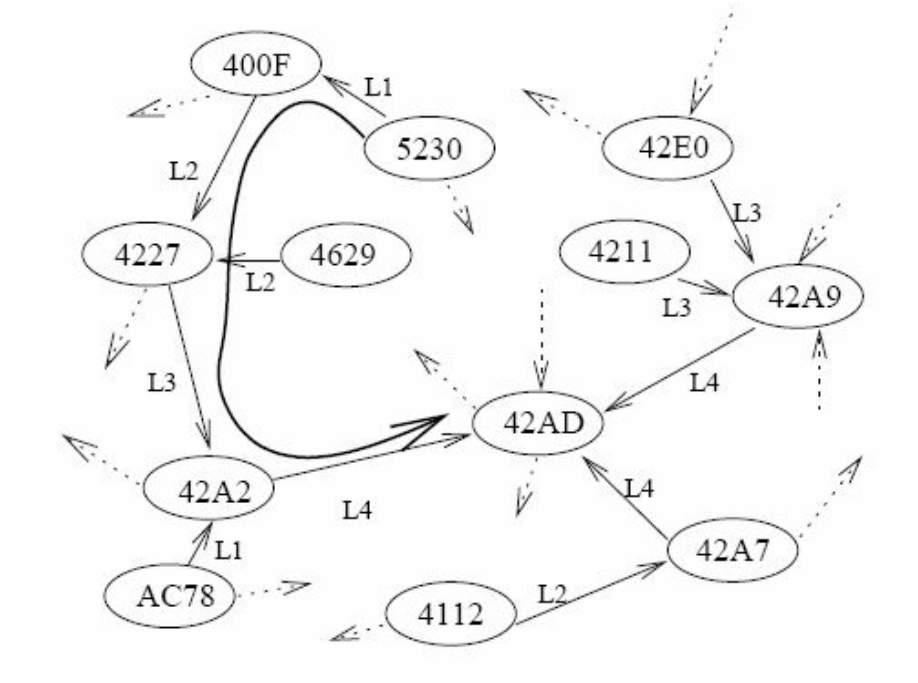


Fig. 2.4: Path of a message. The path taken by a message originating from node 5230 destined for node 42AD in a Tapestry mesh. In each step of the routing, the message is sent to the node whose prefix matches more digits (Zhao et al. [85]).

binary search trees by assigning key-interval-responsibilities to peers.

2.3.5 A Note on BitTorrent

Today BitTorrent is the most successful peer-to-peer protocol for file transfers, causing a significant part of the whole internet traffic. Yet BitTorrent focus on efficient content delivery and does not include any forms of content localization. That means the search process has to be handled outside of the peer-to-peer network, often realized by web archives using the classical client-server-model. One of the most popular ones is piratebay.org [15].

What makes BitTorrent fundamentally different to other peer-to-peer applications is that it does not build up a user network sharing content, but just generates a file transfer session for each content. These file transfer sessions, called "torrents", are completely independent of each other. Peers involved in such a file transfer session, replicate the file among each other using swarming techniques. This means that the file is split into small pieces, which are exchanged. Each peer thereby maintains a list of other peers and which parts

of the file these peers offer. To join a transfer session, a peer downloads a "torrent"-file from a web server, which contains meta information about the file as well as the ip-address of a so-called "tracker" which keeps track of all peers currently involved in the transfer session. The "tracker" hereby is the only centralized component of BitTorrent, but it is not directly involved in the actual data distribution. For more information see Legout et al. [65].

2.3.6 The JXTA Project

JXTA is an open-source peer-to-peer framework with focus on interoperability. With its core functionality and the protocols it tries to give a useful de-facto standard for peer-to-peer applications. For an overview of JXTA see Gong [47].

Its main goals are:

- Interoperability: Peers are enabled to locate each other, participate and offer services to each other across different P2P systems and different communities.
- Platform independence: JXTA is designed to be independent from programming languages, system platforms and networking platforms.
- Ubiquity: JXTA is designed to be implementable on nearly every digital device, including home computers as well as mobile phones.

JXTA thereby starts from a low level, meaning that it is not related to any existing peer-to-peer application or protocol. JXTA defines several protocols for peer discovery, peer group memberships, and establishing connections.

Technically JXTA uses a hybrid model, by introducing super peers called "rendezvous peers", which are responsible for maintaining a shared resource distributed index (SRDI). Normal or so called "edge peers" advertise their content. These advertisements are stored at their "rendezvous peer's" SRDI, which could be used for queries. Queries themselves are only propagated between "rendezvous peers", which is very efficient concerning the number of peers involved in handling a query.

Halepovic and Deters [50] review the performance in JXTA networks using benchmarking. Some aspects of JXTA will be described in more detail in Chapter 4.

2.3.7 Semantic Peer-to-Peer and Bibliographic Peer-To-Peer Projects

Another interesting development in peer-to-peer networking is including semantics, generating a new kind of peer-to-peer networks.

Edutella by Nejdl et al. [70] is an open source project, which aims to provide an RDF-based metadata infrastructure for peer-to-peer applications. Edutella thereby builds upon JXTA. The core service of Edutella is the query service, a query exchange mechanism for RDF-metadata. Peers register the queries, which they may be asked, by specifying metadata schemas or individual properties or

individual values for these properties. So each peer acts as a RDF-metadata-repository, which additionally publishes what kind of metadata it offers. Queries then are sent to the peers, which are interested in this type of query. OAI-P2P presented by Ahlborn et al. [21] reuses the concept of Edutella in the context of the Open Archives Initiative.

RDF-based peer-to-peer networks, and more general schema-based peer-to-peer networks are discussed by Nejdl et al. [71], who propose a super-peer topology as a suitable topology for schema-based peer-to-peer networks. Super-peer indices, they argue, exploit the RDF ability to uniquely identify schemas, schema attributes and ontologies, and therefore can be well used for routing between super-peers and peers as well as within the super-peer backbone network.

Löser et al. [67] propose to use such schema-based peer-to-peer systems for the exchange of scienitific documents. Scientific documents, which are annotated with small, but well-defined sets of metadata using standard taxonomies, are predestinated for this approach. Löser et al. also introduce a "Semantic Overlay Clusters" layer. That is a virtual context-specific view on selected peers, which could be chosen by content classification or query abilities.

Arumugam et al. [22] propose a ontology driven approach called "P2P Semantic Web", which has capabilities to find relevant sets of ontologies, to facilitate the reuse of existing ontologies to create additional ontologies, and to advertise and share the resulting ontologies.

A peer-to-peer architecture called "KEx" for distributed knowledge management is proposed by Bonifacio et al. [28]. In "KEx" queries can be accompanied with a "focus", which is a part of an ontology. When a peer receives a query, its matching algorithm tries to match the focus of the query semantically and syntactically. The syntactic matching process is using an indexer to search for the occurrence of specific keywords in the set of documents owned by the peer. For the semantic matching it is tried to find a correlation between a provider's context and the query focus. If the focus points to other peers, the peer will forward the query.

Haase et al. [49] also propose a peer selection model based on semantic similarity. In comparison to "KEx" the peers share a common ontology, which makes it easier to calculate similarity. Peers advertise (publish) expertise, and this advertisments are used to route queries to the peers, whose expertise is most similar to the query. The peer selection model proposed by Haase et al. is used for the application Bibster, which will be described in more detail later.

Ehrig et al. [42, 43] suggest a framework for distributed knowledge management systems called "SWAP" which also uses the concepts of RDF repositories, ontologies and semantic querying. The "SWAP" architecture is used by Bibster.

Bibster [4] by Broekstra et al. [33] is an award winning peer-to-peer system for exchanging bibliographic metadata among computer science researchers. Bibliographic entries made available for Bibster are classified according to two different ontologies: The first ontology is the SWRC ontology[6], which for short describes different generic aspects of bibliographic metadata and would be valid

[6] Semantic Web for Research Communities ontology; described in Chapter 2.1.1.

across many different research domains. The second ontology is the ACM Topic Hierarchy which describes specific categories of literature for the computer science domain. These ontologies are exploited to optimize data storage, query formulation, query routing and answer presentation.

For querying a new RDF query language was defined by Broekstra and Kampman [34], called "SeRQL", which met the key requirements in compositionality (complex queries can be composed from simpler queries), schema awareness (for semantic querying the query language has to be aware of the structure it is querying), functionality to deal with optional values (properties which may or may not be present in the data for a particular resource), functionality for navigating the class/property hierarchy and data typing.

For data storage an architecture was developed to efficiently store and query large quantities of metadata in RDF and RDF Schema. It is called "Sesame", and has been developed by Broekstra and Kampman [35]. For the routing, the former mentioned peer-selection algorithm by Haase et al. [49] is used. Bibster implements also a semantic duplicate detection algorithm, which considers the underlying structure of the resources. With these enabling techniques, Bibster offers a tool for computer science researchers to import, share and search bibliographic metadata.

The main difference between the system proposed in this work and the systems described in this section, is the focus on interactive visual exploration and user-enabled information sharing. While previous work is focussed on textual representations, this work proposes a visualization enabled framework with interactive exploration possibilities. While previous work is limited to the sharing of data and metadata, in LinkVis it is possible to share data as well as metadata, as well as user-generated meta-information or "meta-metadata", meaning data describing metadata. By adding attributes, links and groups to the metadata, the knowledge of the users can be shared, which can be used to search, explore, analyze, and visualize the article network.

2.4 Visualization and Interaction

The following section deals with visualization and interaction. Starting from general visualization and interaction concepts, this chapter describes the current state of the art from issues of graph visualization to bibliographic visualizations like domain visualization and visual interfaces to digital libraries.

2.4.1 Visualization Concepts and Graph Visualization

Concerning visualization and interaction, Ben Shneiderman offers a good starting point by describing seven tasks for information visualizations [79] which should be considered when designing an advanced graphical user interface. These tasks are overview, zoom, filter, details-on-demand, relate, history, and extract. A user would like to get at first an overview of the whole data, then when he has noticed some interesting items, he would like to be able to zoom in on them, and maybe filter out the uninteresting ones. After that he may want

to get more detailed information about them. Shneiderman summarized these tasks in his "Visual Information Seeking Mantra": Overview first, zoom and filter, then details-on-demand.

The other tasks are:

- relate – allows the user to visualize relationships between items
- history – keeps track of the previous actions, to support undo and redo
- extract – allows the user to extract a sub-collection of items in a format which allows further usage.

These tasks are useful for nearly every information visualization application, regardless of the type of data to be visualized.

Herman et al. [52] give a survey on graph visualization and navigation techniques quite useful to complement Shneidermans design principles for applications of graph drawing. Herman et al. identify three important factors concerning graphs and graph drawing algorithms. The first one is planarity, that means that a graph can be drawn without crossing edges. Few or no edge crossings seem to be significant for graphs to be comprehensible for human users, but unfortunately the minimizing edge-crossings problem is NP-complete. The second important factor is predictability, meaning that two different runs of one graph drawing algorithm involving the same or a similar graph should not lead to radically different visual representations, so the "mental map" of the user is preserved. The third factor is time complexity. Any visualization application needs to provide near real-time interaction. The graph drawing algorithm has to support this.

The key problem of graph drawing is handling large graphs. Despite the problem that large graphs are bound to be overloaded regarding the available space and therefore bringing some difficulties in perception and understanding for the users, most sophisticated graph layout algorithms are only useable for relatively small graphs. As a solution for this problem, spanning trees can be used. So the problem is reduced to calculate a spanning tree for the graph, and then the much less complex tree layouting algorithms can be used. Other approaches to deal with large graphs are using 3d instead of 2d, or using hyperbolic space [60]. Further the focus and context technique [60, 44, 78] can be used to make large graphs more insightful and easier to explore. Another method is to use clustering to simplify the graph. By combining items into one group and only representing the group as a whole, the number of visual items needed to be displayed can be reduced. Clustering can be applied structure-based or semantic-based. Finally there are incremental exploration techniques, where only a small portion of the full graph is displayed and other parts of the graph are only displayed when needed. So the problem of graph size is relaxed, because only a small subgraph has to be drawn.

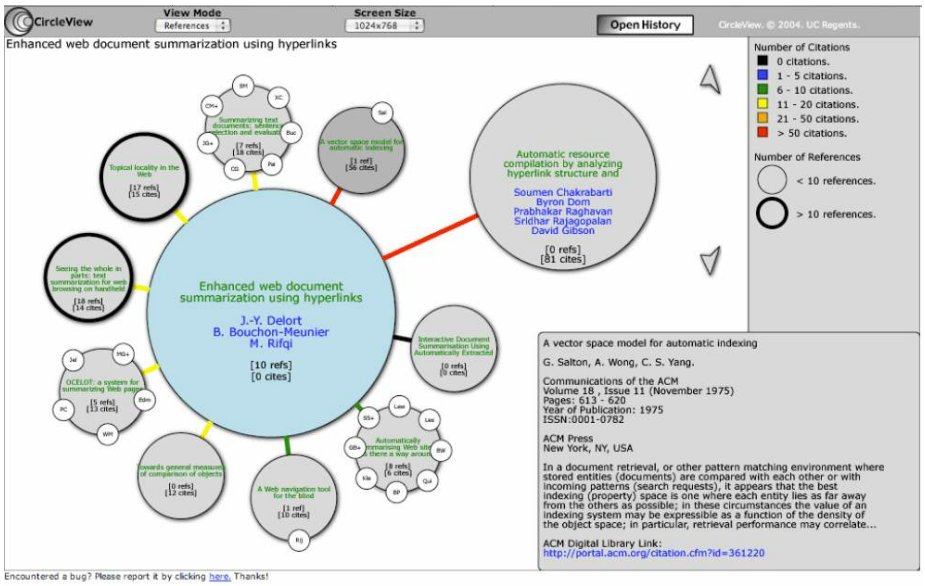


Fig. 2.5: CircleView by Bergström and Whitehead Jr. [23].

2.4.2 *Bibliographic Visualizations*

Considering now the application of visualizing scientific articles and the corresponding citation network, the question arises whether to use hierarchical tree or directed graph drawing methods. On one hand the article network does not have an inherent hierarchy, as it can not be decided on which level of hierarchy a specific paper is situated, but on the other hand when using clearly directed relations (like citations), an at least locally valid hierarchy can be established. If one looks only at articles related to one starting article, it is possible to use tree based drawing methods. If the depth of the tree increases (by recursively following the relations), the hierarchical ordering usually associated with trees does not fully apply anymore[7].

Nevertheless a tree-based approach is used by Bergström and Whitehead Jr. [23] for their project CircleView, a citation network browser that uses circles around circles as visualization method to show one focus paper and additionally two levels of its citation network. The restriction to two levels of the citation network, along with highlighting items representing the same paper at mouse-over, allows the use of the simpler tree model (see Figure 2.5).

Börner et al. [27] review visualization techniques that can be utilized to map the domain structure of scientific disciplines. With the example of a bib-

[7] When paper A is citing paper B and C, and paper B is also citing paper C, paper C would be a direct child of paper A and a direct child of paper B; this does not appear in a strict hierarchical data set.

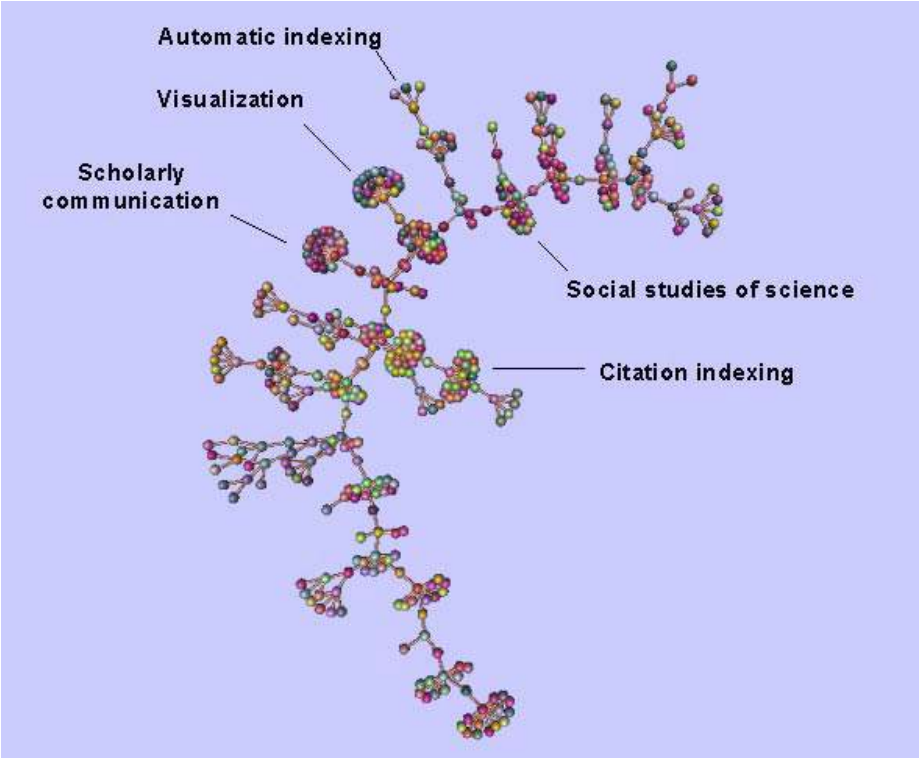


Fig. 2.6: Document Co-citation map; each dot represents a document with 10 or more citations (Boerner et al. [27]).

liographic data set – including articles from citation analysis, bibliometrics, semantics and visualization literatures – they describe and compare various visualization algorithms. Figure 2.6 shows an example of a document co-citation visualization of their data set.

Chen [37] describes the development and application of visualization techniques for users to access and explore information in a digital library (for an example see Figure 2.7). Chen et al. [39] also did an domain analysis in the field of computer graphics in the year 2001.

Boyack et al. [30, 31] use their visualization tool called VxInsight to create domain analysis for science and technology management respectively for digital libraries (see Figure 2.8).

Börner et al. [29] describe the application of collaborative visual interfaces to digital libraries that support social navigation. Börner et al. [27] identified also six steps which are generally appearing in the process of domain visualization. These steps are (1) data extraction, (2) definition of unit of analysis, (3) selection of measures, (4) calculation of a similarity between units, (5) ordination, or the

assignment of coordinates to each unit, and (6) use of the resulting visualization for analysis and interpretation (see Figure 2.9).

Finally Lin et al. [66] describe a prototype visualization system to enhance author searching based on co-citation analysis and self-organizing maps ([56, 57]) which is able to generate interactive author maps in real time.

To close this chapter the top ten list of problems in visual interfaces to digital libraries by Chen and Börner [38], will be given. In analogy to Hilbert's famous list of mathematical challenges at the beginning of the 20th century, this list tries to define a new research agenda for the future:

1. Theoretical Foundation:

   According to Chen and Börner the research of visual interfaces to digital libraries lacks solid theoretical foundations. Theoretical principles concerning visual interfaces to digital libraries would be of great use.

2. Empirical Foundation:

   Identification of common elements of successful visual interfaces to digital libraries, as well as identifying beneficial and useful features, would be of great importance.

3. Scalability

   As the volume of data to be handled increases, scalability issues arise in both data management as well as in creating visualization which could deal with large data sets.

4. Labeling

   The problem of labeling visual interfaces includes selecting meaningful labels (also selecting the items which should be labeled) and displaying readable labels.

5. Individual Differences

   Research in human-computer interaction has shown that individual differences of the users can be the most significant factor in performance. How visual interfaces should accommodate these differences is quite a question.

6. Supporting Collaborative Work

   Given the individual differences and the diversity of social norms in the internet, supporting collaborative work is a challenging task.

7. Benchmarking and Standardization of Testing

   Benchmarks and standards for testing would be helpful to push development of digital libraries forward.

8. Evaluation

   Evaluative studies are needed to find out what is working for designers as well as users.

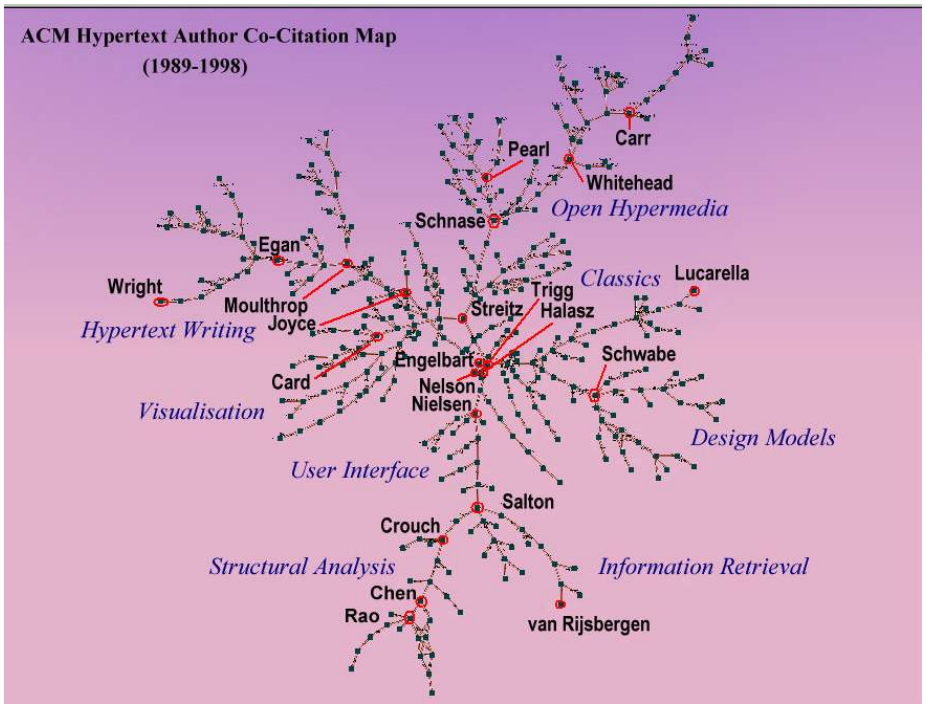


Fig. 2.7: Author co-citation map; each dot representing an author (Chen [37]).

9. Personalization

   Customized and personalized information delivery is a trend. While individual differences concern the style, this point concerns the content of the information to be displayed.

10. Standardization

    Standardization of services, representation formats and protocols, would be of great use, to avoid reinventing the same solutions over and over again. Standards of course would also improve interoperability.

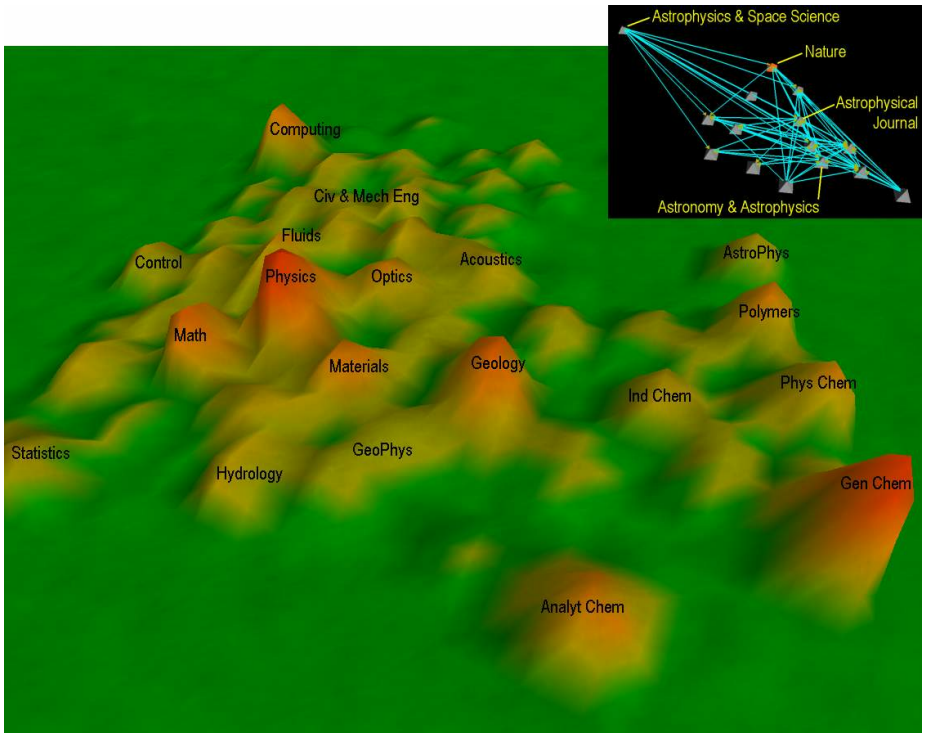


Fig. 2.8: Landscape of the physical sciences by Boyack et al. [30].

DATA EXTRACTION	UNIT OF ANALYSIS	MEASURES	LAYOUT (often one code does both similarity and ordination steps)		DISPLAY
			SIMILARITY	ORDINATION	
SEARCHES ISI INSPEC Eng Index Medline ResearchIndex Patents etc. BROADENING By citation By terms	COMMON CHOICES Journal Document Author Term	COUNTS/FREQUENCIES Attributes (e.g. terms) Author citations Co-citations By year THRESHOLDS By counts	SCALAR (unit by unit matrix) Direct citation Co-citation Combined linkage Co-word / co-term Co-classification VECTOR (unit by attribute matrix) Vector space model (words/terms) Latent Semantic Analysis (words/terms) incl. Singular Value Decomp (SVD) CORRELATION (if desired) Pearson's R on any of above	DIMENSIONALITY REDUCTION Eigenvector/ Eigenvalue solutions Factor Analysis (FA) and Principal Components Analysis (PCA) Multi-dimensional scaling (MDS) Pathfinder networks (PFNet) Self-organizing maps (SOM) includes SOM, ET-maps, etc. CLUSTER ANALYSIS SCALAR Triangulation Force-directed placement (FDP)	INTERACTION Browse Pan Zoom Filter Query Detail on demand ANALYSIS

Fig. 2.9: The steps during the process of domain visualization (Börner et al. [27]).

3. LINKVIS – THE CONCEPT

The first part of this chapter describes the concept of LinkVis from a functional perspective, explaining it from the view of the functionality the system intends to realize. Therefore the focus lies on the process of scientific literature research, which is dedicated the first two sections. Considering the requirements which come along with this research process, necessary features can be extracted, which will be described in the following sections.

The second part is designated to the technical design of LinkVis, beginning with an overview over the whole system, followed by important aspects which are described in the respective subsections. These aspects include data structure, web, peer-to-peer networking and user interface.

3.1 Functional Concept

In this section the process of scientific literature research is examined and the emerging requirements are identified. Along with some more general and well known needs, these build the list of requirements which are challenging the LinkVis design.

3.1.1 The Scientific Literature Research Process

Scientific literature research usually starts with searches for publications which matches a specific pattern. This search pattern can include keywords, topic, year of publication and so on. When publications are seen as entities with a number of given attributes (like title, topic, a list of keywords, year of publication and so on) search queries can be formulated as (attribute, value) pairs, which can be matched against the attributes of the publication entities.

Another often used search process uses the citation network to find similar publications. By following the citations or the backreferences of a starting publication, publications which are with high probability from a similar research area, can be found. To handle this kind of search, citation links can be added between the publication entities. The resulting model of publication entities connected by citation links can then be used to formulate search queries. A query hereby consists of a (key, link type) pair, where the key corresponds to the (unique) key of the starting publication, and the link type declares the type of links which should be considered. The search of publications by author can be handled similarly, by adding links between publication entities and one or several person entities.

After getting a result to the search query, the number of publications is optionally filtered by additional parameters (i.e., an appropriate number of articles citing the resulting publications). The process of reading the publications, which are classified as being of interest, is left to the user.

Another important task, which takes place after searching, finding and studying a publication, is to store it. Often it can be useful to have the original publication at hand, to look up the details, which were lost from memory after some time. The most crucial aspect hereby lies in storing the publication so that it can be easily retrieved again. Therefore a "mental map" for the user is of great help. This map can be generated by the user by grouping, linking, and annotating publications. Furthermore, such a "mental map" of the publication pool can help to get more insight in the domain itself and the connectivity in it (and between different domains).

These "mental maps" of course can be useful to other researchers in a similar area too. So by sharing these user-generated groups, links and annotations, this knowledge can be passed on to others. The scientific literature research process can be quite accelerated by this sharing.

So far, the following tasks can be identified:

- Searching and Filtering: Search methods include searching by (attribute, value) and (key, link type) pairs. Furthermore the search results may be filtered according to an additional parameter.
- Storing: Storing and retrieving metadata as well as data (the actual electronic paper) in a practical way for the user is an essential feature of the system.
- Building "mental maps": By adding arbitrary annotations, links and groups a "mental map" of the publication pool can be built.
- Sharing: To allow others to profit from these "mental maps", sharing is required.

3.1.2 Case Study: How to Create a State of the Art Report

In this section the scientific literature research process will be further examined in a case study, in which I am doing some research for writing the state of the art chapter of this thesis. Fictionally I already have LinkVis as research tool at hand. This section therefore not only examines further the process of scientific literature research, but also presents some results of the actual LinkVis system. All figures in this section were created with LinkVis by using the bibliography of this thesis.

Step 1: Exploring Since I have already collected some articles before I have heard of LinkVis, I begin with importing the folder where I store the papers. Then I import the BibTeX-file describing all papers (generated by the tool I use to manage BibTeX-entries). After the importing has finished, I visualize all

the articles and persons, which gives a rather confusing picture. Nevertheless I notice a network of authors, who seem to collaborate quite often. To examine it in more detail, I remove all visible items, despite one of the authors of that collaboration network. Then I add all items, which are linked to that node again to the visualization. I repeated this step and received the picture shown in Figure 3.1.

Since I am not interested in the articles at this stage, I choose to create a new link type, by connecting two "author of" links to one "co-author" link. The result is shown in Figure 3.2.

To get an idea of the time line of these collaborations, I include the articles again. I set the label for the articles to their BibTeX-key (which includes the year of publication), and choose to use grouping instead of links to connect the co-authors. The result can be seen in Figure 3.3.

I concluded that co-author maps can already be created directly after the importing process, since all necessary information is given in the BibTeX-files.

Step 2: Sorting To sort the publications I use the grouping functionality. By searching the articles matching a keyword, like i.e., "citation", "web", "peer" and "vis", I got useful groups. These keywords are corresponding to the topics of the sections in my state of the art chapter, and therefore I see what papers I will cite in each section. An interesting aspect I notice while interactively grouping the articles, is that there are some publications which corresponds to the keyword "citation" as well as "vis", while there are no articles with both keywords "peer" and "vis". So the combination of peer-to-peer networking with visualization methods seems to be a new topic (see Figure 3.4 and Figure 3.5).

To connect the BibTeX-metadata with the actual paper, I use "metadata-data" links between the corresponding nodes, so I can open the article directly from LinkVis.

Step 3: Using the Network After I have sorted my document collection, I choose to look if there are new articles, which I may want to use for my state of the art report. I connect to the network and let LinkVis start a search for articles, which are citing one of the articles in my collection. To get also the articles which are not connected to them directly, I start a search request for articles containing the term "vis" or "peer" in the title too.

After studying the search results, I download some articles, which seem interesting. Then I use the pdf-parser to extract the citations from the files.

The resulting picture can be seen in Figure 3.6. Then I let LinkVis add the authors to the visualization, once using links (see Figure 3.7), once using groups (see Figure 3.8).

I concluded that citation maps can only be created automatically, when there is access to the actual publication (the data file).

Step 4: Collaboration After reading the articles, I choose to add some of them to the groups I will use for my state of the art report. Since a colleague of me

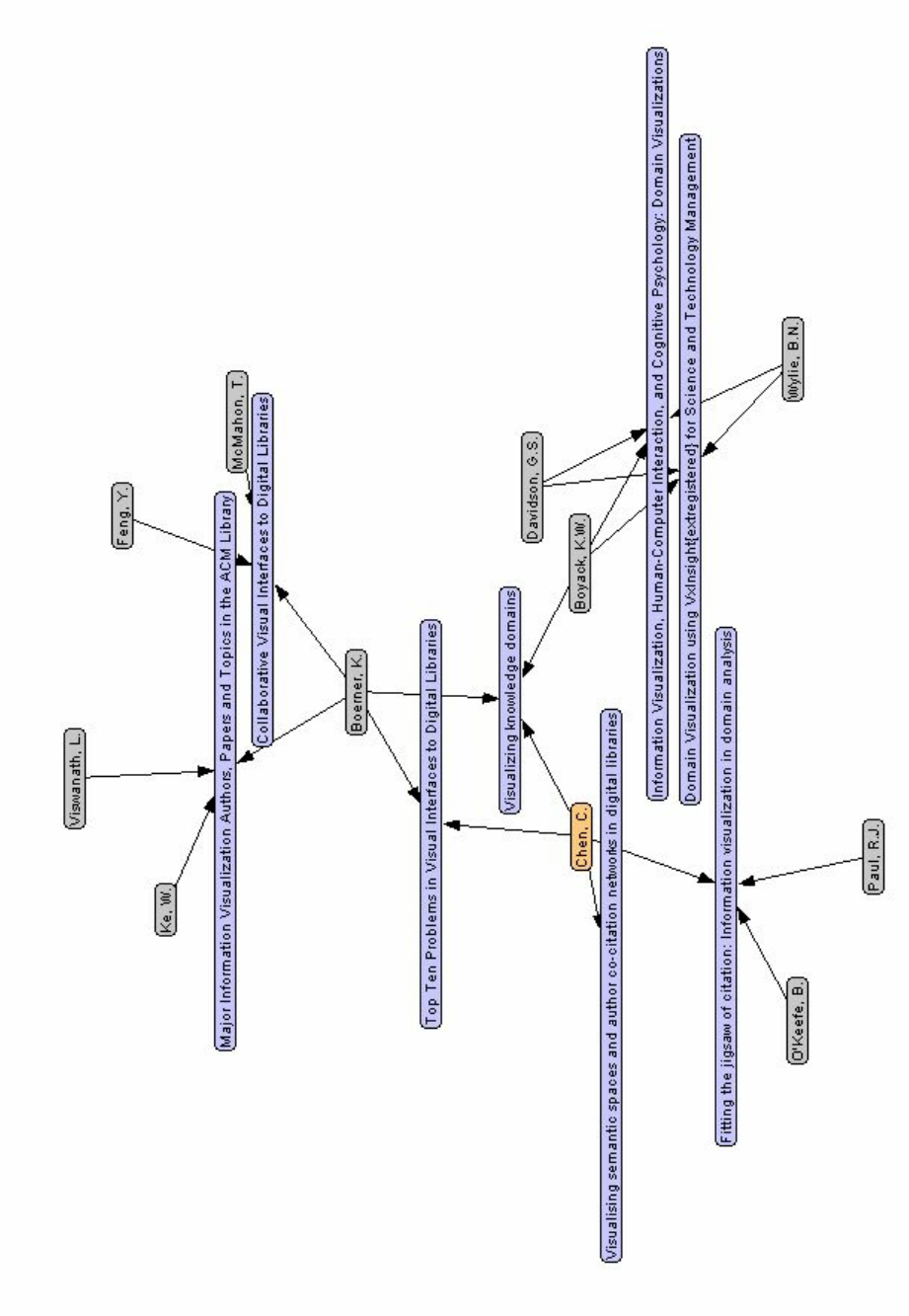


Fig. 3.1: After importing, co-author maps can be shown. In this case starting from author C. Chen all links are followed recursively to get the collaborating authors and the corresponding articles.

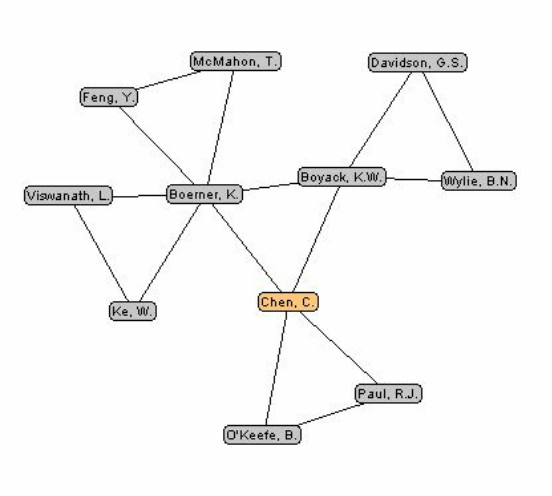


Fig. 3.2: By connecting two "author of" links, "co-author" links can be generated. This figure shows the co-author-map of Figure 3.1 without the intermediate articles.

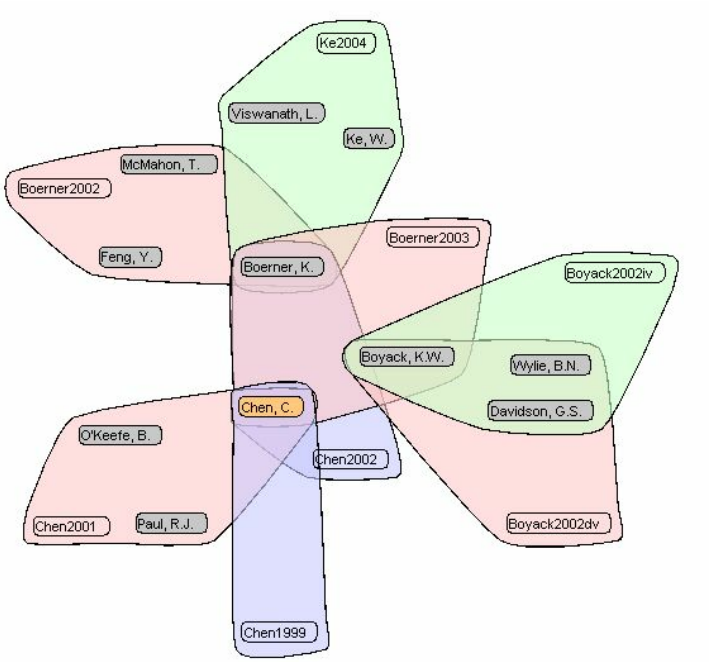


Fig. 3.3: This figure shows a co-author map with the same data as Figure 3.1 by groupings instead of links. Articles are labeled with their BibTeX-key and are shown transparent.

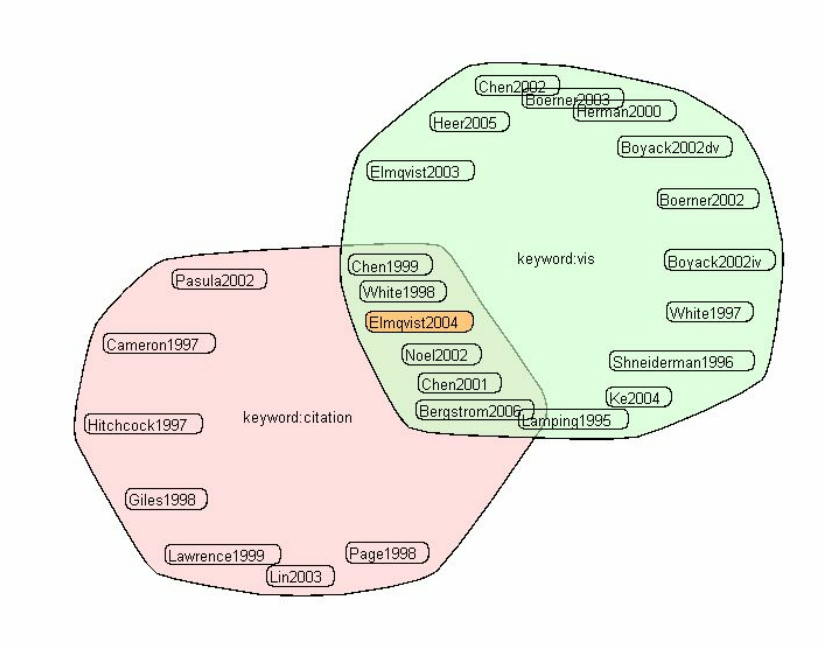


Fig. 3.4: Articles, labeled with their BibTeX-key, grouped according to keyword. The shown articles were retrieved by a search query, which matches "citation" and/or "visualization" to the title of the articles.

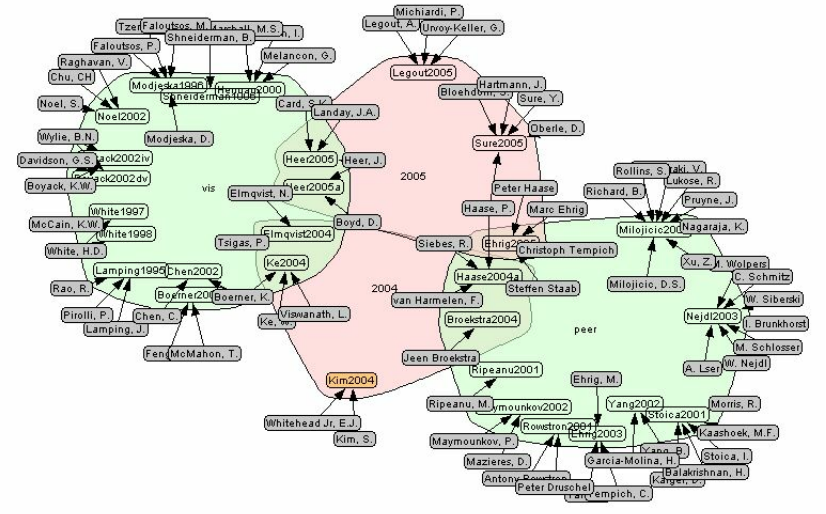


Fig. 3.5: Articles, labeled by BibTeX-key, grouped according to keyword and year of publication, with the authors linked to.

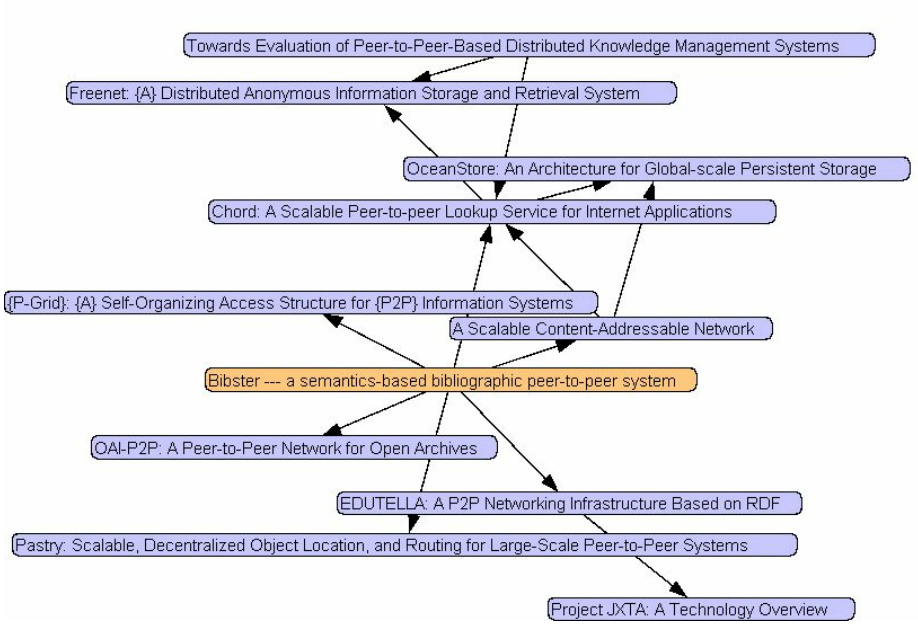


Fig. 3.6: Selected articles, connected by citation links.

is also working in a similar area, I also publish them in a new group and send him an Email, that I have found some publications he may be interested in.

After some hours, while I was working on my state of the art report, my colleague had left me a message about the new papers and that he has also found some interesting articles. I visualize the group again and examine the new papers. He also had added a link between two papers, which says that the methods described in the first article were necessary for the algorithm described in the second paper. After reading the new publications, I add comments to some of these and publish my update again.

In summary, the case study includes a lot of different tasks and features, which would be useful. Most of them have already been identified in the section before, but two important requirements can be added:

- Data Extraction: It is necessary to provide the user with useful ways of data importing and extracting, so that the user can concentrate on the task to analyze, manipulate and represent the data. The extraction methods include pdf parsing as well as the automatic generation of new links and groups based on the already existing data.
- Visual Interactive Exploration: Exploration of search results, building a "mental map", and sharing these "mental maps", all these tasks can be improved by visual interactive exploration.

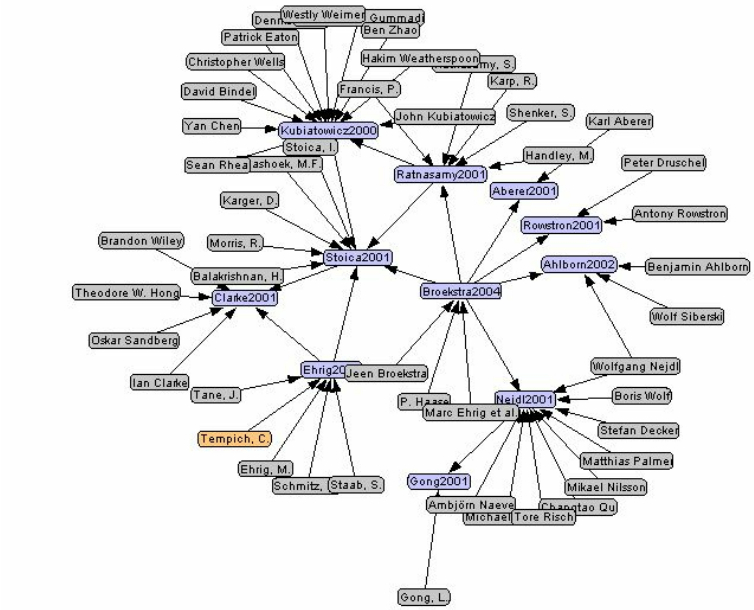


Fig. 3.7: The same set of articles as in Figure 3.6 (the label is the BibTeX-Key instead of the title), with the authors linked to.

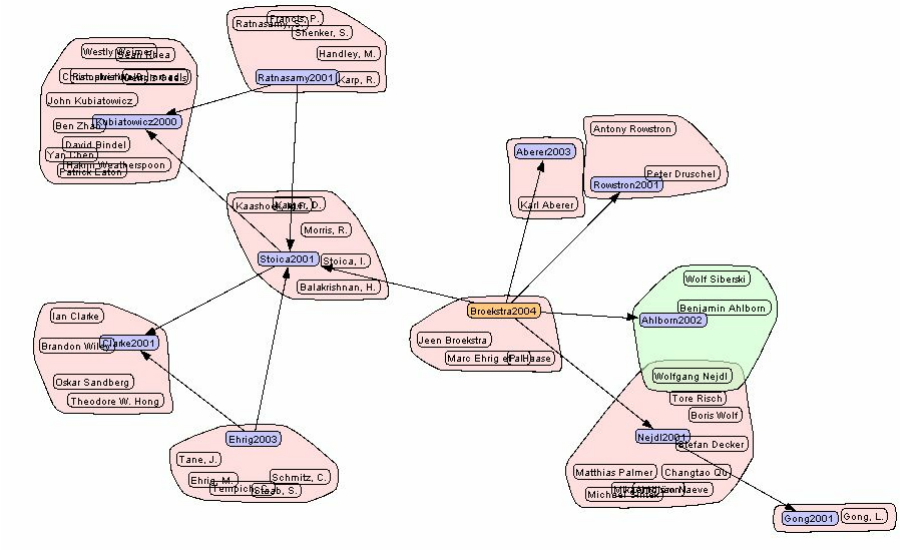


Fig. 3.8: The same data as in Figure 3.7 with groupings instead of links.

3.1.3 Resulting Functional Requirements

Now the tasks are identified, which are useful or needed for the scientific literature research process, a list of requirements can be given. This list includes also some other requirements, which are at least partially necessary for an application to get accepted by users.

Beginning with the tasks identified in the previous sections, the requirements include:

- Provide useful search methods: These methods include search functionality by group membership, (attribute, value) and (key, link type) pairs. Also an additional filtering may be applied to the search results.
- Useable data import and extraction facilities: it is necessary to provide the user with useful ways of data importing and extracting, so that he can concentrate on the task to analyze, manipulate and represent the data.
- Store and retrieve: Storing and retrieving metadata as well as data in a practical way for the user is an essential feature of the system.
- Attributing, connecting and grouping resources: The former mentioned retrieving process can be improved by building "mental maps" of the data by attributing, connecting and grouping data items. These functions also can be used for collaboration and for searching, analyzing and visualizing the data set.
- Visualisation: Offer visualizations which help to analyze the data set and allow building accurate "mental maps" of it. I will describe further aspects dealing with visualization and interaction in the following section.
- Sharing: Allow to share data and meta-information which others.

Further the following requirements can be added:

- Performance: each task should be executed in an acceptable time span. Longer lasting processes should be indicated (i.e., with a progress bar). Additionally, visualization applications should provide a near real-time interaction behavior [52].
- Intuitive user interface: The user interface should allow the user to execute tasks intuitively and quickly. In the past, powerful applications often restricted their user pool by having a too complicated user interface [80].
- No additional effort for the user: One common feature of successful peer-to-peer applications is that the sharing of resources does not require an additional effort for the user [32].

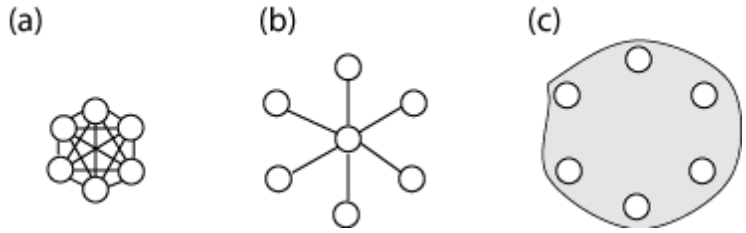


Fig. 3.9: (a) when all nodes in the group are linked together, the coupling is very dense, (b) nodes are connected by a central node representing the group, (c) an outline shows the group membership.

3.1.4 Visualization and Interaction Concept

Generally there are a lot of ways to visualize a data set consisting of interlinked nodes. Because many of these have their specific advantages, it may be useful to provide multiple of these visualizations in parallel. Nevertheless I choose to use a spring layout visualization at first hand, because it is very flexible in parameterization and also rather intuitive to understand and interact with.

A spring layout, or force directed layout, uses a simulation of physical forces to determine the placement of the nodes. Edges are usually modeled as springs, nodes as mass particles. Additionally the node particles are charged, so they are repelled by other particles (and edges). Then the physical forces are simulated, pushing and pulling the nodes to places according to the spring lengths between them. This usually leads to a rather aesthetic layout, with few edge crossings and overlappings.

The graph view of LinkVis uses such a spring layout by modeling resources as nodes, and links as edges. Groups are modeled as aggregations containing one ore more nodes, which are all linked together by a central node, representing the group (see Figure 3.9).

The node's shape, size, color and label can be set by a parameter, according to i.e., an attribute of the corresponding resource. Similar the edge's length, line width, color and arrow shape can be assigned by an attribute of the link. This allows the inclusion of additional data attributes to the visualization.

The user has several ways to interact with this visualization. For the first, he simply can drag a node to a place he prefers. Secondly, he can add and remove nodes to and from the visualization. He also can add and remove edges to the graph. The appearance of the nodes (label, color and so on) and the edges can be set according to a parameter.

And there is also the possibility to link links together, meaning that two links are replaced by one (circumventing the intermediate node). When connecting links there are basically two options, either both links are pointing in the same direction or they are facing towards to or away from each other. In the first case the new generated link has the same direction as the two original links, in the second case, the new generated link is undirected (see Figure 3.10). An example

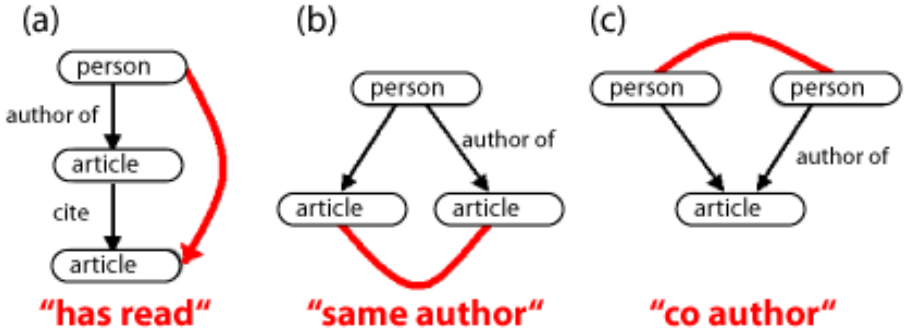


Fig. 3.10: (a) two links in the same direction, an "author of" and a "cite" link are connected to an directed "has read" link, (b) two incoming "author of" links are connected to an undirected "same author" link, (c) two outgoing "author of" links are connected to an undirected "co-author" link

(for the first case) may be an "author of" link from a person to an article and a "cites" link from the article to another article: by connecting these two links, an "has read" link from the person to the second article can be derived.

Finally, the user can add all nodes to the visualization which are linked to a starting node (see Figure 3.11). These features should allow the user to explore the article network visually and interactively.

3.2 Technical Concept

This section deals with the technical concept of LinkVis. Beginning with a section giving an overview of the whole system, important aspects are described in the following sections, like data structure, web, peer-to-peer networking, user interface and interaction.

3.2.1 Architecture

The LinkVis architecture is divided into three layers, which are associated with different tasks. These layers can be seen in Figure 3.12. The first layer considers the functionality of interaction with the user, therefore including components which allow the user to view and interact with the data. Besides general GUI-elements which allow the user to import, specify or search for data, a variety of views on the data are contained in this layer. A more detailed description of these components will be given in Section 3.2.5.

The task of the second layer is to manage the communication between the first layer and the third layer (the user interface and the data sources respectively). It therefore is responsible to handle queries it receives from the user interface and also to combine and synchronize the data it gets from the different data sources. This component is the core of the system, managing the flow

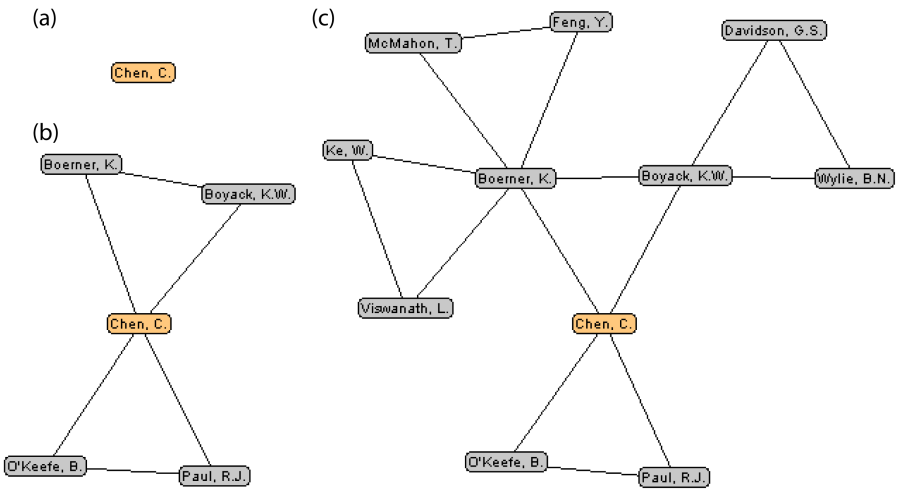


Fig. 3.11: (a) a starting node, (b) nodes connected by "co-author" links are added, (c) repeated adding of co-authors.

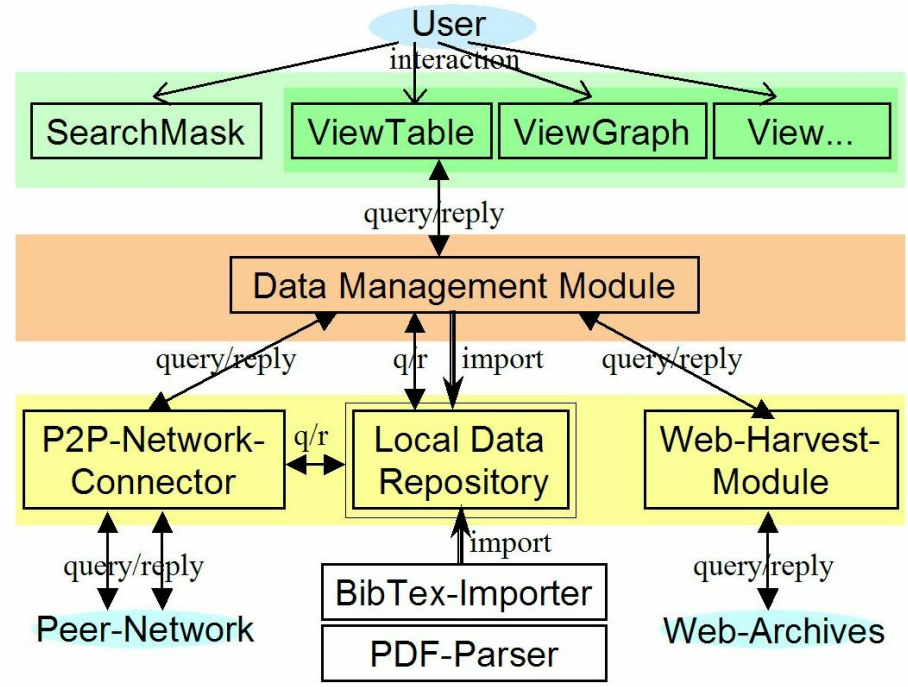


Fig. 3.12: The layered architecture of LinkVis; green: user interface layer, red: management layer, yellow: data source layer.

of the data.

The third layer consists of data sources. The main part here is the local data repository, which stores the data available to the peer. Other components in this layer include a peer-to-peer connector, which is responsible for maintaining a connection to the peer-to-peer network as well as to deal with queries from and to the network. Another type of components are web harvesting modules, which connect to web archives offering an appropriate interface (i.e., the Open Archives Initiative Protocol for Harvesting Metadata [59, 14]).

The exchange of query and response messages between the modules is handled by simple interface calls. An alternative approach may use XML-files transmitted by TCP for the query and response messages exchanged between the layers. That would result in highly independent components, which could be implemented in different programming languages and even be run on different machines, while still function as one system. But this approach would be slightly over-complicated and would also increase the network traffic.

3.2.2 Data Structure

The data structure of LinkVis is designed to give the user as much freedom as possible in specifying and adapting the system to his needs. In short the user is allowed to declare different resources, attributes corresponding to these resources, links between different kind of resources and groupings of resources.

Considering now the data which is needed for this application, the following entities can be identified:

- Resources: Entities are needed which represent articles or persons. Since both have a variable number of optional attributes, links and groups, the only required attribute of a resource is an identifier (key). This reduction of resources to a simple key, allows further the use of resources of any other kind (i.e., journal resources).

- Attributes: Attributes are (attribute name, value) pairs, which are assigned to a resource (like RDF triples). Further each attribute name can be assigned only one concrete value. That leads to the fact, that "attributes" containing multiple values (like authorship or keywords in the case of publications) are best modeled as links between the resource and multiple "value" resources (in the example before, person resources for authorship and keyword resources for keywords). The link type hereby corresponds to the attribute name.

- Links: A link is a typed connection between two resources. Therefore a link corresponds to a (source resource, target resource, link type) triple (again like RDF). Furthermore a link also can be seen as a resource, since a link may have additional attributes, links to other links or be in a group with other links.

- Groups: Groups are sets of resources. The number of contained resources is variable or the set even may be empty. Therefore the only requirement

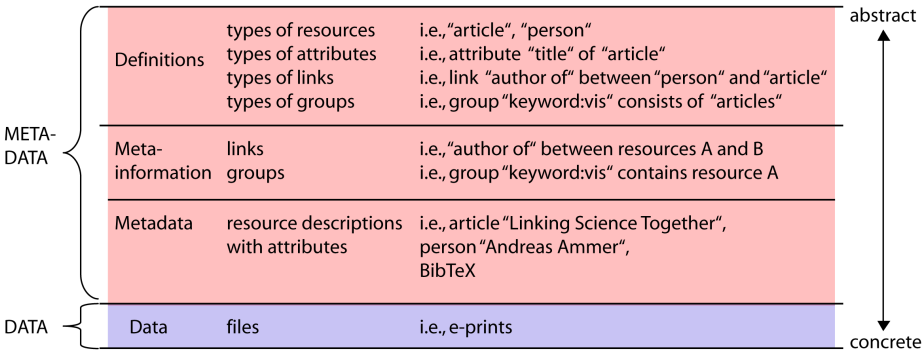


Fig. 3.13: This figure shows the abstraction level of various data elements. The data in each layer describes the data of the layer below (the definitions describe both metadata and meta-information). In LinkVis all four layers can be modified and shared.

here is a group name, to identify a group. Groups also may be seen as resources, since groups also can have additional attributes, links to other groups or be in a group with other groups. Groups also can be modeled as links between the resource representing the group and the resources contained in the group.

Summarizing, it can be stated that many aspects of the data can be modeled as an attribute, as well as a link, as well as a group. For an example consider the attribute "topic" of a publication resource. First each publication resource could simply given a corresponding attribute. Second a topic resource for each topic of a given ontology can be created, and the corresponding publication resources link to these topic resources. Or third, a group for each topic is used, and the publication resources are added accordingly to these groups.

Nevertheless, each way of representing the data has its advantages and drawbacks. Therefore the user is allowed to choose the model, which suits best his requirements.

To keep track of the different kind of resources, their corresponding attributes, the different types of links and groups, a definition level is introduced. The following definition entities, describe the four concrete entities identified earlier (Figure 3.13 illustrates the abstraction level of the data):

- Resources definition: The resource definition is used to define and type resources. It is clearly useful to be able to distinguish different types of resources at a relative early stage of processing (i.e., for building search queries, or visualizing only one type of resources).

- Attribute definition: The attribute definition shows what attributes a resource type can have.

- Link definition: The link definition declares which type of links can be established between what kind of resources.
- Group definition: The group definition shows what type of resources are contained in a group.

With these additional four definition entities, it is possible to formulate meaningful queries as well as augment the visualizations and user interface with useful hints (i.e., showing possible link types when adding a new link between two entities).

Figure 3.14 shows the resulting eight entities and the relations between them. A detailed list of these resulting eight entities, with their attributes and examples follows:

1. **Resource definition** – for defining types of resources

   Type: the name of the resource type

   Example 1: (article) – there are resources like articles.

   Example 2: (person) – there are resources like persons.

2. **Attribute definition** – for defining attributes of a resource type

   Attribute name: the name of the attribute

   Resource type: the type of the resource, which has this attribute

   Example 1: (title, article) – title is an usual attribute for article resources.

   Example 2: (year, article) – articles can be attributed with the year when they were published.

   Example 3: (name, person) – persons have a name.

3. **Link definition** – for defining link types between resources

   Link type: the name of the link type

   Resource1 type: the type of the resource this link type originates from

   Resource2 type: the type of the resource this link type points to

   Example 1: (cites, article, article) – A citation link occurs between two article resources.

   Example 2: (author of, person, article) – The "author of" connection is between a person and an article resource.

4. **Group definition** – for defining groups among resources

   Group name: the name of the group

   Resource type: the type of resources, which may be contained in this group

   Example 1: (Keyword: Visualization, article) – The group "Keyword: Visualization" contains articles.

   Example 2: (Staff of Vienna University of Technology, person) – The group "Staff of Vienna University of Technology" consists of persons.

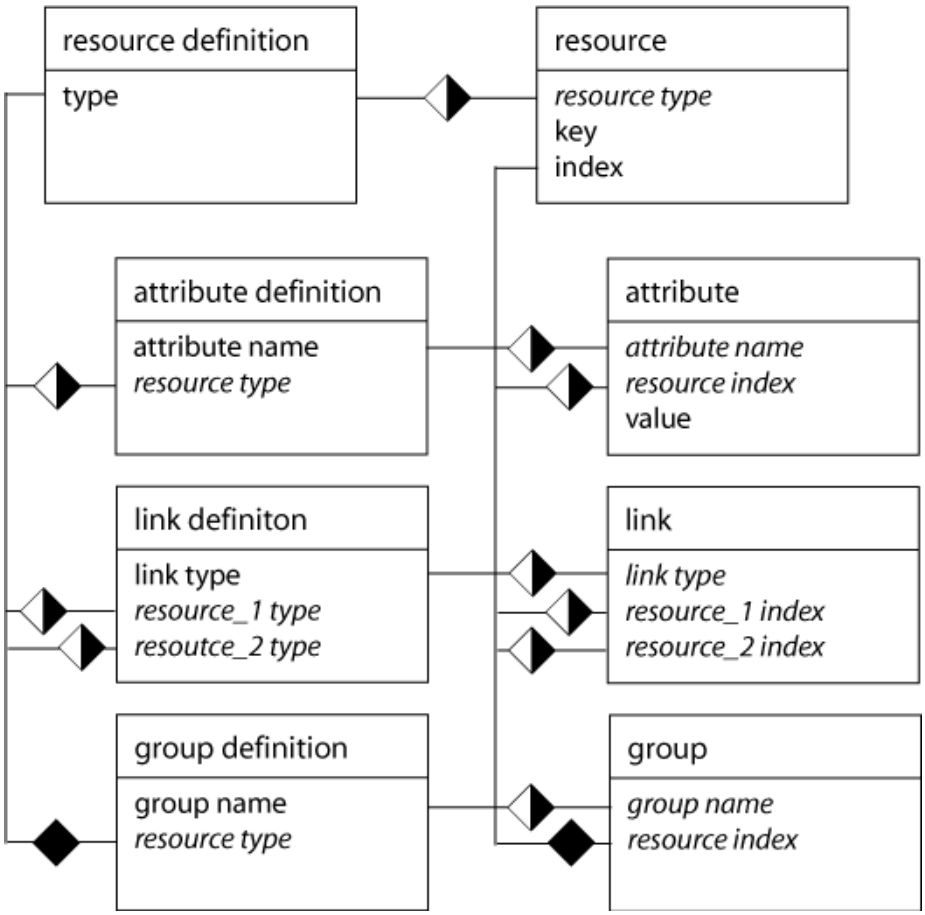


Fig. 3.14: Entity-relationship diagram of the underlying data structure; italic attributes correspond to foreign keys.

5. **Resource** – a concrete instance of a resource

   Resource type: the type of the resource

   Key: a key, which is used to identify the resource uniquely

   Index: an index, used internally to associate resources with attributes, links and groups

   Example 1: (article, Linking Science Together: How Networking Can Support Research – a Peer-to-Peer Approach, 1001) – A resource of type article with key "Linking Science..." and index 1001.

   Example 2: (person, Andreas Ammer, 1002) – A resource of type person with key "Andreas Ammer" and index 1002.

6. **Attribute** – an attribute of a resource

   Attribute name: the name of the attribute

   Resource index: the index of the resource, this attribute belongs to

   Value: the value of the attribute

   Example 1: (year, 1001, 2008) – The resource with index 1001 has a "year" attribute with value "2008".

   Example 2: (first name, 1002, Andreas) – The resource with index 1002 has an "fist name" attribute with value "Andreas".

7. **Link** – a link between two resources

   Link type: the type of the link

   Resource1 index: the index of the resource the link originates from

   Resource2 index: the index of the resource the link points to

   Example 1: (cite, 1001, 473) – There is a citation link between resources 1001 and 473.

   Example 2: (author of, 1002, 1001) – There is an "author of" link between resource 1002 and resource 1001.

8. **Group** – a group membership of one resource

   Group name: the name of the group

   Resource index: the index of the resource, which is member of this group

   Example 1: (Staff of Vienna University of Technology, 209) – The resource with index 209 is in the group "Staff of Vienna University of Technology".

   Example 2: (Keyword: Visualization, 1001) – The resource with index 1001 is in the group "Keyword: Visualization".

In consequence of this data structure, the LinkVis system is flexible enough do support not only the area of scientific publications, but also any other application area, which can be modeled as a network of interlinked nodes. A possible application could be for example a movie metadata set, where actors and directors are linked to movies they have participated and directed respectively.

3.2.3 Web

For the web harvesting modules, the OAI-protocol for metadata harvesting is used (already mentioned in Section 2.2). This protocol is supported already by some large scientific web archives and fulfills all requirements necessary for the tasks of querying and retrieving scientific publications.

Since LinkVis is implemented in Java, it is also possible to run LinkVis as an applet in a web browser. So, if an appropriate interface module is available, LinkVis also could be used as web interface for web archives.

3.2.4 Peer-to-Peer Networking

For the benefit of platform independence and interoperability the peer-to-peer platform JXTA is used. JXTA uses a super-peer based model with advertisement publishing (data describing the offered content is stored at the super-peers) for content location.

The definition entities (the entities 1-4 described in Section 3.2.2) are published in this way. So when a peer connects to the network, it is informed about all resource types which are available, along with according attributes, groups and links between them.

However the data structure used for the concrete entities (the entities 5-8 described in Section 3.2.2) is slightly different from that used internally in the system. The resource entity is encapsulated with all its belonging attribute entities and advertised as a whole. This is useful, because the number, the type and even the value of the attributes are considered to be rather stable. In contrast to that, the links and the group memberships are thought to be changing relatively quickly and therefore are published separately.

After connecting to the network and receiving the resource types (and the attribute, group and link definitions), a peer is able to search for specific resources. Possible searches include searches by (attribute, value) pair, (key, link type) pair, and group membership.

To enable file transfers peers also publish (file key, peer address) pairs. If a data link to a resource is given, the file can be downloaded by establishing a direct connection to a peer which advertises the corresponding file key.

3.2.5 User Interface

The user interface is designed so that the information flow is directed from top to bottom and from left to right (see Figure 3.15 and 3.16).

In the case of the LinkVis user interface, that means that the data which is corresponding to the information the user has selected in the explorer or entered in the search mask, is shown in the table view (actual in different tables). There the user can select the items he wants to visualize in the graph view. The detail and the combined link and group view are showing the data of the one resource which is currently selected (in the graph view or the table view). A screenshot of LinkVis is shown in Figure 3.16.

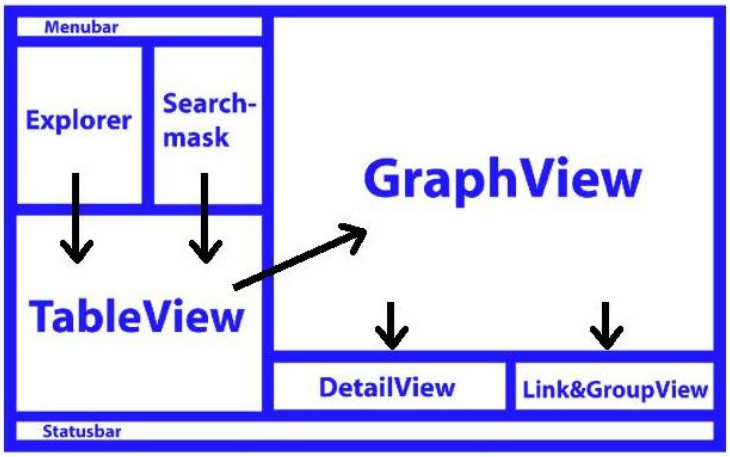


Fig. 3.15: Schema of the user interface, showing the information flow

This follows Shneiderman's "Visual Information Seeking Mantra" mentioned in Section 2.4. LinkVis first gives an overview of the available resource types (along with attributes, link types and groups), then the user may choose a resource type he wants to focus on while filtering out all other types. He can also specify another filter by search query. The details of one resource can be examined by simply selecting it. The other tasks Shneiderman mentions – relate, history and extract – are, with the exception of history, also given.

For more information about the user interface see Section 4.5.

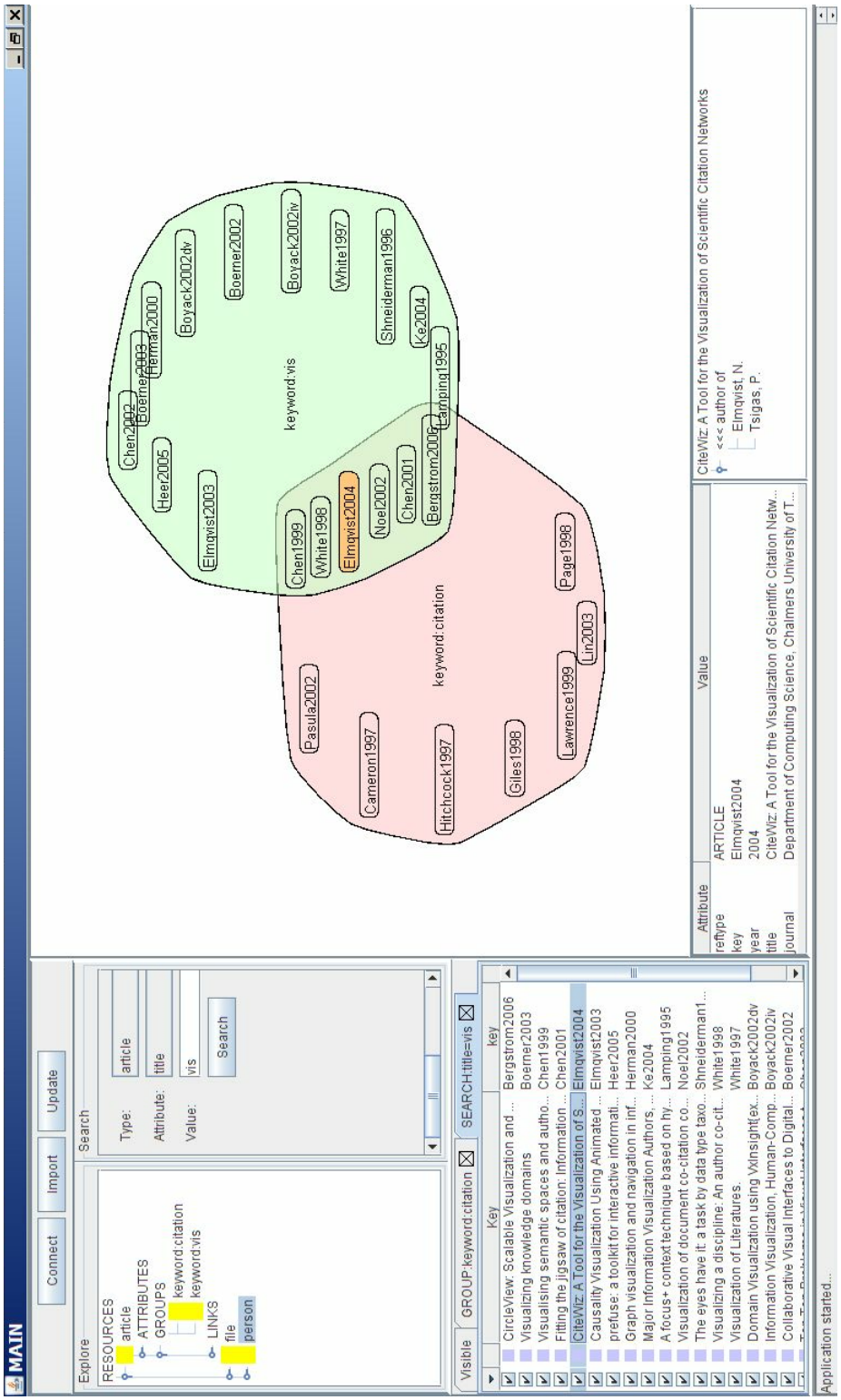


Fig. 3.16: Screenshot of LinkVis

4. IMPLEMENTATION ISSUES

This chapter deals with the implementation of LinkVis. First some design principles are given, then the implementation issues concerning the aspects in Chapter 3.2 are handled in the respective subsections.

4.1 Design Principles

The first design principle of LinkVis is to set the user at the center, which means that LinkVis is following a user-centered approach. So focusing on the needs of the user, starting from visualization and interaction, data structures and algorithms are designed, which are needed to fulfill the required tasks. Beside the functional requirements identified in Chapter 3.1.3, the following technical principles are added:

- Platform independence: only platform independent techniques, like Java [11] and JXTA [12] are used, so that all users are allowed to join the knowledge sharing.
- Modularity: to ease adding further features, it is tried to factorize LinkVis in rather independent modules.

4.2 Architecture

The layering of the architecture described in Chapter 3.2.1 is reflected by the modularization into packages. The next sections describe the packages and the containing classes in more detail.

4.2.1 Package Data

The data package contains the classes for defining the data structure, accessing the data base, importing and exporting data and handling data queries.

The DataManager module hereby plays a central role, it corresponds to the data management layer described in Section 3.2.1. Other modules like views can subscribe to the management module to be informed when an aspect of the data changes. If a change appears, the DataManager module calls the update functions of the subscribers along with a parameter to indicate what was changed (like the design pattern observer-notifier [45]). Table 4.1 shows the containing classes of the data package and their purposes.

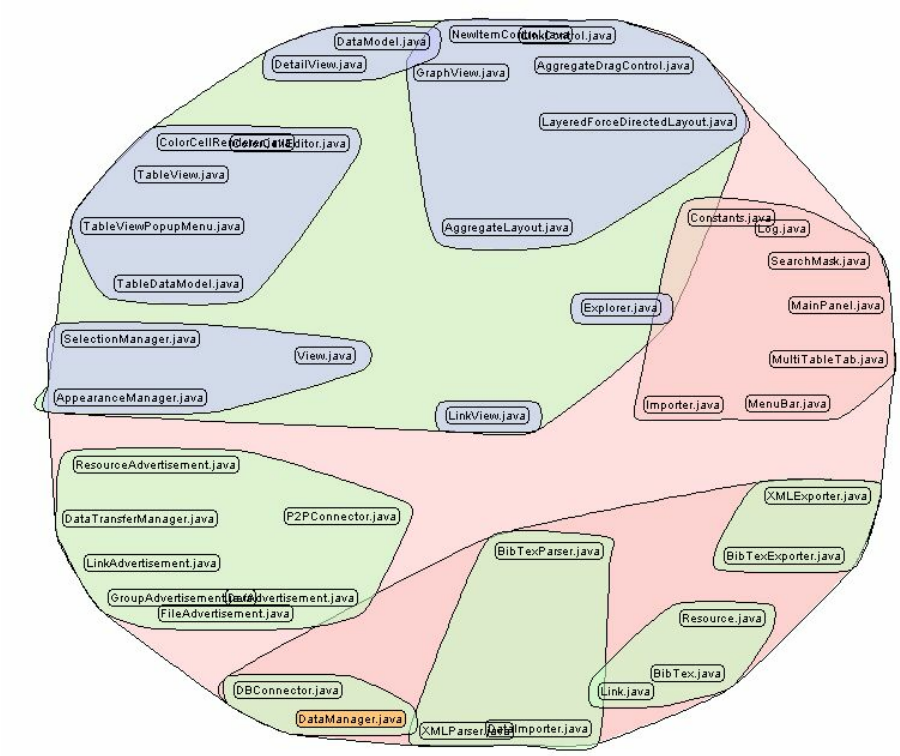


Fig. 4.1: The package structure of LinkVis. This picture was generated with LinkVis by importing the source folder of LinkVis, and grouping the files according to package.

class	package	description
Resource	data.structure	class representing the resource entity
BibTex	data.structure	holds the information needed to deal with BibTeXentries
Link	data.structure	class representing the link entity
Group	data.structure	class representing the group entity
DataImporter	data.imports	data importing module, can import directories; uses BibTex and XML importer classes
BibTexImporter	data.imports	importer class for BibTeXfiles
XMLImporter	data.imports	importer class for XML files
BibTexExporter	data.exports	exporter class for BibTeXfiles
XMLExporter	data.exports	exporter class for XML files
DataManager	data.management	management module, handles data queries
DBConnector	data.management	connector module to the database, encapsulating all SQL-Syntax

Tab. 4.1: The classes contained in the data package.

4.2.2 Package Network

The network package consists of classes for connecting to the peer-to-peer network, publishing and searching for advertisements, and for file transfers.

The P2PConnector module is of special importance; it implements all necessary functionality to connect to the peer-to-peer network, publish advertisements, send search queries and handle response messages. Also important is the DataTransferManager module, which listens to file transfer requests, answers these with sending the requested file and is capable of sending file requests, along with receiving the incoming files. Table 4.2 shows the containing classes of this package.

4.2.3 Package GUI

This package holds classes dealing with the text-based user interface. The corresponding classes of the gui package are shown in Table 4.3.

The MainPanel class can be used in an application as well as in an applet. Since the starting process of applications and applets differs slightly, a separate main class exists for both cases (they are not included in the listed packages).

class	package	description
DefAdvertitsement	network.-advertisement	class representing a definition advertisements
Resource-Advertisement	network.-advertisement	class representing a resource advertisements
LinkAdvertisement	network.-advertisement	class representing a link advertisements
GroupAvertisement	network.-advertisement	class representing a group advertisements
FileAvertisement	network.-advertisement	class representing a file advertisements
P2PConnector	network.-management	module responsible for maintaining the connection to the peer-to-peer network
DataTransfer-Manager	network.-management	module handling the actual file transfers

Tab. 4.2: The classes contained in the network package.

class	package	description
Importer	gui	class with the user interface to import files and directories
Log	gui	text area which can be used for logging, used as status bar
MenuBar	gui	class with the gui for starting imports, connecting to the network and so on
SearchMask	gui	class with the user interface to enter queries
MultiTableTab	gui	provides the functionality for tabbed table views
MainPanel	gui	class holding all gui-components and views, can be used in an application or an applet

Tab. 4.3: The classes contained in the gui package.

4.2.4 Package Views

The view package is rather a combination of multiple packages, which contains the various views. The content is shown in Table 4.4.

4.2.5 Used Libraries

LinkVis is implemented in Java. Further all used libraries are also platform independent and open-source, what makes the whole system independent of the used operating system and also possibly open-source.

LinkVis uses the following libraries:

- JXTA: the peer-to-peer platform LinkVis uses.
- prefuse: prefuse (by Heer et al. [16, 51]) is a graph drawing toolkit, which is used by the graph view.
- mysql-connector: this library is used to connect to the MySql-database, which is used as local data repository.

4.3 Data Management

For the realization of the data structure, a MySql-database-server was used. This leads to good scalability of the data collection, in trade off with a slightly more complicated installation routine.

LinkVis uses two databases with the same set of tables (see following subsection). One is used for caching purposes, the other for persistent data. When data is retrieved from the peer-to-peer network or a web harvesting module (i.e., by search responses), it is first stored in the cache database. The user has the possibility to store this data permanently in the database dedicated for persistent storage. This gives the user the control over which data is stored on his computer.

It is important to notice that the current implementation of the data structure basically supports only strings. Generally, it would be possible to use non-string attributes, if conversion wrappers would be put around these attributes.

4.3.1 Data Structure

The data tables corresponds to the eight entities described in Section 3.2.2, namely: resource definition, attribute definition, link definition, group definition, resource, attribute, link and group.

The data structure needed for a bibliography is generated by the BibTeX-parser during the importing process. Resources like "articles" and "persons" are created, the attributes are derived from the BibTeX-attributes, and "author of" links are established between the corresponding resources (the corresponding definitions were added too).

class	package	description
View	views.management	interface a view must implement
AppearanceManager	views.management	manages the appearance (color, label) of visual items
SelectionManager	views.management	manages the selection
Explorer	views.explorer	contains the explorer functionality
DetailView	views.detailview	the detail view, shows the attributes of the selected resource
DataModel	views.detailview	the data model used by the detail view table
LinkView	views.linkview	the link and group view, shows links and groups of the selected resource
TableView	views.tableview	the table view module
TableDataModel	views.tableview	holds the data model used by the table
TableViewPopupMenu	views.tableview	provides the functionality for the popup menu of the table
ColorCellRenderer	views.tableview	allows to render colors in the table
ColorCellEditor	views.tableview	allows to edit colors
GraphView	views.graphview	the graph view
LayeredForceDirectedLayout	views.graphview	the spring layout
AggregateLayout	views.graphview	layout which allows aggregating
LinkControl	views.graphview	provides the functionality to link resources together
AggregateDragControl	views.graphview	allows dragging of aggregates
NewItemControl	views.graphview	allows to create new items directly in the graph view

Tab. 4.4: The classes contained in the view package.

To include the actual publications themselves it is also possible to include files in LinkVis. They are modeled as resource entities, where the key corresponds to the hash of the file and one special attribute points to the path of the file on the local system. Metadata resources can be linked to such data entities by a special link type (a "metadata-data" link), reserved for such metadata-data-connections.

The following resources are created during the importing process:

- Resource "article": The key of the resource is set to the title of the publication. The corresponding (attribute name, value) pairs are extracted from the BibTeX-file.
- Resource "person": describes an author. The key is set to the name of the person.
- Resource "file": A hash of the file is computed and used as key for the resource. (Attribute name, value) pairs are derived from the file system, including i.e. filename, size, and date of last modification.

These entities are connected by the following links:

- Link "author of": is used to link "person" resources to "article" resources.
- Link "metadata-data": is used to link any kind of metadata resource to a file resource (i.e., an "article" resource to a "file" resource representing an e-print).

Furthermore the user is allowed to specify any kind of resource types as well as arbitrary link types.

4.4 Information Exchange

For the information exchange the peer-to-peer platform JXTA was used. JXTA is an open-source project, which is itself still under development (and poorly documented). Nevertheless it offers some quite useful methods, although some functionality is still missing almost entirely.

4.4.1 Advertisements

JXTA uses a model of publishing, which stores information about the available content at so-called "super peers". These information pieces are called advertisements. LinkVis uses several types of advertisements, as follows:

- ResourceAdvertisement: beside id, type and key of the resource, all contained attributes of the resource are included in the advertisement.
- LinkAdvertisement: a link advertisement contains beside id and "link type", the key and the type of both resources, which are interlinked.

- GroupAdvertisement: a group advertisement contains beside id and groupname the key and the type of one resource.
- DefAdvertisement: definition advertisements are used to publish declarations of resources, attributes, links and groups as well. Therefore they contains beside id a "definition type", a key, and two arbitrary value fields, which contain the information needed according to what is declared.
- FileAdvertisement: file advertisements are used to publish (file key, peer address) pairs. This information is needed to allow the exchange of data files.

4.4.2 Connecting and Searching

When a peer successfully connects to the network it starts two processes: First, it tries to publish advertisements about all the data it offers to share, second it sends a search request for definition advertisements (see the section before). The first allows other peers to get access to the offered data, the latter allows the local peer to get information about what resources, attributes, links and groups are shared by others. After receiving this data, the local peer is able to send meaningful search requests.

The search methods the LinkVis system offers are relatively simple. At the moment, LinkVis allows only one (attribute, value) pair as search parameter. The user is restricted to one (attribute, value) pair, (resource key, link type) pair or group membership item for each search query.

4.5 User Interface

The user interface of LinkVis is implemented with the Java Swing classes. The following sections will describe the various parts of the user interface in more detail.

4.5.1 The Main Panel

The main panel contains besides menu and status bar, the explorer panel, the search mask, the table, the graph, the detail and the link and group view (see Figure 3.16). This panel can be used in an application as well as in an applet.

4.5.2 The Explorer Panel

The explorer panel shows the available resource types, along with according attribute types, link types and groups (see Figure 4.2). Here it is possible to define new resource and attribute types as well as groups. By selecting the resource types or groups the user wants to examine, the resources corresponding to that type or group are shown in the table view.

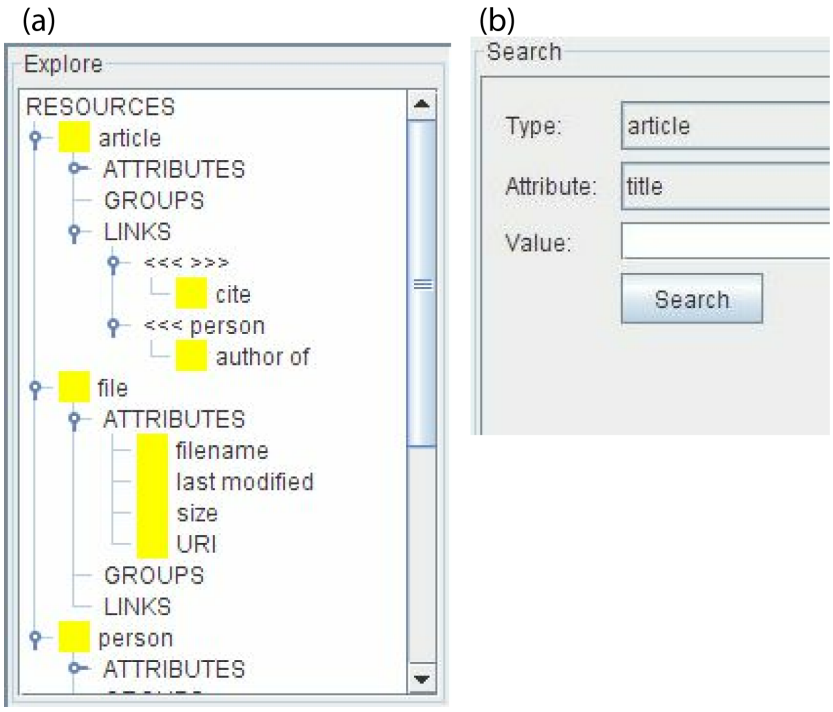


Fig. 4.2: (a) The explorer panel. (b) The search mask

The color encoding next to a resource type, attribute type, link type or group type refers to whether the item is in the cache database or in the persistent database.

4.5.3 The Search Mask

The search mask allows the user to execute a search query with an (attribute, value) pair. The first combobox shows the available resource types, the second the available attributes of the resource type selected in the first combobox. In the textfield the user can enter the value (see Figure 4.2).

Both explorer panel and search mask are updated, when new resource types are received from the peer-to-peer network.

4.5.4 The Table View

The table view shows a list of resources, which can be selected by type (in the explorer panel), by group (also in the explorer panel) or by search query (in the search mask). All these selections open a new table, which is shown in a new tab. There is also one special tab (which can not be closed in contrast to the others), which is reserved to show the visual items, which are shown in the graph view at the moment.

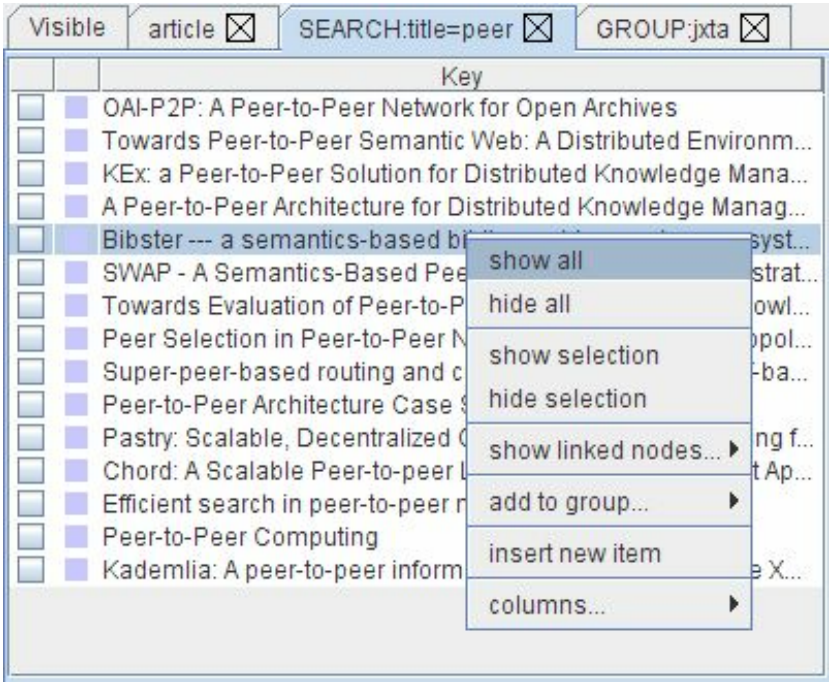


Fig. 4.3: The table view.

The first column is a checkbox, which indicates if the item is visible in the graph view. The second column is a color coding reserved for the same purpose as the color coding in the explorer panel. The third column contains the key of the resource. Other columns can be shown by choosing the appropriate action in the popup menu (i.e., number of links, or any attribute of the resource type). This is quite helpful if you want to order the resources by a special attribute (i.e., year of publication).

The table view offers also other features, like selecting items which to show or remove from the graph view, adding resources to groups, or specifying new resources. For a screenshot of the table view with its popup menu see Figure 4.3.

4.5.5 The Detail View and the Combined Link and Group View

The detail view shows the attributes with their corresponding values of the resource, which is selected in the graph or table view. Here it is possible to edit the attributes (see Figure 4.4).

The link and group view shows the links and groups of the resource, selected in the graph or table view (see Figure 4.5.). The links are grouped by link type (i.e., "author of" or "cite") and it is indicated if the links are outgoing or incoming (by "> > >" and "< < <" respectively).

Fig. 4.4: The detail view.

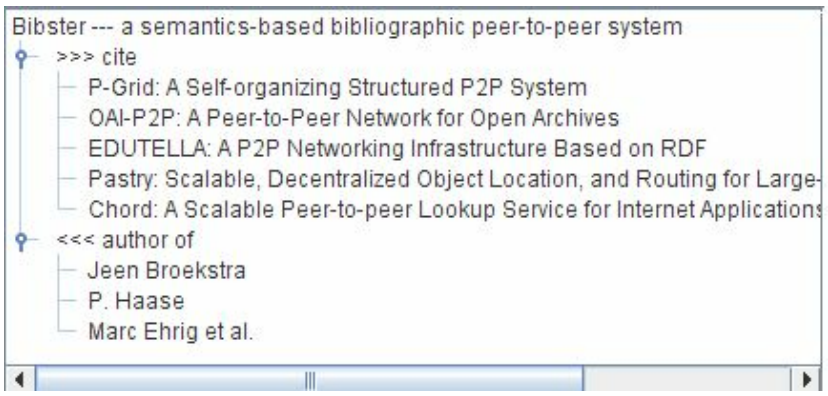


Fig. 4.5: The link and group view.

4.6 Basic Interactions

This section describes the basic interactions available in LinkVis.

4.6.1 Importing

When a user begins to work with LinkVis, he may want to import the collection of publications, he already has. To do this, he chooses the option "Import" at the menu bar and selects the files and folders, which he wants to import. When selecting a folder, all included files and folders are imported recursively. BIBTEX-files are parsed, and "article" and "person" resources are created according to the information found. Other files, beside BIBTEX-files are imported as data files. So when the user wants to import his publication collection, he simply has to choose the folder, which contains the papers in electronic format and the one or more files, which contains the corresponding BIBTEX-entries.

After that, the explorer shows the available resource types (in this example article, person and file) with the available attribute and link types and the data is ready to be visualized. At this first stage already co-author-maps can be shown. For examples see Figure 3.1 and 3.2.

4.6.2 Grouping

To sort the collection of publications, it is possible to group items. A useful approach here would be executing a search query for a selected keyword, and add all publications to a group, which correspond to this keyword (see Figure 4.6). Of course it is also possible to choose other selection parameters, as well as select group membership fully manually.

4.6.3 Linking

To add connections between resources, it is possible to add links between them. Besides the link type "author of", which is valid between persons and articles, and the link type "metadata-data" which is valid between any resource type and files, the user can specify all variations of link types.

To link resources together, the user just clicks at the visual representations of the two resources, he wants to connect, and choose the appropriate link type (see Figure 4.7).

4.6.4 Searching

After connecting to the peer-to-peer network, the declarations of what resources, attributes, groups and links are available are received and shown in the explorer panel.

Search queries by an (attribute, value) pair can be executed by using the search mask. Searches by a (resource key, link type) pair can be started by clicking at a resource in the table view or the graph view and selecting a link

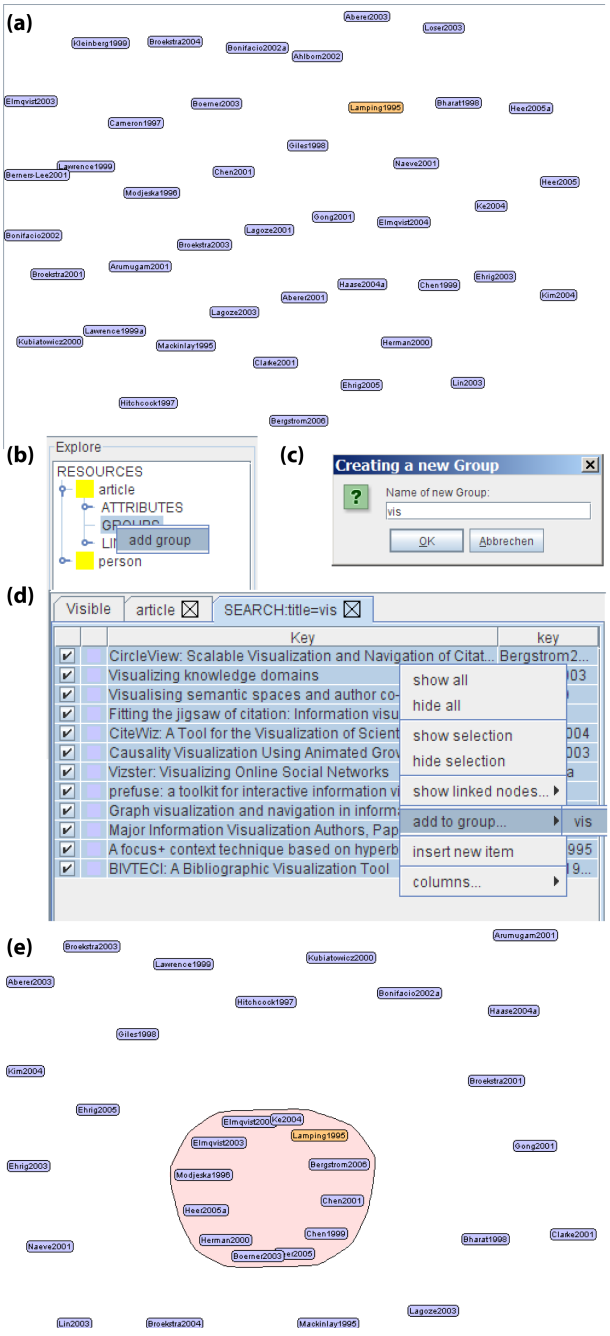


Fig. 4.6: Grouping: (a) unordered nodes, (b) and (c) define group name, (d) use the search mask to get resources with a special attribute, (e) add found resources to corresponding group, (f) grouped nodes.

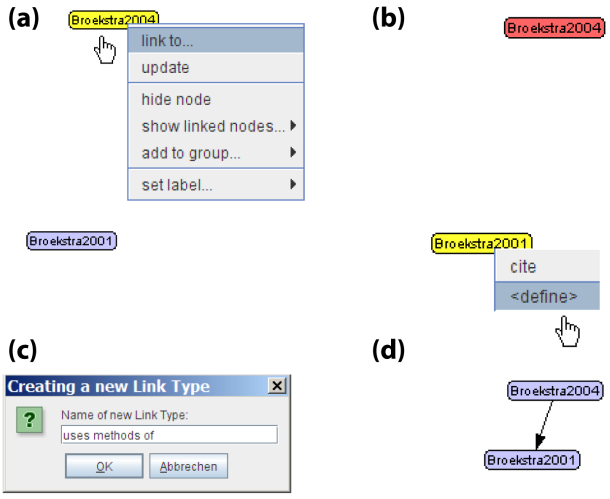


Fig. 4.7: Linking: (a) click on source node to open popup menu and select "link to", (b) click on target node and select link type, (c) optionally define new link type, (d) the link is added.

type in the corresponding popup menu. Group membership searches simply can be started by choosing a group in the explorer panel.

4.6.5 Defining

One of the strengths of LinkVis is that it does not depend on a specific kind of data (along as it can be modeled as network of interlinked nodes). The user is allowed to define arbitrary resource types with any kind of attributes. These user-defined resources can be connected and grouped by arbitrary user-defined link types and groups. An example was created by importing the source folder of LinkVis and grouping the class files according to the packages (see Figure 4.1).

5. RELATED WORK

In this chapter related work is compared to LinkVis. The mentioned systems were also described in Chapter 2. This comparison should allow to identify differences, advantages and disadvantages to already existing solutions.

- Publication sharing communities: CiteULike [5] is a web-based publication sharing platform, which allows user to add tags to the publications. These tags are used to improve search results. Furthermore the users are allowed to group publications and share these groups among other users. CiteULike therefore has quite a similar goal to LinkVis, in allowing researchers to share user-specified information about publications among each other. Nevertheless CiteULike does not include any visualization methods or the functionality to add links between publications. LinkVis also uses a peer-to-peer approach instead of a web-based system, and allows the usage of any kind of data, beside publications.

- Semantic Peer-to-Peer: Semantic Peer-to-Peer Applications like Edutella [70], OAI-P2P [21] or Bibster [33] offer a platform to share ontologies. These ontologies can be used to improve search quality, routing performance and data aquisition. LinkVis is there quite similar; by sharing the definition level of the resources (see Figure 3.13) an common ontology is established. Again the main difference to LinkVis is that these applications does not include any visualization methods at all. Furthermore LinkVis includes a functionality to follow links. By following citation or co-author links, publications which are with high probability in the same or in a similar research area, can be found.

- Visual Interfaces to Digital Libraries: Although there are already some existing visual interfaces to digital libraries, almost every solution has its own drawbacks. More complex systems do not provide real-time visual interaction possibilities, which is acceptable, when the data set is rather stable and the visualizations can be pre-calculated and rendered. Also the functionality to include user-generated information is quite seldom. LinkVis provides real-time visual interaction by a graph view using the simple mass-spring-model. LinkVis also allows the user to include arbitrary information and share this information with all other users.

Summarizing LinkVis tries to combine the advantages of these three application areas into one system. LinkVis allows the research community to share

publications, along with user-generated metadata, like annotations, links and groups and also provides visualization methods to analyze them. This visualization methods and the functionality to follow links, enable researchers to visual explore the article network.

6. CONCLUSION AND RESULTS

6.1 Conclusion

Collaboration in scientific research becomes more and more important these days, therefore the need for tools supporting the researchers arise. Today's process of scientific literature research also requires optimization.

The system proposed in this work, called LinkVis, is a solution to these problems. LinkVis is a tool for searching and sharing publications with the features of attributing, linking and grouping articles in a way which is useful to other researchers in a similar area. Hence the insight gained during the process of scientific literature research can be shared with others, leading to an optimization of this stage of the overall research process by reducing the time necessary to identify important publications or finding out less obvious connections.

LinkVis therefore uses methods from peer-to-peer networking and visualization: the first to allow sharing attributes, links and groups as well as the publications themselves, the latter to analyze the connectivity network of the research area and to build "mental maps" of the domain.

Due to its flexible data structure, LinkVis also can be used for any other application area, where its features of visualization and sharing may be useful, as long as the data can be modeled as a network of interlinked nodes.

Summarizing, LinkVis provides researchers with a collaborative tool for the scientific literature research process, whose features of visualization and sharing help to accelerate and optimize this process.

6.2 Results

Here some resulting images of visualizations in LinkVis are given. The following visualizations are based on data extracted of the publication list of M. E. Gröller (with about 100 publications and about the same number of co-authors). For a the description of the images refer to the captions.

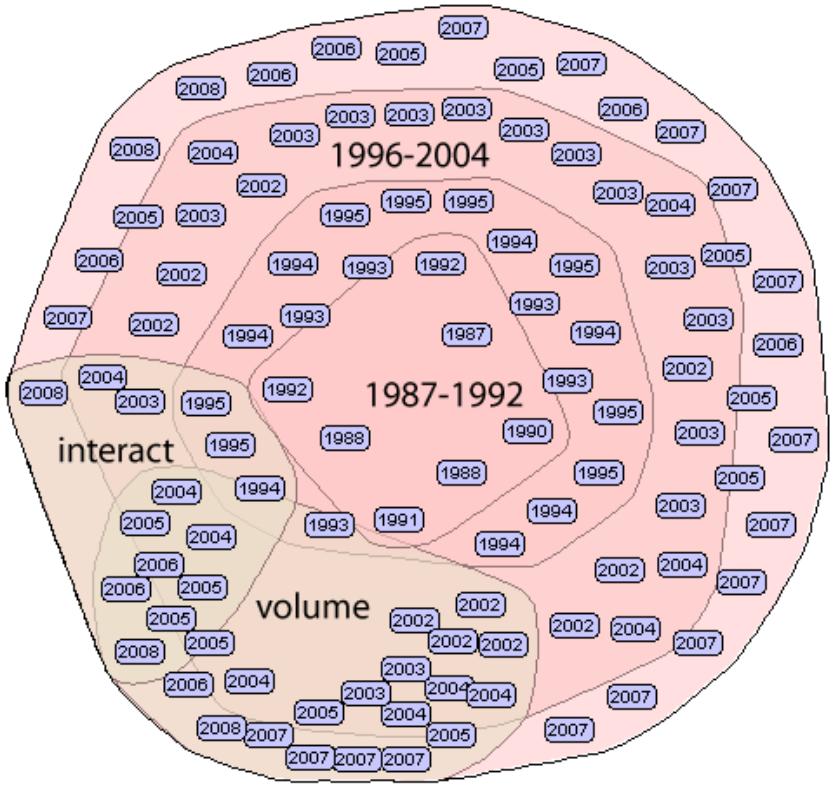


Fig. 6.1: This image shows the publications of M. E. Gröller, ordered by publication year. Early publications are located at the center, newer publications at the edge of the graph. The articles are labeled with the year when they were published (mouse-over shows title). Further two groups corresponding to the term "interact" and "volume" in the titles are included in the visualization.

.

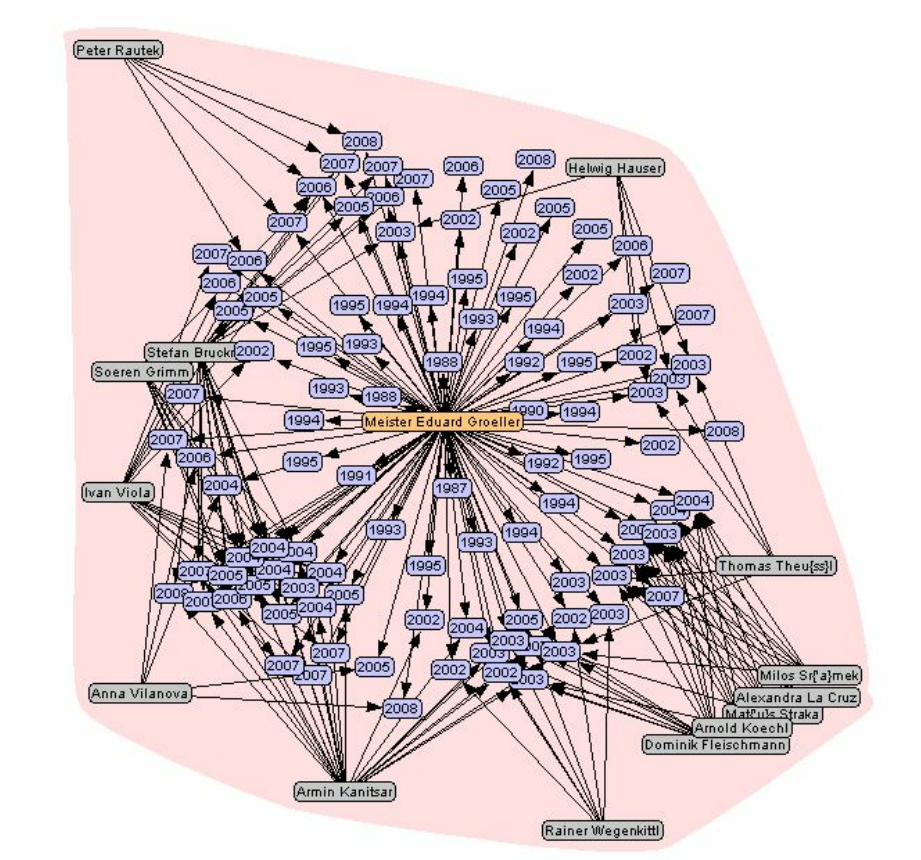


Fig. 6.2: This image shows co-authors of M. E. Gröller, who have five or more collaborative publications with him. Again the earlier publications (labeled with year) are located at the center, newer at the edge of the graph.

.

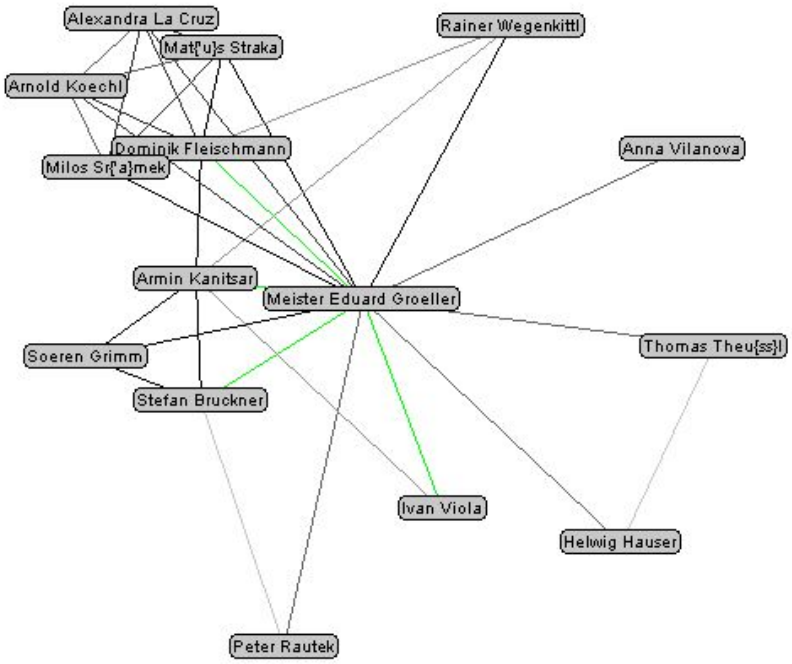


Fig. 6.3: This image shows again the co-authors of M. E. Gröller, who have five or more collaborative publications with him. The length and color of the links is indicating the number of collaborations. Green links mean that there are more than 10 common publications, black links mean that there are fewer. The opacity is used for the black links to differ between the actual numbers of publications, so co-author links with fewer collaborations are drawn lighter than links with more collaborations (only links corresponding to three or more common publications are shown).

.

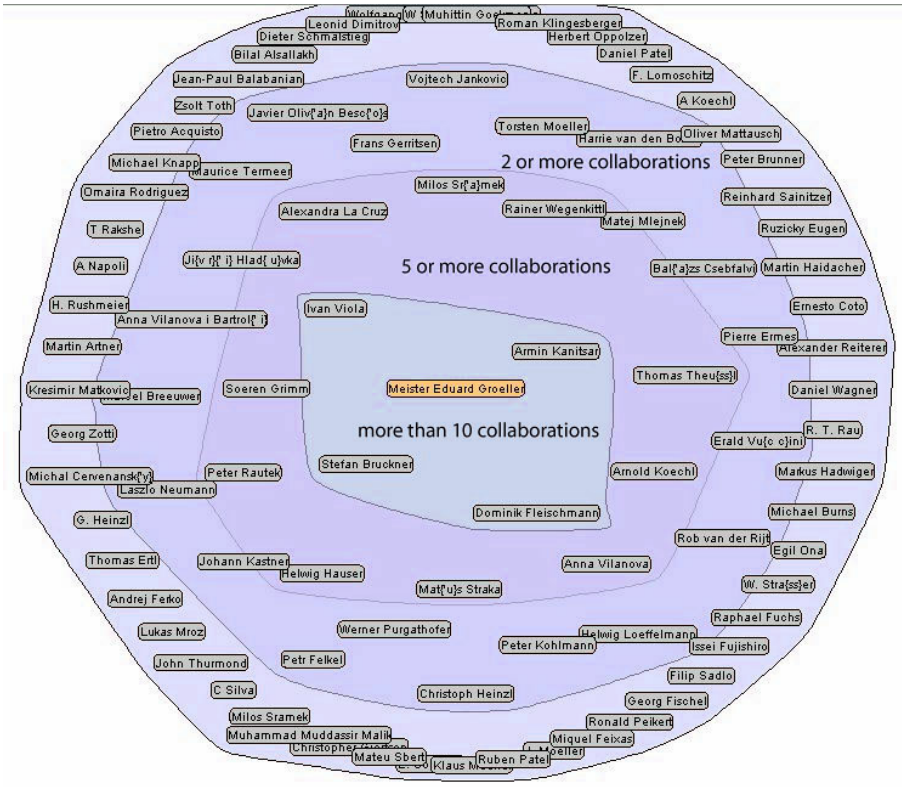


Fig. 6.4: In this image all co-authors are shown. Here the authors with the most common publications with M. E. Gröller are located in the center, authors with fewer collaborations are found on the edges.

.

BIBLIOGRAPHY

[1] The acm portal. http://portal.acm.org/. Online; accessed 21-Oct-2008.

[2] Adobe. http://www.adobe.com/. Online; accessed 21-Oct-2008.

[3] arxiv.org e-print archive. http://arxiv.org/. Online; accessed 21-Oct-2008.

[4] Bibster. http://bibster.semanticweb.org/. Online; accessed 21-Oct-2008.

[5] Citeulike: Everyone's library. http://www.citeulike.org/. Online; accessed 21-Oct-2008.

[6] Computer and information science papers citeseer publications research index. http://citeseer.ist.psu.edu/. Online; accessed 21-Oct-2008.

[7] Dublin core metadata element set. http://www.dublincore.org/documents/dces/. Online; accessed 21-Oct-2008.

[8] Endnote - bibliographies made easy. http://www.endnote.com/. Online; accessed 21-Oct-2008.

[9] Erdos number project. http://www.oakland.edu/enp/. Online; accessed 21-Oct-2008.

[10] Google scholar. http://scholar.google.com/. Online; accessed 21-Oct-2008.

[11] Java technology. http://www.sun.com/java/. Online; accessed 21-Oct-2008.

[12] jxta: Jxta community projects. https://jxta.dev.java.net/. Online; accessed 21-Oct-2008.

[13] Oaister. http://www.oaister.org/. Online; accessed 21-Oct-2008.

[14] Open archives initiative. http://www.openarchives.org/. Online; accessed 21-Oct-2008.

[15] The pirate bay - the world's largest bittorrent tracker. http://piratebay.org/. Online; accessed 21-Oct-2008.

[16] prefuse | interactive information visualization toolkit. `http://prefuse.org/`. Online; accessed 21-Oct-2008.

[17] Resource description framework (rdf) model and syntax specification. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`. Online; accessed 21-Oct-2008.

[18] Seti@home. `http://setiathome.berkeley.edu/`. Online; accessed 21-Oct-2008.

[19] K. Aberer. P-Grid: A self-organizing access structure for P2P information systems. *Sixth International Conference on Cooperative Information Systems (CoopIS 2001), Lecture Notes in Computer Science*, 2172:179–194, 2001.

[20] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt. P-grid: a self-organizing structured p2p system. *SIGMOD Rec.*, 32(3):29–33, 2003.

[21] B. Ahlborn, W. Nejdl, and W. Siberski. Oai-p2p: A peer-to-peer network for open archives. *International Conference on Parallel Processing Workshops*, 0:462, 2002.

[22] M. Arumugam, A. Sheth, and I.B. Arpinar. The peer-to-peer semantic web: A distributed environment for sharing semantic knowledge on the web. In *WWW2002 Workshop on Real World RDF and Semantic Web Applications. Honolulu, Hawaii (USA)*, volume 46, 2002.

[23] P. Bergström and E.J. Whitehead Jr. Circleview: a new approach for visualizing time-related multidimensional data sets. In *Proceedings of the 2006 Symposium on Interactive Visual Information Collections and Activity (IVICA 2006)*, 2006.

[24] T. Berners-Lee, R. Fielding, and L. Masinter. RFC 3986: Uniform Resource Identifier (URI): Generic Syntax. *The Internet Society*, 2005.

[25] T. Berners-Lee, J. Hendler, O. Lassila, and E. Meaning. The Semantic Web. *Scientific American*, 284(5):28–37, 2001.

[26] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 104–111, Melbourne, AU, 1998.

[27] K. Boerner, C. Chen, and K.W. Boyack. Visualizing knowledge domains. *Annual Review of Information Science and Technology*, 37(1):179–255, 2003.

[28] M. Bonifacio, P. Bouquet, G. Mameli, and M. Nori. KEx: A Peer-to-Peer Solution for Distributed Knowledge Management. *Lecture Notes in Computer Science*, pages 490–500, 2002.

[29] K. Börner, Y. Feng, and T. McMahon. Collaborative visual interfaces to digital libraries. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 279–280. ACM New York, NY, USA, 2002.

[30] K.W. Boyack, B.N. Wylie, and G.S. Davidson. Domain Visualization using VxInsight® for Science and Technology Management. *Journal of the American Society for Information Science and Technology*, 53(9):764–774, 2002.

[31] K.W. Boyack, B.N. Wylie, and G.S. Davidson. Information Visualization, Human-Computer Interaction, and Cognitive Psychology: Domain Visualizations. *Lecture Notes in Computer Science*, pages 145–160, 2002.

[32] D. Bricklin. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, chapter The Cornucopia of the Common. O'Reilly and Associates, Inc., 2003.

[33] J. Broekstra, P. Haase, M. Ehrig, F. van Harmelen, M. Menken, P. Mika, B. Schnizler, and R. Siebes. Bibster — a semantics-based bibliographic peer-to-peer system. In *Proc. of the 3rd Int. Semantic Web Conference, Hiroshima, Japan*, 2004.

[34] J. Broekstra and A. Kampman. SeRQL: A Second Generation RDF Query Language. *Proc. SWAD-Europe Workshop on Semantic Web Storage and Retrieval*, 2003.

[35] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. *Lecture Notes in Computer Science*, pages 54–68, 2002.

[36] R.D. Cameron. A universal citation database as a catalyst for reform in scholarly communication. *First Monday*, 2(4):1396–0466, 1997.

[37] C. Chen. Visualising semantic spaces and author co-citation networks in digital libraries. *Information Processing and Management*, 35(3):401–420, 1999.

[38] C. Chen and K. Börner. Top Ten Problems in Visual Interfaces to Digital Libraries. *Lecture Notes In Computer Science*, pages 226–231, 2002.

[39] C. Chen, R.J. Paul, and B. O'Keefe. Fitting the jigsaw of citation: Information visualization in domain analysis. *Journal of the American Society for Information Science and Technology*, 52(4):315–330, 2001.

[40] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. *Lecture Notes in Computer Science*, 2009:46–66, 2001.

[41] R. Dornfest and D. Brickley. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, chapter Metadata. O'Reilly and Associates, Inc., 2003.

[42] M. Ehrig, P. Haase, S. Staab, and C. Tempich. The SWAP Data and Metadata Model for Semantics-Based Peer-to-Peer Systems. *Lecture Notes in Computer Science*, pages 144–155, 2003.

[43] M. Ehrig, C. Schmitz, S. Staab, J. Tane, and C. Tempich. Towards Evaluation of Peer-to-Peer-Based Distributed Knowledge Management Systems. *Lecture Notes in Computer Science*, pages 73–88, 2004.

[44] GW Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 16–23. ACM New York, NY, USA, 1986.

[45] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Reading, MA, 1995.

[46] C. L. Giles, K. Bollacker, and S. Lawrence. CiteSeer: An automatic citation indexing system. In Ian Witten, Rob Akscyn, and Frank M. Shipman III, editors, *Digital Libraries 98 - The Third ACM Conference on Digital Libraries*, pages 89–98, Pittsburgh, PA, June 23–26 1998. ACM Press.

[47] L. Gong. Project JXTA: A Technology Overview. *Sun Microsystems*, 2001.

[48] J. W. Grossman. The Evolution of the Mathematical Research Collaboration Graph. *Congressus Numerantium*, pages 201–212, 2002.

[49] P. Haase, R. Siebes, and F. van Harmelen. Peer Selection in Peer-to-Peer Networks with Semantic Topologies. *Lecture Notes in Computer Science*, pages 108–125, 2004.

[50] E. Halepovic and R. Deters. The Costs of Using JXTA. *Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, page 160, 2003.

[51] J. Heer, S.K. Card, and J.A. Landay. prefuse: a toolkit for interactive information visualization. *Conference on Human Factors in Computing Systems*, pages 421–430, 2005.

[52] I. Herman, G. Melancon, and M.S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[53] T. Hong. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, chapter Performance. O'Reilly and Associates, Inc., 2003.

[54] S. Kim and E.J. Whitehead Jr. Properties of academic paper references. *Proceedings of the fifteenth ACM conference on Hypertext & hypermedia*, pages 44–45, 2004.

[55] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1997.

[56] T. Kohonen. The self-organizing map. *Neurocomputing*, 21(1-3):1–6, 1998.

[57] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela. Self Organization of a Massive Document Collection. *IEEE Transactions on Neural Networks*, 11(3), 2000.

[58] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. Oceanstore: An architecture for global-scale persistent storage. In *Proceedings of ACM ASPLOS*. ACM, November 2000.

[59] C. Lagoze and H. Van de Sompel. The Open Archives Initiative: Building a low-barrier interoperability framework. In *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 54–62. ACM Press New York, NY, USA, 2001.

[60] J. Lamping, R. Rao, and P. Pirolli. A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, 1995.

[61] L. Lamport. *Latex: a document preparation system*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

[62] S. Lawrence, K. Bollacker, and C. L. Giles. Autonomous citation matching. In Oren Etzioni, editor, *Proceedings of the Third International Conference on Autonomous Agents*, New York, 1999. ACM Press.

[63] S. Lawrence, K. Bollacker, and C. L. Giles. Indexing and retrieval of scientific literature. In *Eighth International Conference on Information and Knowledge Management, CIKM 99*, pages 139–146, Kansas City, Missouri, November 1999.

[64] P. Leach, M. Mealling, and R. Salz. A Universally Unique IDentifier (UUID) URN Namespace. *RFC4122, July*, 2005.

[65] A. Legout, G. Urvoy-Keller, and P. Michiardi. Understanding BitTorrent: An Experimental Perspective. *INRIA Sophia Antipolis/INRIA Rhne-Alpes-PLANETE INRIA France, EURECOM-Institut Eurecom, Tech. Rep., November*, 2005.

[66] X. Lin, H.D. White, and J. Buzydlowski. Real-time author co-citation mapping for online searching. *Information Processing and Management*, 39(5):689–706, 2003.

[67] A. Loser, M. Wolpers, W. Siberski, and W. Nejdl. Efficient data store discovery in a scientific P2P network. In *Proc. of the WS on Semantic Web Technologies for Searching and Retrieving Scientific Data, CEUR WS*, volume 83, 2003.

[68] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the XOR metric. *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, 258:263, 2002.

[69] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B.o Richard, S. Rollins, and Z. Xu. Peer-to-Peer Computing. Technical report, 2002.

[70] W. Nejdl, B. Wolf, C. Qu, S. Decker, Michael Sintek, Ambjoern Naeve, Mikael Nilsson, Matthias Palmer, and Tore Risch. Edutella: A p2p networking infrastructure based on rdf. In *Proceedings of the 11th international conference on World Wide Web*, pages 604–615. ACM New York, NY, USA, 2002.

[71] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Löser. Super-peer-based routing strategies for RDF-based peer-to-peer networks. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(2):177–186, 2004.

[72] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[73] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. *Advances in Neural Information Processing Systems*, pages 1425–1432, 2003.

[74] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *Proceedings of the 2001 SIGCOMM conference*, volume 31, pages 161–172. ACM New York, NY, USA, 2001.

[75] M. Ripeanu. Peer-to-Peer Architecture Case Study: Gnutella Network. In *Proceedings of International Conference on Peer-to-peer Computing*, volume 101. Sweden: IEEE Computer Press, 2001.

[76] J. Ritter. Why Gnutella Can't Scale. No, Really. `http://www.darkridge.com/~jpr5/doc/gnutella.html`. Online; accessed 21-Oct-2008.

[77] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329–350, 2001.

[78] M. Sarkar and M.H. Brown. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 83–91. ACM New York, NY, USA, 1992.

[79] B. Shneiderman. The eyes have it: a task by data type taxonomy for informationvisualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, pages 336–343. IEEE Computer Society Washington, DC, USA, 1996.

[80] B. Shneiderman. *Leonardo's Laptop: Human Needs and the New Computing Technologies*. MIT Press, 2002.

[81] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160. ACM New York, NY, USA, 2001.

[82] Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, and D. Oberle. The SWRC Ontology-Semantic Web for Research Communities. *Lecture Notes In Computer Science*, 3808:218, 2005.

[83] H.D. White and K.W. McCain. Visualization of Literatures. *Annual Review of Information Science and Technology (ARIST)*, 32:99–168, 1997.

[84] B. Yang and H. Garcia-Molina. Efficient search in peer-to-peer networks. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, 2002.

[85] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: A global-scale overlay for rapid service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53, 2004.