Institut für Informatik XI
der Technischen Universität München

BibRelEx: Erschließung bibliographischer Datenbasen durch Visualisierung von annotierten inhaltsbasierten Beziehungen

Britta Landgraf

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende:	Univ.-Prof. G. J. Klinker, Ph.D.
Prüfer der Dissertation:	
1.	Univ.-Prof. Dr. A. Brüggemann-Klein
2.	Univ.-Prof. Dr. Th. Ottmann Albert-Ludwigs-Universität Freiburg

Die Dissertation wurde am 18.11.2002 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 11.12.2003 angenommen.

Kurzfassung

Die Motivation für diese Arbeit ergab sich aus dem bekannten Problem der Informationsüberflutung und der daraus resultierenden dringlichen Notwendigkeit dem einzelnen Wissenschaftler effiziente Verfahren zur gezielten Informationsbeschaffung, zur Organisation der Literatur in seinem Arbeitsgebiet und zum Austausch über Arbeitsergebnisse an die Hand zu geben.

Um diesen Bedürfnissen gerecht zu werden, wird in dieser Arbeit ein Konzept für eine wissensbasierte Recherche und Literaturverwaltung präsentiert. Mit dem System BibRelEx, das prototypisch realisiert wurde, kann die Fachliteratur von Arbeitsgebieten einschließlich inhaltlicher Querbezüge visualisiert und durch neue Recherchemöglichkeiten effizient erschlossen werden. Private Annotationen ermöglichen dem einzelnen Benutzer sich seine eigene Sicht auf sein Fachgebiet aufzubauen. Der Wissensaustausch innerhalb von Fachgruppen wird durch öffentliche Annotationen unterstützt. Durch das aggregierte Expertenwissen entsteht eine Einsicht in das betreffende Gebiet, die das in den Dokumenten enthaltene Fachwissen übersteigt.

Derzeit existierende Systeme zur Navigation in Beziehungsgeflechten beschränken sich auf automatisch aus den Literaturdaten bzw. Volltexten extrahierbare Beziehungen wie die Zitierbeziehung oder Koautorenschaft. Erfahrungswissen, das sich nur aus der Lektüre der Literaturquellen in Kombination mit vorhandenem Expertenwissen ergibt, kann so nicht erfasst werden. Dieses Wissen kann in BibRelEx durch *beliebige* Beziehungen, die durch typisierte und annotierte Links definiert werden, eingebracht und für eine strukturbasierte Recherche genutzt werden. Eine Visualisierung der Wissensstruktur als gerichteter Graph dient zusätzlich zur inhaltlichen Navigation und bietet eine integrierte Darstellung des Fachwissens. Bisherige Systeme bieten nur eine grafische Darstellung festverdrahteter Beziehungsgeflechte. Dagegen ermöglicht BibRelEx eine individuelle Darstellung inhaltlicher Zusammenhänge nach benutzerdefinierten Kriterien. Zusätzlich ist in der Visualisierung eine interaktive Eingabe von Annotationen und Beziehungen möglich.

II

Danksagung

Die vorliegende Arbeit entstand während meiner Lehr- und Forschungstätigkeit als wissenschaftlicher Mitarbeiter an der Fenuniversität Hagen und an der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Dank gebührt vor allem meiner Doktormutter Prof. Dr. Anne Brüggemann-Klein und meinem Betreuer Prof. Dr. Rolf Klein, die mich mit der Gewährung großer Freiheit bei der Bearbeitung meines Forschungsthemas betreut haben.

Prof. Dr. Thomas Ottmann danke ich für seine Bereitschaft, das Zweitgutachten für diese Dissertation zu übernehmen.

Danken möchte ich weiterhin der Technischen Universität München für die finanzielle Förderung durch ein Promotionsstipendium.

Ich möchte mich auch bei allen meinen ehemaligen Kollegen bedanken. Sie haben mich bei meinen Tätigkeiten unterstützt, für ein angenehmes Arbeitsklima gesorgt und zum BibRelEx-System beigetragen. Davon ist insbesondere Dr. Elmar Langetepe zu nennen, der stets ein offenes Ohr für meine Fragen hatte und diese Arbeit sorgfältig durchgesehen hat.

Nicht zuletzt bin ich sehr dankbar für die viele Geduld, Nachsicht und Aufmunterung, die meine Freunde und Familie während der Fertigstellung dieser Arbeit für mich aufbrachten.

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Am Anfang jeder wissenschaftlichen Arbeit steht die gezielte Informationssuche und -beschaffung. Durch das Internet, insbesondere durch das *World-Wide Web* (WWW), stehen dafür neben den klassischen Bibliotheken zahlreiche Informationsquellen zur Verfügung. Durch die ständig wachsende Menge, Vielfalt an Daten und sehr unterschiedliche Qualität der Daten in diesen Quellen wird es allerdings zunehmend schwieriger, gezielt Informationen zu finden. Aufgrund der wohlbekannten Probleme (Abhängigkeit der Suchergebnisse von den gewählten Suchbegriffen, große Ergebnismengen, geringe Präzision) kann sich der Informationssuchende nicht sicher sein, ob die gefundenen Informationen die aktuellsten sind oder ob er andere, weitaus interessantere übersehen hat.

Ein Wissenschaftler sammelt im Laufe der Zeit zahlreiche wissenschaftliche Veröffentlichungen, aus denen er die für ihn relevanten Informationen herauszieht, bewertet und klassifiziert und sich so seine Sicht auf sein Fachgebiet aufbaut. Um später adäquat auf diese Arbeiten zugreifen zu können, muss er neben den bibliographischen Angaben zum korrekten Zitieren dieses Erfahrungswissen parat haben. Er wird sich Stichworte zum Inhalt, individuelle Anmerkungen und offene Fragen, die sich beim Studium der Arbeit ergeben haben, notieren. Neben der Suche nach Informationen ist also die Verwaltung der gefundenen Informationen von genauso großer Bedeutung.

Hinzu kommt, dass bei der exponentiell wachsenden Anzahl wissenschaftlicher Veröffentlichungen der einzelne Wissenschaftler kaum noch den Überblick über sein Themengebiet behalten kann. Für Nicht-Spezialisten aus anderen Gebieten stellt sich das Problem noch viel gravierender dar. Als Folge davon werden vorhandene Resultate neu entdeckt, und gute Lösungen finden

keine Anwendung, weil sie nicht bekannt sind. Hier benötigt der Informationssuchende das Wissen von Experten in dem jeweiligen Fachgebiet. Sie können Auskunft darüber geben, welche Arbeiten besonders wichtig sind, und kennen Zusammenhänge zwischen den Veröffentlichungen.

Zusammenfassend lässt sich sagen, dass der einzelne Wissenschaftler effiziente Verfahren benötigt, um gezielt Informationen zu finden, Wissen über Themenbereiche zu organisieren und sich mit Kollegen über Arbeitsergebnisse auszutauschen.

Um diesen Bedürfnissen gerecht zu werden, kann man sich unserer Meinung nach nicht mehr nur auf die traditionellen Recherchemethoden wie schlüsselwortbasiertes Suchen oder Navigieren entlang systematischer Struktur eines Klassifikationssystems stützen, sondern muss vielmehr den Weg hin zu einer wissensbasierten Recherche und Literaturverwaltung gehen. Dazu ist eine Anreicherung der Datenbasis mit Wissen notwendig, die eine einfache Wissenswiederverwendung ermöglicht.

Hierfür können inhaltsbasierte Beziehungen zwischen Dokumenten wie etwa die Zitierrelation hilfreich genutzt werden. Übersichtsarbeiten erkennt man beispielsweise daran, dass sie viele andere Arbeiten in einem Gebiet zitieren, wohingegen zentrale Arbeiten von vielen anderen Arbeiten zitiert werden. Auch thematisch verwandte Dokumente lassen sich mit Hilfe der Zitierrelation leicht bestimmen. Sie sind daran zu erkennen, dass sie in etwa zu denselben Dokumenten in Zitierrelation stehen.

Durch die Verwendung der Zitierrelation werden die Arbeiten nach Wissensgebieten angeordnet, ohne auf Titel oder Schlüsselworte zurückzugreifen. Damit werden zwei wesentliche Probleme, die bei der Informationssuche mit Hilfe der klassischen Retrievalmethoden auftreten, vermieden: die Wahl geeigneter Suchbegriffe und die Berücksichtigung des Kontextes.

Neben der Zitierrelation gibt es weitere wichtige inhaltliche Beziehungen zwischen Dokumenten, die nicht automatisch aus den Dokumenten gewonnen werden können, sondern sich nur durch die Lektüre ergeben (*Erfahrungswissen*). Beispiele hierfür sind *basiert auf*, *ist ähnlich zu*, *ist eine Anwendung von* oder *ist Folgearbeit zu*.

Solches Expertenwissen sollte in die Datenbasis selbst integriert werden. Dafür kommt nur ein kollaborativer Ansatz in Frage, denn um Vollständigkeit und Aktualität eines Datenbestandes zu erreichen, ist die Beteiligung vieler Experten erforderlich.

Wer aktiv in einem Wissensgebiet arbeitet, wird darüber hinaus seine eigene Sicht des Informationsraums durch private Information ergänzen wollen. Dies kann zum Beispiel durch Aufnahme weiterer Publikationen in den Bestand geschehen, die in den eigenen Arbeiten zitiert werden, oder durch subjektive Anmerkungen wie *Das Zitat von Dokument Y in Dokument X*

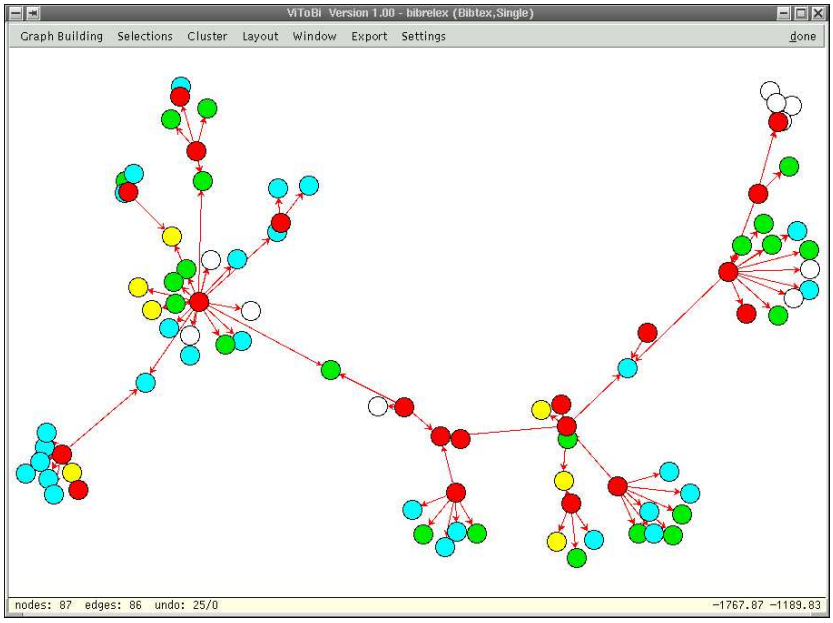
ist relevant oder *Dokument X enthält eine besonders gute Darstellung von Technik B*.

Eine Visualisierung der Wissensstruktur als gerichteter Graph (*Wissensgraph*) kann zusätzlich zur inhaltlichen Navigation dienen. Dabei repräsentieren die Knoten des Graphen die einzelnen Dokumente oder Themen (Cluster). Die Kanten symbolisieren die inhaltlichen Beziehungen wie etwa *zitiert* oder *ist Folgearbeit zu* zwischen den Dokumenten bzw. Themen. Diese Darstellung erlaubt eine umfassende Sicht auf Wissensgebiete und kann für eine effizientere Informationssuche genutzt werden. Beispielsweise sind diejenigen Publikationen, die besonders viele andere Arbeiten zitieren, in der Regel Übersichtsartikel und daher besonders geeignet, um sich einen schnellen Überblick zu verschaffen. Bei Verwendung geeigneter Layout-Verfahren lassen sich diese Übersichtsarbeiten leicht als zentrale Knoten mit vielen ausgehenden Kanten erkennen. Analog lassen sich wichtige Arbeiten in einem Gebiet als zentrale Knoten mit vielen eingehenden Kanten schnell lokalisieren. Die Geschichte der Lösung eines Problems lässt sich mit einem Blick erfassen, wenn man alle Paare derjenigen Arbeiten als gerichteten Graph darstellt, die das Problem behandeln und von denen die *eine* Folgearbeit der *anderen* ist. Die Abbildung 1.1 zeigt exemplarisch zwei verschiedene Darstellungen eines Zitiergeflechts in BibRelEx.

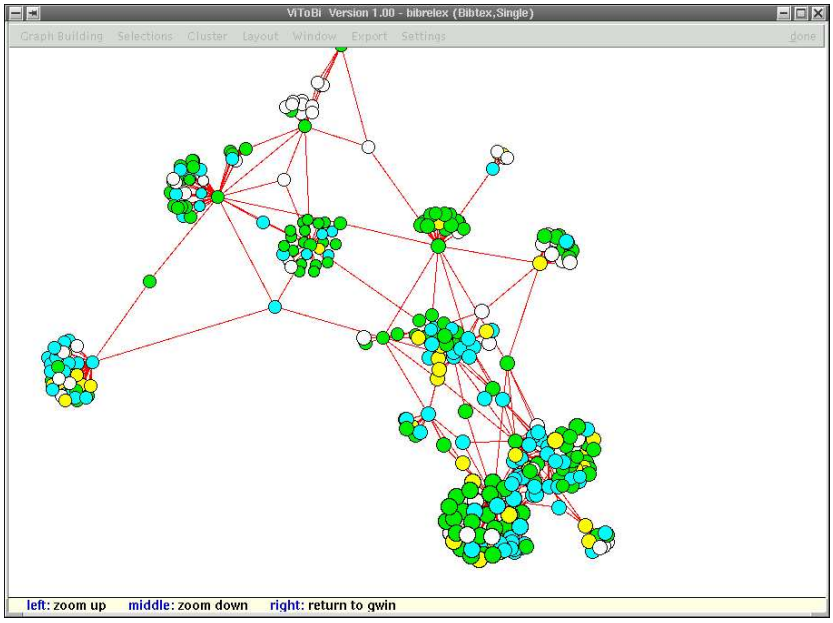
1.2 Ziel der Arbeit

Gegenstand dieser Arbeit ist die Anforderungsanalyse, die Modellierung, der Entwurf, die Realisierung und die Bewertung des interaktiven Systems *BibRelEx* (Bibliographic Relationship Explorer) zur Verwaltung von wissenschaftlicher Literatur und zur Informationsvisualisierung. Mit diesem System soll dem Anwender ermöglicht werden,

- effizient Informationen durch Kombination textueller und visueller Recherchemöglichkeiten unter Ausnutzung des strukturellen Wissens (inhaltliche Beziehungen) zu finden. Dabei soll es sowohl möglich sein gezielt Informationen zu finden als auch sich einen Überblick über ein Gebiet zu verschaffen,
- Gedanken, Ideen, Literatur und andere Quellen zu ordnen und miteinander zu verknüpfen,
- die Literatur in seinem Arbeitsgebiet zu verwalten und um zusätzliches Wissen zu ergänzen,
- einzelne Themen in ihrer Tiefe zu erschließen,



(a) 2D-Darstellung eines minimalen Kozitations-Spannbaums



(b) 3D-Darstellung des vollständigen Zitiergeflechts

Abbildung 1.1: Darstellungsarten eines Zitiergeflechts

- sich mit anderen Experten auszutauschen.

Unser primäres Ziel besteht dabei in dem Nachweis, dass die Aggregation von Expertenwissen, die Berücksichtigung von inhaltlichen Beziehungen zwischen Dokumenten und die Informationsvisualisierung wesentlich dazu beitragen, schnell und gezielt Informationen zu finden und zu verwalten. Dabei kommt es weniger darauf an, neue Algorithmen für die verschiedenen Teilprobleme in BibRelEx wie Layout, Clusterung oder Wissensdarstellung zu finden, als ein Tool bereitzustellen, das hierfür geeignete Techniken in einem System integriert. Wir wollen dem Benutzer ein echtes Werkzeug zur Verfügung stellen.

Darüber hinaus soll BibRelEx flexibel eingesetzt werden können, um beispielsweise verschiedene Algorithmen zu erproben, eine Anbindung an das WWW zu ermöglichen und verschiedene Datenbasen- und Formate nutzen zu können.

1.3 Ansatz

Diese Ziele werden in BibRelEx durch folgende Maßnahmen erreicht:

- Definition und Nutzung beliebiger Beziehungen (neben der Zitierrelation) durch typisierte und annotierte Links,
- Wissensanreicherung durch öffentliche und private Links und Annotationen,
- Erprobung innerhalb einer wissenschaftlichen Community,
- Konsistenzkontrolle,
- Qualitätssicherung der Beiträge durch geeignete Policies,
- Wissensaustausch durch periodische Aktualisierung der Datenbasis,
- dynamische graphische Darstellung von Beziehungsgeflechten mit vielfältigen Interaktions- und Navigationsmöglichkeiten,
- Einsatz effizienter Algorithmen, z.B. bei der Clusterung zur automatischen Bildung von Themengebieten,
- Flexibiltät durch konsequenten Einsatz von Entwurfsmustern,
- Schichtenarchitektur mit Kommandozeileninterface, das eine Anbindung an das WWW mit einer CGI-Schnittstelle ermöglicht,

- Wrapper zur Anbindung verschiedener Datenhaltungssystemen, z.B. BibTEX-Dateien, SQL-Datenbanken.

Der Anwender von BibRelEx hat also die Möglichkeit, eigene Verweise zwischen Dokumenten zu definieren und Annotationen anzubringen. Diese können privat oder öffentlich sein. Private Annotationen machen es möglich, einen individuellen Wissensraum zu gestalten. Durch Aggregation von öffentlichen Annotationen, von Experten beigesteuert, kann eine Einsicht in das betreffende Gebiet entstehen, die das in den Dokumenten enthaltene Fachwissen ergänzt.

Um eine Wissenserschließung zu ermöglichen, die weit über die reine Nutzung der Zitierrelation hinaus geht, können in BibRelEx *beliebige* Beziehungen durch typisierte und annotierte Links definiert und genutzt werden.

Wichtig war uns auch, die Tragfähigkeit des Ansatzes in einer realen Anwendung zu erproben. Dazu wird eine hinreichend umfangreiche Menge an Literatur benötigt, die aber einen dennoch überschaubareren Wissensraum bildet. Hier bietet sich der Einsatz in einer wissenschaftlichen Community an, da die Informationen, die innerhalb der Gruppe ausgetauscht werden, weitgehend selektiert und thematisch begrenzt sind. Wir haben uns für das Gebiet der Algorithmischen Geometrie entschieden, in dem durch freiwillige Arbeit vieler Wissenschaftler im Laufe der Jahre eine bibliographische Datenbank namens GeomBib entstanden ist. Näheres zu GeomBib findet sich in Abschnitt 3.3. Die Einschränkung auf ein einzelnes Fachgebiet ermöglicht uns – zusätzlich zur Zitierrelation – auch die inhaltlichen Beziehungen erfassen zu können, die auf dem Einbringen von Expertenwissen beruhen. Obwohl wir damit den Schwerpunkt auf eine spezielle Datenbasis eines Fachgebiets gelegt haben, wurde BibRelEx so entwickelt, dass es leicht auf andere Wissensgebiete und Datenbasen oder auch in digitalen Bibliotheken angewendet werden kann.

Um den Geometern den Einstieg in die Benutzung des Systems zu erleichtern, ist die Funktionalität der ihnen bekannten Tools bibview [116] und bibindex/biblook [54] integriert worden.

Zur Erreichung einer hohen Akzeptanz innerhalb der Community ist eine gewisse Qualitätssicherung notwendig. Hierfür haben wir zahlreiche Möglichkeiten der Eingabeunterstützung in BibRelEx vorgesehen. Sie helfen zu vermeiden, dass fehlerhafte Einträge in die Datenbasis eingebracht werden. Zusätzlich haben wir das Tool BibConsist entwickelt, dass Inkonsistenzen und doppelte Einträge in der Datenbasis erkennt. Dieses Tool ist vollständig in der Oberfläche von BibRelEx integriert und ist auch hilfreich bei der periodischen Aktualisierung von GeomBib.

Neben der rein formalen Korrektheit der Einträge in der Datenbasis muss auch eine inhaltliche Qualität der Beiträge, insbesondere der Annotationen, gewährleistet werden. Ansätze hierfür werden in Abschnitt 5.3 diskutiert.

Damit die Visualisierung bei der Literaturrecherche nutzbringend und zielgerichtet eingesetzt werden kann, legen wir bei der Darstellung der Beziehungsgeflechte in BibRelEx insbesondere Wert darauf, dass

- die Struktur des Wissensraumes leicht zu erkennen ist,
- die zeitliche Entwicklung von Wissensgebieten nachvollziehbar ist,
- eine Gesamtübersicht möglich ist,
- die Darstellung dynamisch und interaktiv ist,
- bei Veränderungen der anzuzeigenden Dokumentenmenge die *Mental Map* des Betrachters erhalten bleibt.

Die Struktur des Wissensraumes lässt sich leicht erkennen, wenn inhaltlich zusammenhängende Dokumente räumlich nah angeordnet werden. Hierfür bieten sich kräftebasierte Verfahren wie der *Spring Embedder* [49] zur Platzierung der Knoten in der Darstellung an. Er basiert auf einem physikalischen Modell, bei dem die Knoten des Graphen als elektrisch geladene Partikel, die sich gegenseitig abstoßen, betrachtet werden. Die Kanten des Graphen werden als Federn modelliert, die die durch sie verbundenen Knoten anziehen. Ausgehend von einer zufälligen Anordnung der Knoten im Raum strebt ein solches System einen stabilen spannungsarmen Zustand an. Die resultierende Darstellung hat den Vorteil, dass in Beziehung stehende Knoten räumlich nah zueinander angeordnet werden.

Auf Zitiergeflechte angewendet bedeutet dies, dass Cluster von Knoten Dokumente repräsentieren, die ähnliche Referenzen haben. Damit kann der Benutzer inhaltlich zusammenhängende Dokumente leicht erkennen. Eine ausführliche Beschreibung des Algorithmus findet sich in Abschnitt 3.2.1.

Der Spring Embedder führt in seiner ursprünglichen Form nur bei kleineren Graphen zu guten Resultaten. Bei großen Graphen müssen Heuristiken verwendet werden, um ein akzeptables Laufzeitverhalten zu erreichen. Für große Graphen bietet BibRelEx hier die Möglichkeit, den Graphen nach verschiedenen Kriterien zu clustern, bevor ein Spring Embedding ausgeführt wird, und somit den Graphen, auf den der Embedder anzuwenden ist, klein zu halten.

Gute Ergebnisse erreicht man auch mit dem GEM3D-Algorithmus von Bruß und Frick [28], der zusätzlich eine virtuelle Temperatur zur Justierung der Knotenverschiebung verwendet. Er kombiniert so den Spring Embedder

mit einem energiebasiertem Verfahren, dem sogenannten *Simulated Annealing*. Unter Verwendung weiterer Heuristiken erreicht man so insgesamt eine bessere Laufzeit und höhere Layoutqualität.

Für BibRelEx haben wir den GEM-Algorithmus um zusätzliche anziehende Kräfte erweitert, durch die gemeinsam referierte Dokumente zwischen den referierenden Dokumenten angeordnet werden. Auf diese Weise kann speziell für Beziehungsgeflechte die Zahl der Kantenüberkreuzungen reduziert werden und die Anordnung gibt besser die Beziehungen zwischen den Dokumenten wieder.

Um die zeitliche Entwicklung von Wissensgebieten zu untersuchen bietet sich eine hierarchische Darstellung nach dem Erscheinungsjahr an.

Die Einflüsse von Arbeiten innerhalb einer Dokumentenmenge lassen sich auch gut in einer Darstellung des minimalen Spannbaums basierend auf Kozitationsdistanzen erkennen [36, 37, 38]. Für BibRelEx ist uns wichtig nicht nur die Zitierrelation zu betrachten, sondern die Spannbaumdarstellung für beliebige Beziehungen zu ermöglichen.

Eine weitere Möglichkeit die Einflüsse von Arbeiten nachzuvollziehen bieten dynamische Darstellungen, die jederzeit um Anfrageergebnisse ergänzt werden können und die Veränderung des Layouts durch die neu hinzugekommenen Knoten inkrementell anzeigen.

Die inkrementelle Anzeige von Zwischenschritten trägt auch wesentlich dazu bei, dass der Benutzer nicht die Orientierung verliert. Ein Benutzer macht sich üblicherweise eine Vorstellung von der Organisationsstruktur der Datenbasis, indem er eine kognitive Repräsentation der Darstellung (Mental Map) aufbaut. Durch einen kontinuierlichen Übergang bei Änderungen in der Darstellung, kann der Benutzer durch Mitverfolgen der Knotenbewegungen seine Mental Map an die Änderungen anpassen und behält so den Überblick.

Wir haben eine Reihe von Visualisierungssystemen untersucht [163], um ein geeignetes System für die Realisierung von BibRelEx zu finden. Ein solches System sollte die dreidimensionale Darstellung von Beziehungsgeflechten unter Berücksichtigung der obigen Kriterien ermöglichen und komfortable Navigationsmöglichkeiten bieten. Neben diesen recht allgemeinen Anforderungen sind noch einige anwendungsspezifische Wünsche zu berücksichtigen. So sollte es möglich sein, Knoten und Kanten per Mausklick auszuwählen und zugehörige Informationen wie den bibliographischen Eintrag selbst und ggf. zugehörige Annotationen in einem Textfenster anzuzeigen.

Weiterhin soll das Visualisierungsystem einfach mit der Datenhaltungskomponente von BibRelEx (BibManage) zusammenarbeiten können, um die klassischen Retrievalmöglichkeiten, die BibManage bietet, mit der visuellen Erforschung des Informationsraums kombinieren zu können.

Leider hat es sich herausgestellt, das derzeit kein System existiert, dass alle unsere Anforderungen erfüllt. Wir haben uns daher für die Verwendung der C++-Bibliotheken LEDA (Library of Efficient Data Types and Algorithms) [3, 130] und AGD (Algorithms for Graph Drawing) [128] entschieden, mit denen wir einen großen Teil der oben beschriebene Funktionalität implementieren können. LEDA bietet uns notwendige Basisdatenstrukturen für Graphen, zahlreiche geometrische Algorithmen und Komponenten für die Erstellung einer Benutzerschnittstelle. Außerdem lässt LEDA sich erweitern. AGD basiert auf LEDA und bietet zusätzliche Algorithmen zum automatischen Zeichnen von Graphen.

1.4 Eigene Ergebnisse

BibRelEx ist ein sehr umfangreiches Projekt, das viele Teilbereiche der Informatik wie beispielsweise Information Retrieval, Hypertextsysteme, soziale Netze und Visualisierung berührt. Die vorliegende Arbeit stellt in diesem Sinne eine Querschnittsarbeit dar, die deutlich macht, was es alles an Einzelkomponenten in den Teilbereichen gibt und wie man diese in geeigneter Weise in einem System integrieren kann. Die Kombination unterschiedlicher Ansätze, wie sie in dieser Arbeit realisiert wurde, führte zu einem Prototypen, der den Informationssuchenden die Recherche erleichtert und dem Wissenschaftler ein Werkzeug zur Verwaltung seines Fachwissens an die Hand gibt. Der Prototyp von BibRelEx steht unter `http://web.informatik.uni-bonn.de/I/research/BibRelEx/BibRelEx.tar.gz` zum Download zur Verfügung.

Ein solches Projekt ist ohne den geballten Einsatz von „Manpower“ kaum in einem konkurrenzfähigen Zeitraum zu realisieren. Diese stand uns leider nicht zur Verfügung, so dass es uns nicht möglich war das Projekt schneller voranzutreiben. In der Zwischenzeit sind zahlreiche in Teilen ähnliche Projekte entstanden und aufgrund größerer Mitarbeiterzahlen schneller vorangekommen. Dies zeigt den hohen Wert unserer Idee und die Aktualität des Forschungsgebietes. Darüber hinaus ist BibRelEx das einzige Projekt, dass alle Teilbereiche Literaturverwaltung, Wissensaggregation und Visualisierung vereint. Gerade die Wissensaggregation unterscheidet BibRelEx deutlich von den anderen Projekten.

Im Folgenden wird eine Übersicht über die in dieser Dissertation vorgestellten Forschungsergebnisse gegeben. Aufgrund der eben diskutierten Probleme – der Aktualität des Forschungsgebietes und unserer geringen Manpower – kann es vorkommen, dass ähnliche Ergebnisse in verwandten Projekten bereits veröffentlicht wurden. Unsere Ergebnisse wurden unabhängig von diesen Projekten entwickelt. Wo Algorithmen übernommen wurden, ist

das an den jeweiligen Stellen in dieser Arbeit angemerkt.

- Entwicklung eines Verfahrens zur Konsistenzprüfung, das die gemeinsame Nutzung großer Literaturbestände erheblich vereinfacht;
- Entwicklung eines Konzepts zur Wissensaggregation basierend auf Annotationen und Links, das die kollaborative Nutzung des Wissens durch periodische Updates unterstützt;
- Erweiterung verschiedener Suchverfahren zur Nutzung inhaltlicher Beziehungen bei der Recherche;
- grafische Visualisierung von Beziehungsnetzwerken nach benutzerdefinierten Kriterien;
- Erweiterung verschiedener Layout-Verfahren, so dass diese speziell für Beziehungsgeflechte geeignet sind. Beispielsweise wurde durch die Verwendung zusätzlicher anziehender Kräfte beim GEM-Algorithmus eine Darstellung ermöglicht, die die Beziehungen deutlich besser lesbar wiederspiegelt. Darüber hinaus wurde die Spannbaumdarstellung auf beliebige Beziehungen erweitert. Alle Layout-Verfahren sind weitgehend parametrisiert worden, so dass der Benutzer die Darstellung seinen Informationsbedürfnissen anpassen kann;
- Entwurf und Implementierung eines integriertes Systems, das diese Verfahren (Konsistenzprüfung, Wissensaggregation, neue Suchverfahren basierend auf der Struktur des Informationsraumes, visuelle Exploration) Anwendern prototypisch zur Verfügung stellt;
- Erprobung des Prototypen mit dem Ergebnis, dass sowohl die Aggregation von Expertenwissen als auch die Visualisierung von Beziehungsgeflechten eine effiziente Recherche unterstützen.

BibRelEx wurde in Anchorage auf der Konferenz IFMIP 98 [23] und in London auf der Konferenz IV2000 [27] vorgestellt. Neben diversen anderen Veröffentlichungen [24, 25, 26] und der Programmdokumentation, z.B. [106, 107, 108, 110], sind im Rahmen des BibRelEx-Projektes 4 Diplomarbeiten [85, 147, 163, 172] entstanden.

1.5 Gliederung der Arbeit

In Kapitel 1, dieser Einleitung, wurde eine Übersicht über die in der Arbeit behandelte Problematik der effizienten Literaturverwaltung und Recherche

mit Hilfe von Wissensaggregation und Visualisierung gegeben. Es wurden die Ziele und Forschungsergebnisse dieser Arbeit vorgestellt.

Die Wissensaggregation erfolgt mit Hilfe von Annotationen und annotierbarer inhaltsbasierter Beziehungen, die mit Hilfe von Links realisiert werden. Eine gewisse Sonderstellung nimmt dabei die Berücksichtigung der explizit gegebenen Zitierrelation ein, die seit vielen Jahren Gegenstand intensiver Forschung ist. Eine effiziente Recherche wird durch linkbasierte Retrievalmethoden und der Visualisierung der entstehenden Beziehungsgeflechte unterstützt. Eine Analyse des Stands der Technik in diesen Bereichen – Berücksichtigung der Zitierrelation, linkbasierte Recherche, Annotationssystemen und Visualisierung – gibt das nächste Kapitel 2.

Anschließend werden in Kapitel 3 die Verfahren aus den Bereichen Zitatenanalyse und Visualisierung näher beschrieben, die in BibRelEx Anwendung finden und die Datenbasis GeomBib vorgestellt, an der wir unsere Ideen erprobt haben. In diesem Kapitel werden auch viele Fachbegriffe erklärt. Für Leser, die nicht so mit einem in der Arbeit behandelten Gebiet vertraut sind, empfiehlt es sich dieses Kapitel zuerst zu lesen.

Während die bisherigen Kapitel mehr allgemeine Aspekte des Projekts behandelt haben, werden in Kapitel 4 die Möglichkeiten von BibRelEx an konkreten Nutzungsbeispielen vorgestellt.

Kapitel 5 ist der Kernteil der Arbeit und stellt die implementationsunabhängigen Details von BibRelEx vor. Es lässt sich grob in zwei Teile gliedern. Im ersten Teil wird auf die Datenhaltung und text- und beziehungsbasierte Retrievalmethoden eingegangen. Hier werden die für das Projekt neu entwickelten Verfahren, beispielsweise für die Konsistenzprüfung in bibliographischen Datenbanken, vorgestellt. Es wird beschrieben, wie der Wissensaustausch innerhalb der Computational Geometry Community erfolgt. Das hier entwickelte Konzept der Dreiteilung der Datenbasis zur Ermöglichung einer automatischen Abwicklung der zyklischen Aktualisierung lässt sich auf beliebige Wissensgebiete übertragen und ermöglicht auch viel kürzere Update-Zyklen bis hin zur Online-Aktualisierung. Im Anschluß daran diskutieren wir einige Vorschläge für Richtlinien, die das gemeinschaftliche Arbeiten an einem dynamischen Datenbestand erfordert, um eine gewisse Qualität des Datenbestandes zu sichern. Nach diesen verwaltungstechnischen Details, stellen wir vor, wie in BibRelEx Annotationen und Beziehungen realisiert und für eine effiziente Recherche genutzt werden.

Die zweite Hälfte des Kapitels 5 behandelt die Visualisierungskomponente von BibRelEx. Eines der größten Probleme bei der Visualisierung umfangreicher Dokumentsammlungen ist die Handhabung der Informationsmenge. Für die „Lesbarkeit“ der graphischen Darstellung ist es wesentlich, dass nicht zu viele Details dargestellt werden. Andererseits darf durch die Reduktion des

Detaillierungsgrades nicht zu viel der strukturellen Information verloren gehen, damit weiterhin die wesentlichen Zusammenhänge zu erkennen sind und eine zielgerichtete Recherche im Informationsraum möglich ist. Insbesondere auf die in BibRelEx eingesetzten Methoden zur Komplexitätsreduktion und zur Rechercheunterstützung gehen wir in dem zweiten Teil des Kapitels 5 ein. Dazu gehören eine geeignete Vorauswahl der Darstellungsmenge, verschiedene Platzierungstechniken, geeignete Parametrisierung der in Kapitel 3.2 vorgestellten Algorithmen, dynamisches Layout bei weitgehender Erhaltung der Mental Map des Benutzers, Clusterung, Link-Aggregation und verschiedene Navigationstechniken.

Kapitel 6 stellt den Entwurf und die Implementierung des Systems BibRelEx vor. Das System wurde vollständig im Rahmen dieser Dissertation konzipiert und implementiert. Der konsequente Einsatz von Entwurfsmustern hat dabei aus BibRelEx ein flexibles Werkzeug sowohl für den Anwender zur Strukturierung und Recherche in großen Informationsräumen als auch für den Forscher zur Entwicklung neuer Algorithmen in diesem Bereich gemacht.

Die Beschreibung der Evaluierungsphase des Prototypen von BibRelEx und die daraus resultierende Bewertung des Systems erfolgt in Kapitel 7.

Kapitel 8 beschließt diese Arbeit mit einer Zusammenfassung der Ergebnisse und gibt einen Ausblick auf zukünftige Arbeiten.

In den Anhängen finden sich schliesslich ein Interaktionsbeispiel, um die Funktionalität der grafischen Benutzungsoberfläche von BibManage aufzuzeigen; eine ausführliche Beschreibung der Anfragesprache von BibManage, die sowohl textbasierte wie strukturbasierte Anfragemöglichkeiten umfasst, eine Darstellung des Aufbaus und der Verwendung von Konfigurationsdateien in BibRelEx, eine Legende zu den BibRelEx-Visualisierungen und einen Auszug der in dieser Arbeit zur Notation von Klassendiagrammen verwendeten *Unified Modeling Language* (UML).

Kapitel 2

Stand der Forschung

Kapitel 1 gab einen Überblick über die Probleme, die in dieser Arbeit behandelt werden: die Nutzung der inhaltlichen Beziehungen zwischen den Dokumenten zur effizienten Recherche, die Aggregation von Wissen in Literaturverwaltungssystemen durch Annotationen und die Visualisierung des Beziehungsgeflechts. Damit soll dem Informationssuchenden die Möglichkeit gegeben werden, sich im Wissensraum zu orientieren und mit seiner Struktur vertraut zu machen. Durch Kombination der Navigation im Beziehungsgeflecht und textbasierter Such- und Filtermöglichkeiten wird ihm die Möglichkeit einer effizienten zielgerichteten Informationssuche gegeben. Das Einbringen eigenen Wissens ermöglicht ihm, den Wissensraum nach eigenen Kriterien zu ordnen und sich mit anderen Wissenschaftlern auszutauschen.

Es existiert eine Vielzahl individueller Lösungen für einzelne dieser Aspekte wie Berücksichtigung der Zitierrelation, typisierte Links, strukturbasierte Recherche, Informationsaustausch mittels Annotationen und Visualisierung. Mit dem Projekt BibRelEx werden erstmals alle diese Techniken in einem System integriert. Viele der schon in Abschnitt 1.4 erwähnten Konkurrenzprojekte haben beim Projektstart von BibRelEx noch nicht existiert.

Im folgenden wird eine Übersicht über den Stand der Forschung gegeben. Sie enthält die wesentlichen zu Teilaspekten von BibRelEx verwandten Systeme und Projekte. Diese werden danach unterschieden, welche Teile der oben genannten Funktionalität sie anbieten. Dabei werden auch Querbezüge zu Fragestellungen in verwandten Gebieten wie etwa Soziale Netzwerke, Web-Engineering und Link-Analyse einbezogen. Sowohl das WWW als auch Zitiergeflechte sind im Prinzip Spezialfälle von sozialen Netzwerken. Dieses Beispiel zeigt, dass es zwischen den einzelnen Gebieten Überschneidungen gibt. Es lassen sich aber auch Ergebnisse aus einem speziellen Gebiet auf viel allgemeinere Probleme übertragen. Wegen der Breite der einzelnen Gebiete kann diese Übersicht nicht vollständig sein. Es wird vielmehr versucht, die

wesentlichen Forschungstendenzen zu verdeutlichen.

2.1 Berücksichtigung der Zitierrelation

Referenzen in wissenschaftlichen Arbeiten nachzugehen, war schon immer ein wesentlicher Teil systematischen Studierens. Es ist daher nicht überraschend, daß es verschiedene Systeme und Projekte gibt, die diese Aktivität unterstützen wollen.

Erste Ansätze, Zitier- und Textdaten für das Information Retrieval zusammen zu nutzen, finden sich in den Arbeiten am *Vektorraummodell* von Salton [164, 165, 166], die bereits in den 60er Jahren publiziert wurden. Er zeigte, dass sich Dokumente durch die Kombination aus Zitierdaten und Indextermen effektiver für den Vergleich mit Suchanfragen repräsentieren lassen als nur durch die Verwendung von Indextermen.

Amsler [5] hat als erster vorgeschlagen, eine Kombination von bibliographischer Kopplung und Kozitationsanalyse zu benutzen, um die Ähnlichkeit von Dokumenten zu bestimmen. Dabei verwendet er nur die Zitierdaten und berücksichtigt nicht den Text.

Aufbauend auf diesen Ergebnissen haben Bichteler und Eaton [12] gezeigt, dass sich die Präzision der Suchergebnisse verbessern lässt, indem man die textbasierten Anfrageergebnisse durch die Kombination von bibliographischer Kopplung und Kozitationsanalyse neu einordnet.

Die Zitatenanalyse selbst, vgl. Abschnitt 3.1, wurde Mitte der 50er Jahre von E. Garfield entwickelt [67] und im *Institut for Scientific Information* (ISI) in Philadelphia weiterentwickelt. Dieses Institut gibt seit 1963 den *Science Citation Index* (SCI) [72, 73] heraus. Der SCI enthält Literaturangaben aus allen Gebieten der Naturwissenschaften, Technik und Biomedizin. Er nennt zu jedem Dokument alle darauf gerichteten Literaturhinweise. Man kann außerdem fragen, in welchen Arbeiten ein Autor zitiert wird, und man kann zu einer Arbeit die Liste aller vom Autor verfassten Publikationen ausgeben. Am ISI wurde auch *SCI-Map* entwickelt, eine Software, die es Benutzern ermöglicht, innerhalb des Zitiernetzes zu navigieren. Ausgehend von einem Autor, einem Dokument oder Schlüsselworten erhält der Benutzer eine lokale Darstellung, die schrittweise um weitere Verweise erweitert werden kann. Die Lizenzkosten des Science Citation Index sind recht hoch. Leider deckt der Index nur einen Teil der wissenschaftlichen Journale ab (solche, deren Artikel häufig zitiert werden); Proceedings und Technische Berichte sind nicht enthalten. Dieser Nachteil wiegt umso schwerer, weil es für Benutzer nicht möglich ist, eigene Dokumente oder Verweise einzugeben.

Small hat Methoden zur Analyse der Struktur wissenschaftlicher Litera-

tur mit Hilfe von Zitiergraphen [178] und durch Studieren von Kozitationsmustern [74] vorgeschlagen.

Auf den ersten Blick könnte man meinen, das WWW selbst stelle bereits eine Lösung für das Problem der Recherche mit Hilfe der Zitierrelation oder allgemeiner von Querverweisen dar: Man richte für jedes wissenschaftliche Dokument eine Seite ein und realisiere die Querverweise durch Links. Dann verwende man ein Visualisierungswerkzeug für das WWW, z. B. *HyperSpace* [202], um die Beziehungsgeflechte darzustellen. Dieser Ansatz birgt jedoch mehrere grundsätzliche Probleme. Erstens ist nicht klar, wer die zu einem Dokument gehörende Seite einrichten und unterhalten soll. Der Idee des WWW entspräche eine verteilte Lösung, bei der etwa jeder Autor für die eigenen Dokumente verantwortlich ist. Hierbei wäre fraglich, wie die erforderliche Kontinuität garantiert werden soll. Zweitens sind die Links im WWW gerichtet. Es ist also nicht direkt möglich, alle Arbeiten zu finden, die Verweise auf eine gegebene Arbeit enthalten; hierzu ist vielmehr eine aufwendige Vorverarbeitung notwendig. Drittens soll die Visualisierung von inhaltlichen Beziehungen mit klassischen Operationen wie der Suche nach Autoren oder Schlüsselwörtern kombinierbar sein. Dies setzt die Einhaltung von global akzeptierten Standards beim Einrichten der Seiten für die Dokumente voraus.

Dieser Ansatz verdeutlicht aber zumindest, dass die Konzepte Literaturrecherche mittels Weiterverfolgen von Hypertextlinks bzw. Zitierverweisen sehr ähnlich sind und sich Lösungsansätze teilweise übertragen lassen.

Zwei Systeme, die Hypertext-Links verwenden, um die Zitierbeziehung nachzubilden, sind das *Hypertext Bibliography Project* von Jones [89] und die *Universal Citation Database* von Cameron [30]. Daneben gibt es eine Reihe von WWW-basierten Zitierindizes, die sich mit dem automatischen Indexaufbau (*Autonomous Citation Indexing* (ACI)) befassen, z.B. der *ResearchIndex* (ehemals *CiteSeer*) [13], das *Open Journal Project* [84] und das Projekt *Clever* [33]. BibRelEx unterscheidet sich von diesen Projekten insofern, dass unser Schwerpunkt die effektive Nutzung der Zitierinformation ist und nicht die automatische Erzeugung dieser Information. Man könnte natürlich später beides nutzbringend miteinander verknüpfen. Darüber hinaus basiert BibRelEx auf dem flexibleren Konzept beliebiger Beziehungen zur Wissensaggregation.

Neben der Zitierbeziehung gibt es noch andere automatisch extrahierbare Beziehungen, die hilfreich bei der Navigation in großen Literaturbeständen eingesetzt werden können. Einige von solchen Systemen, die über die reine Nutzung von Zitiernetzen hinausgehen, betrachten wir im folgenden Abschnitt.

2.2 Autorennetzwerke

Aus den Literaturdaten oder Volltexten von Literaturquellen lassen sich weitere Beziehungen automatisch extrahieren. Hierzu zählen beispielsweise auf Koautorenschaft oder Schlüsselworten basierende Beziehungen. Sie ermöglichen die Beantwortung einer Reihe weiterer für die Recherche nützlichen Fragestellungen, wie *Welche Autoren verwenden das Schlüsselwort W?* oder *Welche Koautoren hat ein Autor A?*. Diese Fragen werden häufig verwendet, um weitere Literatur zu einem Thema zu finden.

Ein solches Autorennetzwerk bietet die *Trierer Informatik-Bibliographie* DBLP [114]. Zu jedem Autor kann man eine Liste seiner Veröffentlichungen abfragen und ist eine Navigation zu den Koautoren möglich. Zur Verbreiterung der Datenbasis hat Ley [114] Teile der Datenbestände mit dem oben bereits erwähnten *Hypertext Bibliography Project* ausgetauscht.

Basierend auf bibliographischen Daten der DBLP ist die *ACM SIGMOD Anthology* [115], eine digitale Volltext-Bibliothek für das Forschungsgebiet Datenbanksysteme, aufgebaut worden. Als Besonderheit werden in der Anthology alle Literaturreferenzen sowohl *aus* einer Publikation als auch *auf* eine Publikation aufgelistet. Darüber hinaus sind Links auf Hompages der Autoren und von themenbezogenen Konferenzen integriert. Die notwendigen Informationen werden in gemeinschaftlicher Aktivität vieler Verleger und Einzelpersonen zusammengetragen.

Ein anderer Ansatz ist, solche Beziehungen automatisch aus den Literaturdaten bzw. Volltexten zu extrahieren. Diesen Ansatz verfolgt das *MyView*-System [83, 201]. Der Schwerpunkt von *MyView* liegt dabei in der Integration von verschiedenen verteilten und heterogenen Informationsquellen. Im Gegensatz zu BibRelEx können aber nur automatisch extrahierbare Beziehungen genutzt werden.

Die hier beschriebenen Systeme bieten nur die Möglichkeit der Hypertext-Navigation, eine grafische Visualisierung der Beziehungsnetzwerke wird nicht unterstützt. Systeme, die die grafische Visualisierung der Autoren-Koautoren-Beziehung ermöglichen, werden in Abschnitt 2.6 beschrieben. Weder bei diesen Systemen noch bei einem der in diesem Abschnitt genannten Systemen hat der Benutzer die Möglichkeit eigene Beziehungen zu definieren oder Notizen zuzufügen.

Eine Möglichkeit explizit beliebige Beziehungen zwischen Dokumenten zu realisieren, bietet die Verwendung typisierter Links, deren Stand der Forschung wir im folgenden Abschnitt kurz reflektieren.

2.3 Typisierte Links

Typisierte Links werden schon lange in Hypertextsystemen eingesetzt, um Informationen zu strukturieren. Bereits 1983 schlägt Trigg [192] als einer der Ersten die Verwendung von typisierten Links zur kooperativen Literaturkritik in Hypertextsystemen vor. In seinem System *TextNet* unterscheidet er zunächst zwischen normalen Links und Kommentarlinks. Normale Links dienen zur Verbindung zwischen Dokumenten, wobei er darunter nicht nur reine Navigationslinks versteht, sondern auch Zitate und andere inhaltliche Zusammenhänge wie Verallgemeinerung oder Weiterentwicklung. Kommentarlinks dienen der Bewertung von Dokumenten, z.B. trivial oder irrelevant. Insgesamt unterscheidet er 85 Linktypen. Durch diese umfassende Vorgabe soll eine Verwirrung des Benutzers oder des System durch eigenmächtig eingeführte Linktypen vermieden werden.

Neben *TextNet* gibt es noch weitere Systeme, z.B. *SEPIA* [184], den *Debate Browser* [148] oder *MUCH* [197], die die Linktypen fest vorgeben. Andere Systeme, wie *MacWeb* [139] oder *NoteCard* [78] erlauben beliebige Linktypen oder erweiterbare Mengen von Linktypen. Dies ermöglicht eine größere Flexibilität, birgt aber auch die Gefahr inkonsistenter Linktypen [156].

Das Internet-System *HyperWave* [40, 127] (siehe auch Abschnitt 2.5) bietet ebenfalls einige Verweistypen für die verwalteten Dokumente. *HyperWave* basiert auf einer Dokumentdatenbank, in der die Dokumente hierarchisch angeordnet sind. Die angebotenen Verweistypen spiegeln leider nur diese hierarchischen Strukturen wieder.

Berners-Lee [10] sieht die Zukunft des WWW in einem *semantischen Netz*. Dazu wurden am *World Wide Web Consortium* (W3C) in den letzten Jahren u.a. ein Standard zur Formulierung von Metadaten entwickelt. Durch *Resource Description Framework* (RDF) sollen Webinhalte nicht nur maschinenlesbar, sondern auch maschinenverstehbar werden. Das Fundament des Semantischen Webs ist die *eXtensible Markup Language* (XML). Die *XML Linking Language* (XLink) erweitert die Möglichkeiten von XML um die Funktionalität von Links. Allerdings steht auch für XLink eine Lösung des Problem der Festlegung geeigneter Linktypen noch aus [146][1].

Auch der ISO-Standard *Topic Maps* [87], der die Beschreibung semantischer Informationen mittels XML erlaubt, legt (absichtlich) keine vorbestimmte Semantik für Konzepte oder Relationen fest. Topic Maps sind damit offen für beliebige semantische Modelle, auf die sich Nutzergruppen in Anwendungsbereichen einigen müssen. Auch wenn die Idee von BibRelEx,

[1] In [146] findet der interessierte Leser auch eine sehr ausführliche Darstellung der Historie über die Verwendung von Linktypen in Hypertextsystemen.

semantische Netze in Dokumentenmengen aufzubauen und zur Recherche zu nutzen, in die selbe Richtung geht wie Topic Maps, stellen diese beim derzeitigen Stand keine Lösung für BibRelEx dar. Topic Maps selbst sind keine grafischen Darstellungen im Sinne von Karten, eine entsprechende Visualisierung zur Navigation im Wissensraum müsste also nach wie vor implementiert werden. Darüber hinaus gibt es bisher keine Anfragesprache für Topic Maps. Ebenso muss auch noch die Erstellung der Topic Maps entsprechend unterstützt werden.

2.4 Strukturbasierte Recherche

Die im vorherigen Abschnitt beschriebenen typisierten Links können zur strukturbasierten Recherche genutzt werden. Obwohl bereits die *Hypertextmodelle Dexter* [79] und *HyTime* [46] in der Funktionalität mit XLink vergleichbar waren, bieten bislang nur wenige Hypertextsysteme strukturbasierte Anfragemöglichkeiten.

Grønbæk hat das *Open Hypermedia Interchange Format* (OHIF) vorgestellt, das XLink ähnlich ist, aber über ein reicheres Datenmodell verfügt. Mit dem *Webvise Open Hypermedia System* von Hansen u. a. [80] können OHIF Dokumente dargestellt werden. *Webvise* extrahiert die Metadaten, die in einem Dokument gespeichert werden und zeigt sie in einem separatenen Anwendungsfenster an. Es verfügt über ein Typsystem für Links und bietet die Möglichkeit Dokumente zu annotieren. Mit Hilfe eines einfachen Suchdialogs kann der Benutzer nach Links bestimmter Typen suchen. Darüber hinaus gibt es keine Analysemöglichkeit der Linkstruktur.

Dagegen gibt es eine Reihe von Projekten, die die Linkstruktur des Web verwenden, um die Relevanz von Web-Seiten zu analysieren. Das einfachste Ranking-Verfahren für Web-Seiten, das die Linkstruktur auswertet, ist das *Backlink Count*. Es basiert auf der Annahme, dass Seiten auf die viele andere Seiten verweisen wichtiger sind als solche auf die kein oder nur wenige Links existieren. Ein wesentlicher Nachteil des Backlink Counts ist, dass das gesamte WWW durchsucht werden muss, um alle Links zu einer Seite (Backlinks) zu finden. Daher wird von Crawlern die Anzahl der Backlinks geschätzt, indem sie nur die Links von Seiten zählen, die sie schon mal besucht haben. Außerdem können Betreiber von Webseiten Backlink Count einfach hintergehen, indem sie Pseudoseiten einrichten, die auf ihre Webangebote verweisen.

Der *HyperSearch*-Algorithmus von Marchiori [123] bestimmt die Relevanz einer Seite p aus der Summe der textbasierten Relevanz der von p aus erreichbaren Seiten, gewichtet mit einem Dämpfungsfaktor, der exponentiell mit der Linkdistanz abnimmt. Um nicht das gesamte Web durchsuchen zu

müssen, betrachtet Marchiori dabei nur Seiten innerhalb einer maximal noch nachzuverfolgenden Linkdistanz.

Diese Idee der Informationsausbreitung über Links wird von Page u. a. [150] erweitert. Ihr *PageRank*-Algorithmus verwendet ausschließlich die Linkstruktur zur Relevanzbestimmung, während Kleinbergs *Hypertext-Induced Topic Search* (HITS) [94, 95] zunächst noch den Inhalt der Webseiten berücksichtigt, um mittels einer Suchanfrage eine Startmenge für die Linkanalyse zu erhalten.

Bharat und Henzinger [11] kombinieren die Linkanalyse nach dem HITS-Verfahren mit der Textanalyse nach dem Vektorraummodell von Salton [167]. Sie vermeiden so einige für HITS typischen Probleme wie das Abdriften der Suchergebnisse von der eigentlichen Anfrage.

Dean und Henzinger [43] verwenden heuristische Verbesserungen aus [11] und [32], um ähnliche Web-Seiten zu finden. Alternativ geben sie einen auf Kozitation basierenden Algorithmus an.

Lu u. a. [118] verwenden einen ähnlichen Algorithmus wie Dean und Henzinger [43] und wenden diesen auf die Zitierrelation an. Ihr Ansatz die Ähnlichkeit von Dokumenten über den Zitiergraph zu bestimmen, ist ähnlich dem in BibRelEx verwendetem Ähnlichkeitsmaß basierend auf der Kozitation. Allerdings hatten wir die Idee wesentlich früher, und unsere Idee geht weit über die Ähnlichkeitsbestimmung hinaus. So berücksichtigen Lu u. a. [118] beispielsweise keine anderen Linktypen und bieten keine Visualisierung des Zitiergraphen.

Pitkow und Pirolli [154] clustern Web Seiten ebenfalls mittels Kozitationsanalyse.

Terveen und Hill [189] verwenden *n-Clan-Graphen*, um Gruppen von verwandten Web Seiten zu finden. Ein n-Clan-Graph ist ein gerichteter Graph, in dem jeder Knoten mit jedem anderen Knoten über einen Pfad der maximalen Länge n verbunden ist und alle Pfade nur über Knoten innerhalb des Clans verlaufen. Mit Hilfe eines heuristischen Verfahrens bestimmen sie einen lokalen Clan-Graphen ausgehend von einer Menge relevanter Seiten zu einem Thema, die der Benutzer vorgeben muss. Diese Konstruktion tendiert dazu, irrelevante Seiten auszufiltern und zusätzliche relevante Seiten zu finden. Der resultierende Graph wird im sogenannten *Auditorium View* angezeigt, in dem die Seiten Reihe für Reihe in Halbkreisen um ein Zentrum angeordnet werden, wobei die Relevanz der Seiten von innen nach außen sinkt.

Mukherjea u. a. [137] verwenden *Pre-Trees* um Hierarchien in der Linkstruktur des Webs zu visualisieren. Pre-Trees sind verallgemeinerte Bäume, in denen es eine Wurzel gibt, aber die Kinder keine Bäume zu sein brauchen, sondern beliebige Graphen sein können. Diese *Zweige* dürfen untereinander

nicht mit Links verbunden sein. Mukherjea u. a. [137] bestimmen die Pre-Trees durch eine Kombination von Text- und Linkanalyse.

Alle diese Verfahren eignen sich von der Konzeption her nicht für die eigentliche Bewertung von Seiten bezüglich spezifischer Suchanfragen, sondern ausschließlich dazu, zentrale Seiten zu einem Thema zu finden. Sie bieten sich damit für den automatischen Aufbau von Verzeichnissen an.

Der automatischen Klassifikation von WWW-Seiten anhand ihrer Linkstruktur und der thematischen Anordnung von Dokumenten anhand der Zitierrelation liegen die gleichen Ideen zugrunde (*interessante Seiten sind gut vernetzt* vs. *wichtige Arbeiten werden häufig zitiert*). BibRelEx wurde völlig unabhängig von den WWW-Aktivitäten entwickelt. Dass die Ansätze sich jedoch sehr ähneln und auch die Ergebnisse vergleichbar sind, zeugt von der Leistungsfähigkeit und Generalisierbarkeit der zugrundeliegenden Idee, strukturelle Zusammenhänge für die Recherche nutzbringend zu verwenden. Die vorgestellten Systeme zur Analyse der Linkstruktur verwenden allesamt die vorhandene Link-Struktur; keines erlaubt – im Gegensatz zu BibRelEx – den Benutzern, direkt eigene Beziehungen zuzufügen. Unsere Arbeit geht über die oben beschriebenen Arbeiten noch hinaus, indem wir schon frühzeitig eine Visualisierung des Beziehungsgeflechts vorgeschlagen haben, als auch dadurch, dass unsere Arbeit unseres Wissens nach das einzige System ist, das zusätzlich auch noch die Einbringung und den Austausch von Expertenwissen mit Hilfe von Annotationen ermöglicht.

Wegen ihrer Bedeutung für BibRelEx betrachten wir zunächst den *PageRank*-Algorithmus, den HITS-Algorithmus und seine Erweiterung für den *Automatic Resource Compiler* genauer, bevor wir uns anschließend den Annotationssystemen zuwenden.

PageRank

Der von Brin und Page entwickelte *PageRank*-Algorithmus [150] stellt eine Erweiterung des Backlink Count Verfahrens dar. Auch *PageRank* geht von der Annahme aus, dass Seiten mit vielen Backlinks wahrscheinlich einen hohen Interessantheitsgrad besitzen, bewertet die Links aber je nach referierender Seite unterschiedlich. Die Relevanz einer referierenden Seite wird dabei rekursiv aus den gewichteten Links zu ihr berechnet. Zusätzlich wird ein Dämpfungsfaktor d verwendet, der miteinbezieht, dass man beim Verfolgen von Links nur eine gewisse Zeit einem Thema folgt. Gibt es n Seiten $P_1, \ldots, P_n$, welche einen oder mehrere Links auf eine Seite P gesetzt haben und sei c_i die Anzahl der aus Seite P_i ausgehenden Links, dann berechnet

sich der PageRank der Seite P wie folgt:

$$Rank(P) = (1 - d) + d \sum \frac{Rank(P_i)}{c_i}$$

Diese Werte können während des Indizierens iterativ berechnet werden.

Der Dämpfungsfaktor d kann bei entsprechender Variation auch zur Realisierung eines komplexeren Reihungskriteriums, wie etwa zur Personalisierung auf einzelne Anwender, eingesetzt werden.

Die Suchmaschine *Google* hat den *PageRank*-Algorithmus übernommen. Da Dokumente oftmals durch den *Anker*-Text[2] besser beschrieben werden, als durch den Dokumenteninhalt, assoziiert *Google* außerdem den Anker-Text eines Links mit dessen Zieldokument. So können auch Bilder, Programme, Datenbanken und andere, von textbasiert arbeitenden Suchrobotern nicht erfaßbare Dokumente indiziert werden.

Dokumente werden von *PageRank* nur erfasst, wenn auf sie verwiesen wird. Da aber gerade neue Dokumente dem Benutzer oftmals noch nicht bekannt sind, gibt es keine oder nur wenige Links zu neueren Dokumenten. Als Nachteil von *Google* ist daher zu sehen, dass neuere Inhalte von der Suchmaschine unbeachtet bleiben.

Kleinberg [94] hat zwei weitere Probleme von *Google* beschrieben. Erstens gibt es häufig Seiten, die hochgradig relevant für eine Anfrage sind, obwohl sie keinen der Anfrageterme enthalten. Dieser Fall tritt beispielsweise bei der Anfrage „search engines“ auf. Bekanntermaßen sind die Seiten www.yahoo.com oder www.netscape.com relevant für diese Anfrage, enthalten aber nicht die Anfrageterme und werden daher auch nicht von *Google* gefunden. Das zweite Problem kann ebenfalls anhand dieser Seiten aufgezeigt werden. Da sehr viele Seiten auf Seiten wie www.yahoo.com oder www.netscape.com verweisen, kann man davon ausgehen, dass *PageRank* sie hoch bewertet. Wenn eine Anfrage gestellt wird, deren Terme diese Seiten enthalten, z.B. „Automobil“, werden diese als hochgradig relevantes Ergebnis geliefert, obwohl sie zu der eigentlichen Bedeutung der Anfrage keinen erkennbaren Bezug haben. Die ausschließliche Bewertung der Relevanz über eingehende Links gewährleistet also keine ausreichende Balance zwischen Relevanz und Popularität.

Hypertext-Induced Topic Search

Jon M. Kleinberg unterscheidet anhand der Linkstruktur des WWW zwei Arten von Seiten, siehe Abbildung 2.1. Eine Seite wird als Hub für eine Anfrage Q bezeichnet, wenn sie viele Links auf Seiten enthält, welche für Q

[2]Der Anker-Text ist der Text, durch den in einem Hypertextdokument der Anfangspunkt eines Links (=Anker) dargestellt wird.

relevant sind. Eine Seite wird als Authority für eine Anfrage Q bezeichnet, wenn viele Hubs für Q auf sie verweisen. Er geht davon aus, dass zwischen Authorities und Hubs folgende Wechselbeziehung besteht: Seiten mit vielen eingehenden Links, insbesondere von Hubs, sind vermutlich besonders relevante Authorities und Seiten mit vielen ausgehenden Links, insbesondere auf Authorities, sind vermutlich besonders relevante Hubs.

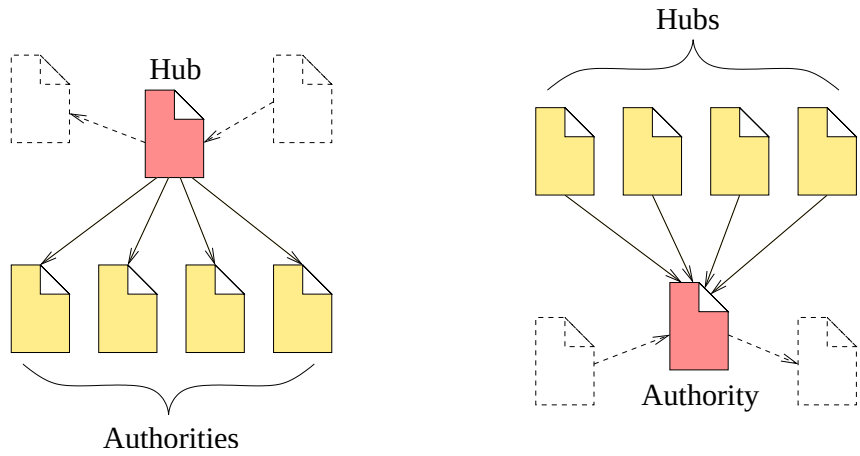


Abbildung 2.1: HITS-Algorithmus: Hub- und Authority-Seiten

Diese Beziehung hilft genau die oben beschriebenen Probleme von PageRank zu vermeiden. Kleinberg nutzt sie in seinem *Hypertext-Induced Topic Search* (HITS)-Algorithmus.

Im ersten Schritt des HITS-Algorithmus wird zunächst eine kleine Teilmenge des WWW bestimmt, die möglichst viele für die Anfrage relevante Authorities enthält. Dazu wird eine herkömmliche Textanfrage an eine indexbasierte Suchmaschine wie beispielsweise AltaVista gerichtet. Aus der Trefferliste werden die t höchst-bewerteten Treffer in die *Ursprungsmenge* übernommen. Die so gewonnene Ursprungsmenge enthält viele für die Anfrage relevante Seiten, aber nicht alle guten Hubs bzw. Authorities.

Daher wird im zweiten Schritt die Ursprungsmenge um die Seiten erweitert, die von den Seiten der Ursprungsmenge referenziert werden. Weiterhin werden je Seite der Ursprungsmenge maximal d Seiten, welche auf diese Seite verweisen, hinzugefügt. In der so entstandenen *Basismenge* werden noch Verweise innerhalb derselben Domäne entfernt, da diese häufig nur Navigationshilfen darstellen.

Im dritten Schritt werden die Seiten der Basismenge bezüglich ihrer Eignung als Authority oder Hub gewichtet. Dabei wird nur noch die Linkstruktur betrachtet und nicht mehr der Inhalt der Seiten. Da die Definition rekursiv ist (eine Seite ist von hoher Güte, wenn viele Seiten von hoher Güte auf sie

verweisen), verwendet der Algorithmus iterative Verfahren um die Gesamtqualität einer Seite zu bestimmen.

Kleinberg verwendete $t = 200$ und $d = 50$ in seinen Experimenten. Bei diesen Werten erreichte die Basismenge eine Größe von 1000 bis 5000 Seiten und konvergierte der Algorithmus nach 10-50 Iterationen [69, 94].

Der HITS-Algorithmus unterscheidet sich von *PageRank* in drei wesentlichen Punkten: Erstens unterscheidet er anhand der ein- und ausgehenden Links zwischen Übersichtsseiten (Hubs) und Inhaltsseiten (Authorities). Zweitens erfolgt die Gewichtung der Seiten dynamisch unter Berücksichtigung der Anfrage und nicht global und unabhängig von der Anfrage. Drittens benötigt HITS für seine Berechnungen jeweils nur einen relativ kleinen Ausschnitt des Webs.

Allerdings hat der ursprüngliche HITS-Algorithmus auch einige Probleme. Beispielsweise wertet er eine Seite zu stark als Authority, wenn viele Seiten eines Autors/einer Domäne diese Seite referenzieren. Analog wird eine Seite zu stark als Hub gewertet, wenn diese viele Seiten eines Autors/einer Domäne referenziert. Durch eine Kantengewichtung mit der inversen Zahl von Seiten des Autors bzw. der Domäne vermeiden Bharat und Henzinger [11] dieses Problem.

Ein weiteres Problem ist das Abdriften von der Anfrage, das vor allem dann auftritt, wenn die initiale Anfrage zu spezifisch ist. In diesem Fall gibt es evtuell nur wenig wirklich relevante Seiten für die Anfrage, so dass die zu dieser Anfrage konstruierte Basismenge aufgrund der Verlinkung eng verlinkte Seiten ohne Bezug zur Anfrage enthält. Diese Seiten werden häufig vom HITS-Algorithmus aufgrund ihrer (korrekten) Verlinkung als gute Authority oder Hub gewertet, obwohl sie zu der eigentlichen Bedeutung der Anfrage keinen erkennbaren Bezug haben. Dieses Abdriften von der eigentlichen Anfrage vermeiden Bharat und Henzinger [11], indem sie die textbasierte Relevanz bzgl. der Anfrage ebenfalls als Kantengewicht einbringen. Damit wird außerdem der Einfluss von automatisch erzeugten Links (z.B. Werbebanner) deutlich reduziert.

Weiterhin haben Bharat und Henzinger [11] ausgehend von der Annahme, dass nicht alle Seiten einen gleich starken Einfluss auf das Ergebnis des HITS-Algorithmus haben, mit Hilfe von *Partial Content Analysis* (PCA) die Kosten der textbasierten Analyse deutlich reduziert.

Der erweiterte Algorithmus [11] hat allerdings wie der ursprüngliche Algorithmus das Problem recht langer Retrievalzeiten, die aufgrund des Anfragespezifischen Vorgehens bei jeder Anfrage anfallen. Die Berechnungen von PageRank sind zwar ähnlich aufwendig, fallen aber nur einmal bei der Indizierung an. Eine sinnvolle Anwendung von Kleinbergs Algorithmus ist daher eher im automatischen Aufbau von Katalogen zu sehen und nicht die Bear-

beitung genereller Suchmaschinen-Anfragen.

Chakrabarti u. a. [31] haben zwei weitere Probleme des HITS-Algorithmus beschrieben. Wenn innerhalb einer Seite verschiedene Themengebiete behandelt werden, dann verweisen die ausgehenden Links auf Seiten verschiedener Themengebiete abhängig von ihrer Position auf der ursprünglichen Seite. Wenn diese Seite viele ausgehende Links enthält, bekommt sie im HITS Algorithmus ein hohes Gewicht als Hub zugewiesen, welches ein hohes Gewicht der referenzierten Seiten als Authority nach sich zieht, unabhängig von ihrer Relevanz bezüglich der ursprünglichen Anfrage.

Das zweite Problem, das ebenfalls in [69, 94] behandelt wird, ist die *Themenverallgemeinerung*. Ist das Suchthema zu eng gefasst, liefert HITS häufig gute Resultate für allgemeinere Themen. Beispielsweise liefert die Anfrage „mango fruit“ viele Seiten zu dem allgemeineren Thema „fruit“.

Die Probleme von PageRank und HITS zeigen, dass ein reiner linkbasierter Ansatz diverse Nachteile hat.

Aus diesen Gründen ist der HITS Algorithmus in dem nachfolgend beschriebenen *Automatic Resource Compiler* (ARC) weiter verfeinert worden. Eine ausführliche Beschreibung weiterer Modifikationen des HITS-Algorithmus findet man in [124].

Automatic Resource Compiler

Chakrabarti u. a. [33] haben den HITS-Algorithmus für den *Automatic Resource Compiler* (ARC) des Projekts Clever (Suchmaschine) von IBM erweitert, indem sie den Text in der Umgebung der Links bei der Berechnung der Hub- und Authority-Gewichte mit einbeziehen. Der ARC-Algorithmus stimmt weitgehend mit dem HITS-Algorithmus überein. Verändert wurde die Erweiterung der Ursprungsmenge zur Basismenge. In ARC werden auch Seiten mit in die Basismenge aufgenommen, die mit einer Linkdistanz von 2 zu erreichen sind. Dies wird dadurch erreicht, dass der Schritt 2 - Erweiterung der Ursprungsmenge zur Basismenge - des HITS-Algorithmus zweimal durchgeführt wird. Weiter geht bei der iterativen Berechnung der Hub- und Authority-Gewichte die Anzahl der Anfrageterme im Text in der Umgebung des Links mit ein. Dies beruht auf der Annahme, dass der Text in der Umgebung eines Links zu einer anderen Seite den Inhalt dieser Seite beschreibt. Man kann insbesondere dann davon ausgehen, dass eine Seite eine gute Authority-Seite ist, wenn der Text in der Umgebung des auf sie verweisenden Links auf einer gute Hub-Seite das Thema treffend beschreibt. Als Umgebung werden 50 Bytes vor und nach dem Link verwendet. Dieser Wert wurde experimentell ermittelt, indem die Vorkommenverteilung des Terms

„Yahoo“ in der Umgebung des Links „http://www.yahoo.com“ in 5000 Seiten überprüft wurde.

Das ARC-Verfahren wurde von Chakrabarti u. a. [33] auch experimentell getestet und zeigte sich dabei qualitätsmäßig vergleichbar mit Infoseek und wurde etwas schlechter bewertet als Yahoo. ARC hat wie HITS die Tendenz zur Generalisierung bei zu spezifischen Anfragen. Durch die Berücksichtigung der Suchworte bei der Hub- und Authority-Gewichtung bleibt die Suche jedoch fokussierter.

2.5 Annotationen

Die im letzten Abschnitt vorgestellten Systeme zur Analyse der Linkstruktur verwenden allesamt die vorhandene Link-Struktur; keines erlaubt den Benutzern, eigene Informationen zuzufügen. BibRelEx bietet zusätzlich die Möglichkeit, Erfahrungswissen in Form von Annotationen zu externalisieren und anderen Anwendern zugänglich zu machen. Dieser Ansatz ist relativ neu und wird bisher nur in wenigen Systemen berücksichtigt, von denen wir zwei, *Ariadne* und *HyperWave*, in diesem Abschnitt exemplarisch vorstellen wollen.

Die Beteiligung der Benutzer bei der Aktualisierung des Datenbestandes einschließlich der Ergänzung durch Anmerkungen wird vom *Ariadne*-Server [48, 63] unterstützt, der im Rahmen des MeDoc-Projekts [21] entwickelt wurde. *Ariadne* ist ein Werkzeug zur interaktiven Produktion und Verteilung von Informatik-Fachinformation im WWW. Benutzer können neue Quellen angeben und Ereignisse, Quellen und Veröffentlichungen kommentieren. Die Qualität der Information lässt sich durch Moderation kontrollieren, und man kann elektronische Fachdiskussionen führen. Die inhaltlichen Beziehungen zwischen Dokumenten werden aber nicht zur Erschließung herangezogen oder kommentiert.

In Zusammenhang mit Literaturverweisen finden sich feste Literatursammlungen einzelner Personen mit persönlichen Annotationen, die von ihnen als Referenzseiten im Web angeboten werden. Der Haupteinsatzort von Annotationssystemen ist derzeit das WWW und UsenetNews, mit dem Ziel der Diskussion. Die Arbeiten in diesem Gebiet beschäftigen sich mit dem Verwalten der Annotationen und vor allem mit dem Problem der Skalierbarkeit [161, 162, 171].

Ein System, das sowohl Annotationen ermöglicht als auch Beziehungen zwischen Dokumenten verwalten kann, ist *HyperWave* (kommerzielle Variante des *Hyper-G* Systems) [40, 127]. *HyperWave* ist ein Client-Server-basiertes Internet-Informationssystem mit dem Ziel, eine große Anzahl von verteilt

vorliegenden hypermedialen Informationseinheiten in strukturierter Form anzubieten, zu verwalten und den Zugriff zu bereits bestehenden Informationssystemen, z.B. dem WWW, zu ermöglichen. *HyperWave* vereinigt die top-down-orientierte hierarchische Navigation entlang sogenannter Kollektionen mit den Vorzügen von Hyperlinks, in Kombination mit einer Datenbank, die Attribut- und Volltextsuche erlaubt. Textdokumente werden beim Einbringen volltextindiziert. Links existieren als eigenständige Objekte in der Datenbank und sind bidirektional. Es gibt ein abgestuftes Nutzer- und Gruppenkonzept mit einer flexiblen Vergabe von Zugriffsrechten auf Dokumente.

Diesen großen Vorzügen stehen für unsere Ziele leider einige Nachteile gegenüber. Zum einen sind unsere bibliographischen Daten strukturiert. Diese Struktur wird für die Recherche benötigt; sie darf nicht durch Speicherung als Volltext in der internen Datenbank verlorengehen. Eine hierarchische Zerlegung in Kollektionen scheint für diesen Zweck nicht sinnvoll. Hält man aber die bibliographischen Daten in einer externen Datenbank, entsteht erheblicher Mehraufwand bei den Annotationen: Zu jeder Annotation müssen zusätzliche Daten gespeichert werden, aus denen hervorgeht, welches Dokument annotiert wird. Für eine schnelle Recherche der Annotationen zu einem Dokument empfiehlt es sich außerdem, alle Annotationen zu protokollieren; siehe Fellner u. a. [58].

Zum anderen wollen wir nicht von jedem Benutzer verlangen, einen eigenen *HyperWave*-Server einzurichten und zu betreiben. Bei entferntem Zugriff auf einen zentralen Server scheint es aber fraglich, ob die für interaktives Navigieren benötigte Performanz erreicht werden kann.

Schließlich bieten weder *HyperWave* noch *Ariadne* die Möglichkeit, Graphen zu visualisieren.

In den letzten drei Abschnitten haben wir Annotationen, typisierte Links und strukturbasierte Anfragen betrachtet. Sie stellen geeignete Hilfsmittel bei der Strukturierung und Wiederverwendung von Informationen dar. Um den Informationsbedarf des Einzelnen zu decken, ist neben diesen Methoden, eine Visualisierung des Wissensgeflechts nach benutzerdefinierten Kriterien bei der Recherche und Interpretation der Datenbasis hilfreich. Eine interaktive Visualisierung kann zusätzlich auch zur einfachen Eingabe von Beziehungen und Annotationen dienen. Mit dem Stand der Forschung von grafischen Visualisierungsoberflächen für Literaturdatenbanken befassen wir uns im folgenden Abschnitt.

2.6 Visualisierung

Die meisten existierenden Anwendungsoberflächen für Literaturdatenbanken bieten nur eine einfache Maske für die Eingabe von bibliographischen Daten oder Schlüsselwörtern. Teilweise werden alternative Anfragewörter in Auswahllisten vorgeschlagen, oder es können gefundene Dokumente zur Anfrageerweiterung verwendet werden.

Nur in wenigen Systemen werden inhaltsbasierte Beziehungen visualisiert. Diese Systeme verwenden allerdings jeweils nur einen Typ inhaltlicher Beziehungen, etwa die Zugehörigkeit zur selben Kategorie eines Klassifikationsschemas, ein- und ausgehende Zitate oder gemeinsam vorkommende Schlüsselwörter.

So kann man beim *Facet-Space Interface* [4] Dokumente anhand des Klassifikationsschemas der ACM *Computing Reviews* suchen; gleichzeitig ist eine Thesauri-Term-Identifikation möglich. Im Facet Display Widget werden die Klassifikationsmarken angezeigt und ausgewählt; die Marken können in eine Liste aktueller Constraints aufgenommen werden. Im Shelf Widget werden diejenigen Dokumente angezeigt, die alle Constraints erfüllen.

Das *Butterfly*-System [122] von Xerox dient der bibliographischen Recherche über Literaturverweise, verfügt aber zusätzlich über eine graphische Oberfläche. Der aktuelle Artikel, bei dem die Suche gerade angekommen ist, wird in Form eines Schmetterlings dargestellt: Sein Kopf beinhaltet den Titel, den Autor, das Erscheinungsjahr und den Erscheinungsort des Artikels. Im linken Schmetterlingsflügel werden die im Artikel vorhandenen Verweise auf andere Dokumente aufgelistet und im rechten Flügel die Dokumente, die den Artikel zitieren. Auf diese Weise wird ein einzelner Knoten des Zitiergraphen dargestellt.

Man kann den Referenzen nachgehen und so zu anderen Knoten gelangen. Der dabei besuchte Teilgraph lässt sich in einem sogenannten Scatterplot dreidimensional darstellen. Eine Darstellung des gesamten Informationsraumes, aus der sich eine direkte Übersicht ergäbe, ist aber nicht möglich. Die Oberfläche ist nicht leicht zu bedienen. Im *Butterfly*-System ist es auch nicht möglich, andere Beziehungen als die Zitierrelation darzustellen.

Auch das System *LyberWorld* [191] bietet nur eine lokale Sicht auf diejenigen Dokumente, die im Laufe der Anfrage gefunden wurden. Es setzt auf dem IR-System *Inquery* [29] auf und stellt verschiedene Visualisierungstools zur Verfügung. *Kegelbäume*, eine dreidimensionale Erweiterung traditioneller Bäume, werden zum Navigieren in der Datenbasis entlang eines inhaltsorientierten Suchpfades verwendet. Die Kegel stellen in abwechselnder Reihenfolge entweder Begriffe oder Dokumenttitel dar: Bei Auswahl eines Dokumentes werden im nächsten Kegel die Schlüsselbegriffe präsentiert, die den Inhalt

des markierten Dokuments umreißen. Die Auswahl eines dieser Begriffe führt wieder zu einem Kegel mit Dokumenttiteln, und so fort. Relevanzkugeln unterstützen die Nutzer dabei, einen Kern von relevanten Dokumenten aus dem mit Hilfe der Navigationskegel gewonnenen Resultateraum herauszuarbeiten.

Die mit einem Kegelbaum gewonnenen Ergebnismengen können in *LyberWorld* mit Hilfe von Referenzkugeln näher untersucht werden. Die Terme, die im Kegelbaum ausgewählt wurden, werden gleichmäßig verteilt auf der Oberfläche der Relevanzkugel angezeigt. Im Innern der transparent dargestellten Referenzkugel befinden sich die Dokumente der zugehörigen Ergebnismenge. Die Position eines Dokuments ergibt sich durch Addition der Anziehungsvektoren zwischen dem Dokument und jedem einzelnen Termknoten, wobei die Stärke der Anziehung durch die jeweilige Termfrequenz bestimmt ist. Aus den resultierenden Dokumentpositionen kann der Benutzer weitere Informationen bzgl. der Relevanzen und Gruppierungen der Dokumente ablesen.

In *LyberWorld* wird das Dokument-Begriffs-Netzwerks auf eine Baumhierarchie reduziert, die durch Schlüsselwörter definiert ist; eine Visualisierung oder Recherche anderer inhaltlicher Beziehungen wie etwa des Zitiergeflechtes ist nicht möglich. Weder bei *Butterfly* noch bei *LyberWorld* kann man mit einer globalen Übersicht starten und sie dann verfeinern oder sich auf Teilstrukturen konzentrieren.

Ebenfalls mit einer Gravitationsmetapher, wie sie bei der Relevanzkugel genutzt wird, arbeiten die Systeme *VIBE* [149], *VINETA* [101] und *SENTINEL* [60].

In *Narcissus* und *HyperSpace* [81, 82, 202] wird das Layout komplexer Netze wie beispielsweise Hyperlinkstrukturen als dreidimensionaler Graph dargestellt, indem ein selbstorganisierendes System anziehende und abstoßende Kräfte simuliert. Die Funktionsweise solcher kräftegesteuerten Layoutalgorithmen (*Force Directed Placement*, FDP) wird in Abschnitt 3.2.1 ausführlich beschrieben.

Seit einigen Jahren werden zweidimensionale *Wissenskarten* zur Darstellung der Struktur von Dokumentenbeständen verwendet. Diese Karten drücken semantische Nähe von Dokumenten durch geringe räumliche Distanz aus. Die Vorgehensweise bei der Erstellung der Wissenskarten erfolgt dabei stets nach demselben Prinzip. Zunächst wird eine Matrix berechnet, die die Ähnlichkeit zwischen den Dokumenten wiedergibt. Dabei wird die Ähnlichkeit mit Hilfe von Zitieranalyse [180], Termanalyse, Klassifikationsschemen, *Latent Semantic Analysis* (LSA) [15, 36] oder Faktorenanalyse [37, 129, 157] bestimmt. Zusätzlich werden *Domain Maps* immer üblicher, die auf der Kozitationsanalyse basieren [198]. Anschließend werden die Dokumente mit verschiedenen Verfahren, wie z.B. hierarchisches Clustern, k-means

Algorithmen oder *Pathfinder Network* (PFNET)[3], geclustert. Am häufigsten werden die von Kohonen [99] entwickelten selbstorganisierenden Karten (*Self-Organizing Maps*, SOM) eingesetzt.

Eibl und Mandl [51] haben gezeigt, dass die verschiedenen Methoden sowohl bei kleinen als auch großen Dokumentenmengen zu sehr unterschiedlichen Karten führen und dass es zwischen den Karten keine Korrelation gibt. Ein weiteres Problem ist der enorme Informationsverlust durch die Reduktion der zahlreichen Dimensionen des Term-Raums auf eine zweidimensionale Darstellung. Es ist eher unwahrscheinlich, dass eine so starke Dimensionsreduktion die Struktur der Dokumentenmenge noch angemessen widerspiegelt. Die zweidimensionale Darstellung kann in einer größeren Dokumentenmenge insgesamt nur eine begrenzte Anzahl von Beziehungen in Form räumlicher Nachbarschaft darstellen.

Eine neuere Entwicklung im Bereich der Zitatenanalyse ist die Darstellung des minimalen Spannbaumes basierend auf der Kozitationsdistanz zwischen Dokumenten [36, 37, 38]. Dazu werden zunächst mittels eines PFNET Algorithmus die wesentlichen Beziehungen des Autor-Kozitationsnetzwerkes bestimmt. Der Spannbaum zeigt dann die minimale Menge wesentlicher Zitierverknüpfungen innerhalb der Dokumentenmenge und spiegelt so das Netzwerk der direkten Einflüsse wieder. Fundamentale Arbeiten werden im Zentrum des Netzes angezeigt. Neuere Forschungsarbeiten werden an den Kanten entfernt vom Zentrum angeordnet. Das Layout wird mit einem kräftegesteuerten Verfahren bestimmt. StarWalker von Chen [34, 35] ist eine VRML-Anwendung, die es ermöglicht, in dem Netzwerk zu navigieren und sich direkt in Form eines Chats mit anderen Benutzern auszutauschen. Da der Informationsaustausch in StarWalker mit Hilfe eines Chats erfolgt, ist er nur temporär. Ein Benutzer kann nur dann davon profitieren, wenn zufällig gerade der passende Experte im Chat ist. Hier ist BibRelEx mit seiner persistenten Wissensanreicherung klar überlegen. Weiterhin ist die Datenmenge bzw. der Graph in dem der Benutzer navigieren kann fest vorgegeben und kann nicht durch den Benutzer durch Aufnahme eigener Arbeiten verändert werden. Das Beziehungsgeflecht ist statisch vorgegeben.

Noel [144] hat die Visualisierung des Spannbaums auf mehrere Arten erweitert. Er schlägt beispielsweise vor, Cluster explizit in der Visualisierung zu zeigen, wobei nur die Kanten innerhalb des Clusters beibehalten werden, und führt weitere Metriken zur Distanzbestimmung basierend auf der Kozitation zwischen den Dokumenten ein. Daneben kombiniert er die Spannbaumdarstellung mit der auf Textähnlichkeit basierenden Landschaftsdarstellung

[3]Ein PFNET hat die Eigenschaft, dass es nur die Kanten enthält, die Teil eines Pfades mit minimalem Gewicht sind.

von ThemeScape [199]. ThemeScape gruppiert Textinformationen ähnlichen Inhalts in dreidimensionalen Landkarten. Hügel stellen Orte mit vielen Dokumenten zum gleichen Thema da, Senken solche mit Einzelthemen. Stehen Hügel nah beieinander, besitzen sie große Ähnlichkeiten.

Das kommerzielle System BibTechMon [7, 100] berechnet auf Basis einer automatischen Beschlagwortung und den daraus resultierenden Häufigkeiten gemeinsamen Vorkommens der Schlagworte Beziehungen zwischen einzelnen Dokumenten. Das so entstandene Beziehungsgeflecht wird nach einer Clusteranalyse mit Hilfe von kräftebasierten Verfahren dargestellt. Auf diese Art können sowohl ein Gesamtüberblick der Datenbasis als auch Rechercheergebnisse dargestellt werden, in denen wie bei den bisher vorgestellten Systemen thematische Zusammenhänge leicht zu erkennen sind. Im Gegensatz zu BibRelEx wird nur ein Begriffsnetzwerk dargestellt. Es können weder explizit vorhandene Beziehungen, wie sie durch Literatuverweise gegeben sind, noch benutzerdefinierbare Beziehungen zur Analyse genutzt werden.

Eine globale, auf Literaturverweisen beruhende Übersicht ermöglicht *Science Landscape* von Sandia [175, 176], bei dem Wissensgebiete als Landschaften dargestellt werden, über die der Benutzer hinwegfliegen kann. Je tiefer man fliegt, um so mehr Teilgebiete werden sichtbar. Auf unterster Ebene sind die Titel der einzelnen Arbeiten dargestellt. Die Darstellung basiert auf einer Clusterung nach Zitierhäufigkeit. Die Zitierdaten sind gegen eine hohe Lizenzgebühr aus dem in Abschnitt 2.1 erwähnten Science Citation Index übernommen worden. Damit übertragen sich automatisch alle schon oben genannten Nachteile des SCI's auf Science Landscape.

Auf die Verwandschaft zwischen den Forschungsgebieten im WWW, in der Bibliometrie und sozialen Netzwerken haben wir schon in der Einleitung zu diesem Kapitel hingewiesen. Viele der Ergebnisse, die in dem einen Gebiet gewonnen wurden, lassen sich auf die anderen übertragen. So basieren sowohl der HITS-Algorithmus von Kleinberg [94, 95] als auch der *PageRank*-Algorithmus von Page u. a. [150] auf der Methodik der Sozialnetzwerkanalyse. Im Bereich sozialer Netzwerke ging man dabei bisher im wesentlichen der strukturellen Analyse der Netzwerke nach, beispielsweise der Beziehungen zwischen den einzelnen Akteuren und der Erkennung von Gruppenstrukturen und zentralen Akteuren. Die Visualisierung sozialer Netzwerke zur Unterstützung dieser Analyse ist hingegen eine bisher kaum verfolgte Forschungsrichtung. Ein sich in der Entwicklung befindendes Forschungssystem, mit dem Algorithmen zur visuellen Analyse von Netzwerken an der Universität Konstanz untersucht werden, ist Visone [9, 194]. Neben der Darstellung von Netzwerken mit kräftegesteuerten Verfahren ermöglicht Visione bisher eine Schichtendarstellung des Status und eine Darstellung der Zentralität auf Niveaukreisen.

Außer diesen Systemen gibt es noch eine ganze Reihe „reine“ Visualisierungstools, die allgemein die Darstellung von Datenmengen zum Ziel haben. Dazu gehören unter anderem der *Kegelbäume* [53, 136, 138, 159, 160], die *Fisheye Views* [64, 170], *hyperbolische Bäume* [104], *Perspective Walls* [53, 121, 159] und schließlich das *Information Visualization and Exploration Environment* [1, 103, 200].

Keines der in diesem Abschnitt genannten Systeme erlaubt es, annotierte Verweise zwischen Dokumenten einzugeben und zu visualisieren. Die verschiedenen Techniken zur Dimensionsreduktion des Informationsraumes, LSA, SOM oder PFNET haben alle gemeinsam, dass sie ein aufwendiges Preprocessing erfordern und nur ein statisches Layout ermöglichen. Damit sind sie für BibRelEx nicht geeignet, da uns ein dynamisches Layout wichtig ist, um den Einfluss von einzelnen Arbeiten auf das Beziehungsgeflecht nachverfolgen zu können. Damit kommen von den eben vorgestellten Visualisierungstechniken nur die kräftegesteuerten Verfahren zur Realisierung der Visualisierung in BibRelEx in Frage. Sie lassen sich leicht zu inkrementellen Verfahren erweitern und erlauben somit eine dynamische Darstellung. Der Nachteil der kräftegesteuerten Verfahren liegt allerdings in ihrem hohen Berechnungsaufwand und damit schlechter Skalierbarkeit. Bei diesen Verfahren müssen die Distanzen aller Knoten voneinander berechnet werden, womit Sie eine quadratische Laufzeit (in Bezug auf die Anzahl der Knoten) haben. Lediglich der 2D VxInsight Ordination Algorithmus VxOrd [16], ein kräftegesteuertes Verfahren, das bei Science Landscape von Sandia, s.o., eingesetzt wird, ist für große Graphen geeignet. Bei VxOrd werden die Distanzen nicht separat berechnet, sondern ein Dichtefeld betrachtet, zu dem jeder Knoten einen Energieanteil abhängig von seiner Position beiträgt. Durch ein Preprocessing kann dieser Anteil in linearer Zeit für alle Knoten bestimmt werden. Damit wird die Laufzeitverbesserung auf Kosten der Dynamik erkauft, und somit ist VxOrd nicht für unsere Zwecke geeignet.

Wegen der schlechten Skalierbarkeit der kräftegesteuerten Layout Algorithmen, müssen geeignete Cluster-Verfahren eingesetzt werden, um die Anzahl der Knoten im Beziehungsgeflecht soweit zu reduzieren, dass ein für den Benutzer annehmbares Laufzeitverhalten erreicht wird. Die Reduzierung des Detaillierungsgrades der Darstellung hat darüber hinaus den Vorteil, dass sie die „Lesbarkeit“ der Darstellung erhöht.

Die Funktionsweise von kräftegesteuerten Layoutverfahren und für die Realisierung unserer Ziele geeigneter Cluster-Verfahren wird im nächsten Kapitel ausführlich vorgestellt.

Kapitel 3

Grundlagen

Kapitel 2 gab einen Überblick über den aktuellen Stand der Forschung. In diesem Kapitel werden nun die Verfahren näher beschrieben, die in BibRelEx Anwendung finden. Zunächst werden die Konzepte der Zitatenanalyse, der Kozitationsanalyse und der bibliographischen Koppelung beschrieben. Diese Konzepte lassen sich leicht auf andere Beziehungen übertragen und bilden eine wesentliche Grundlage der Linkanalyse. Danach werden die in BibRelEx angewandten Layout- und Cluster-Verfahren vorgestellt. Abschließend wird die bibliographische Datenbasis GeomBib vorgestellt, an der wir unsere Ideen erprobt haben.

3.1 Zitatenanalyse

Die Zitatenanalyse ist ein Teilgebiet der Bibliometrie, das sich mit dem Studium der Beziehungen zwischen zitierten und zitierenden Arbeiten und ihrer Anwendung als bibliometrische Untersuchungsmethode befasst. Als bibliometrischen Parameter verwendet die Zitatenanalyse die Zitierhäufigkeit, d.h. es werden die auf eine bestimmte Arbeit, ein bestimmtes Dokument oder einen bestimmten Autor entfallenden Zitate gezählt. Je größer die Zitierhäufigkeit einer Arbeit ist, desto größer wird der Wert der Arbeit veranschlagt. Die Zitierhäufigkeit zeigt, welchen Einfluss (Auswirkung, Resonanz, Impakt) eine Arbeit für andere Wissenschaftler hat oder gehabt hat, sagt aber nichts Endgültiges über die Qualität der Arbeit aus. Es ist jedoch sehr wahrscheinlich, dass eine häufig zitierte Arbeit wichtige Informationen enthält.

Eine wichtige Anwendung der Zitatenanalyse ist die Auswahl von Monographien und Zeitschriften in der bibliothekarischen Erwerbung auf Basis der Zitierhäufigkeit. Zu diesem Zweck wurde von Garfield [68] der sogenannte *Impakt-Faktor* eingeführt, der die durchschnittliche Frequenz anzeigt, mit der

ein Artikel in einem bestimmten Zeitraum zitiert wurde. Der Impakt-Faktor wird vom *Institute for Scientific Information* (ISI) im *Science Citation Index - Journal Citation Report* (SCI-JCR) veröffentlicht [72, 73]. Dabei wird im SCI aufgelistet, welcher Autor welchen anderen Autor im jeweiligen Aufsatz zitiert. Im JCR werden diese Informationen nach Zeitschriften zusammengefasst.

Die Bewertung mittels Impakt-Faktor ist nicht unumstritten. Der Impakt-Faktor berücksichtigt nur die eingehenden Verweise, was – wie bereits bei der Diskussion des PageRank-Algorithmus auf Seite 2.4 gesehen – nicht ausreichend ist. Die Auswahl, welche Zeitschriften ausgewertet werden, liegt allein beim Herausgeber. Zitierraten sind u.a. von den einzelnen Fachdisziplinen abhängig, bedingt durch unterschiedliches Zitierverhalten. Neue Forschungszweige werden seltener zitiert. Nur die Zitate der Artikel der beiden Vorjahre zählen mit. Dadurch fallen „Zitationsklassiker“ weg. Zeitschriften, die viele Review-Artikel enthalten, haben einen höheren Impakt-Faktor. Selbstzitierung und Zitiergemeinschaften können das Ergebnis beeinflussen. Proceedings und Technische Berichte sind nicht im SCI enthalten. Dieser Nachteil wiegt umso schwerer, weil es für Benutzer nicht möglich ist, eigene Dokumente oder Verweise einzugeben. Nicht zu vernachlässigen ist, dass die Lizenzkosten des Science Citation Index recht hoch sind.

Mit Hilfe der Zitierbeziehung lassen sich Dokumente zu Gruppen/Cluster logisch zusammenhängender Dokumente zusammenfassen. Das Konzept der Gruppen des gemeinsamen Zitierens (Kozitationsanalyse) wurde gleichzeitig unabhängig voneinander von Small [178] in Philadelphia und von Marshakova [126] in Moskau entwickelt. Die Kozitationsanalyse basiert auf der Hypothese, dass wenn zwei Arbeiten A, B von einer dritten gemeinsam zitiert werden, es eine kognitive Verknüpfung zwischen beiden gibt. Wenn viele Dokumente die zwei Arbeiten A, B gemeinsam zitieren, bedeutet dies, dass die beiden Arbeiten ein verwandtes Thema behandeln. Die Stärke dieser Verknüpfung wird von der Häufigkeit angegeben, mit der die beiden Arbeiten zusammen zitiert werden und kann mit Garfields Formel berechnet werden:

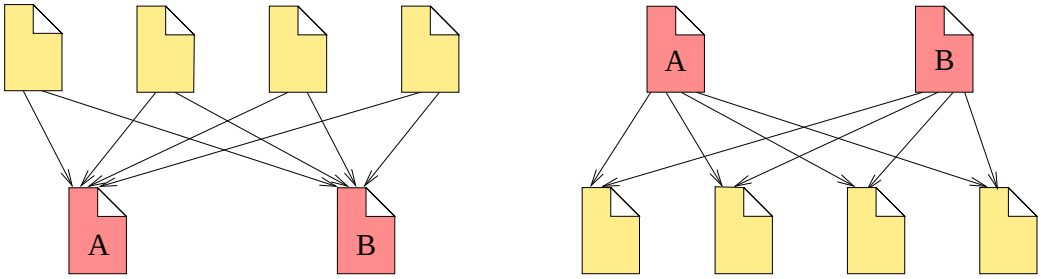
$$S = \frac{\text{Anzahl Kozitate von } A \text{ und } B}{\text{Anzahl Zitate von } A \text{ und } B - \text{Anzahl Kozitate von } A \text{ und } B}$$

Die Gruppe von Arbeiten, die mit einer bestimmten festgelegten Häufigkeit gemeinsam zitiert werden, bildet einen Cluster. Die Cluster beziehen sich auf Gebiete, ihre Verbindungen auf interdisziplinäre Beziehungen. Diese Analyse bietet die Möglichkeit, die Struktur der Fachgebiete zueinander zu erforschen.

Die Kozitationsanalyse hat den Nachteil, dass sie immer eine bestimmte Verzögerung mit sich trägt, da es eine gewisse Zeit dauert bis Arbeiten zi-

tiert werden. Diese Verzögerung wird bei der Analyse der bibliographischen Koppelung vermieden.

Zwei Dokumente sind bibliographisch gekoppelt, wenn ihre Referenzenlisten eine oder mehrere Arbeiten teilen [92]. Bei der bibliographischen Koppelung werden somit jüngere Arbeiten verbunden, da sie gleiche ältere zitieren. Dagegen werden durch die Kozitationsanalyse ältere Arbeiten verbunden, da sie später von jüngeren Arbeiten gemeinsam zitiert werden [181]. Dieser Zusammenhang von Kozitation und bibliographischer Koppelung ist in Abbildung 3.1 dargestellt: Die Dokumente A und B sind in Abbildung 3.1(a) kozitiert, in Abbildung 3.1(b) bibliographisch gekoppelt.



(a) Kozitation (= gleiche Eltern)

(b) Bibliographische Koppelung (= gleiche Kinder)

Abbildung 3.1: Zusammenhang von Kozitation und bibliographische Koppelung

Der Nachteil der bibliographischen Koppelung ist, dass sie subjektiv ist, da die Beziehungen zwischen den Dokumenten auf Angaben ihrer eigenen Autoren beruhen.

Eine Anwendung der bibliographischen Koppelung ist die Suche nach themenverwandten Dokumenten. Dazu wird die Indexierungsgewichtung nach der *Common Citation x Inverse Document Frequency* (CCIDF) Gewichtung berechnet [70]. Sie ist ähnlich der aus dem Information Retrieval bekanntem wortorientierten *Term Frequency x Inverse Document Frequency* (TFIDF) Gewichtung, da sie die gemeinsamen Zitate zwischen jedem Dokumentenpaar mit der inversen Zitierhäufigkeit gewichtet. Seien

$\|C\|$	die Anzahl der Zitate in der Kollektion,
$\|D\|$	die Anzahl der Dokumente in der Kollektion,
d_m^C	die Menge der in Dokument d_m vorkommenden Zitate,
cf_{mi}	die Vorkommenshäufigkeit von Zitat c_i in Dokument d_m,
$\max cf_{mi}$	die maximale Vorkommenshäufigkeit aller Zitate $c_i \in d_m^C$,
n_i	die Anzahl der Dokumente, in denen Zitat c_i vorkommt.

mit $m \in \{1, \ldots, |D|\}$ und $i \in \{1, \ldots, |C|\}$.

Eine Komponente der Gewichtung ist die inverse Dokumenthäufigkeit idf_i, die umso höher ist, je seltener ein Zitat in der Kollektion vorkommt:

$$idf_i = \log \frac{|D|}{n_i}$$

Die zweite Komponente ist die normalisierte Vorkommenshäufigkeit ncf_{mi}. Hierbei werden die Zitate entsprechend ihrer Vorkommenshäufigkeit im Dokument gewichtet. Um den Einfluss der Dokumentenlänge auszugleichen, wird diese Häufigkeit durch die maximale Vorkommenshäufigkeit eines Zitats in dem betreffenden Dokument normalisiert:

$$ncf_{mi} = 0.5 \left(1 + \frac{cf_{mi}}{\max cf_{mi}}\right)$$

Hat man - wie in vielen Literaturdatenbanken üblich - nur die Information vorliegen, ob ein Zitat in einem Dokument vorkommt oder nicht, vereinfacht sich die Vorkommenshäufigkeit zu

$$ncf_{mi} = \begin{cases} 1 & \text{wenn Zitat } c_i \text{ in Dokument } d_m \text{ vorkommt} \\ 0 & \text{sonst} \end{cases}$$

Das unnormierte Indexierungsgewicht berechnet sich als Produkt der beiden Komponenten:

$$\alpha_{mi} = ncf_{mi} * idf_i$$

Abschließend wird üblicherweise der Dokumentenvektor noch normiert, so dass $|\vec{d_m}| = 1$. Der Normierungsfaktor berechnet sich dabei zu

$$w_m = \sqrt{\sum_{c_i \in d_m^C} \alpha_{mi}^2}$$

Damit ergibt sich das endgültige normierte Indexierungsgewicht

$$d_{mi} = \frac{\alpha_{mi}}{w_m}$$

Die Ähnlichkeit zweier Dokumente d_j und d_k mit $j, k \in \{1, \ldots, |D|\}$ ist anhand des Cosinus-Winkels zwischen den zugehörigen Dokumentvektoren $\vec{d_j}$, $\vec{d_k}$ messbar. Je kleiner das Cosinus-Maß des Winkels zwischen den beiden Dokumentvektoren, desto größer ist die Ähnlichkeit der beiden Dokumente:

$$cos(\angle(\vec{d_j}, \vec{d_k})) = \frac{\vec{d_j} \cdot \vec{d_k}}{||\vec{d_j}|| \cdot ||\vec{d_k}||}$$

Die CCIDF Gewichtung wird im Nachfolger von *Citeseer* [70], dem *ResearchIndex* [141], verwendet. Der ResearchIndex ist ein automatisch generierter Zitierindex für wissenschaftliche Literatur, insbesondere aus dem Bereich der Informatik.

Analog zu der Übertragung der TFIDF Gewichtung auf die Zitierrelation lässt sich die TFIDF Gewichtung auch leicht auf beliebige andere Beziehungen zwischen Dokumenten übertragen.

3.2 Layout

Die in BibRelEx verwendete Abbildung von Dokumenten auf Knoten und inhaltlicher Beziehungen auf Kanten führt auf das Problem der Visualisierung von großen Graphen mit gegebenenfalls tausenden von Knoten. Es gibt eine Vielzahl von Algorithmen zur optimalen Darstellung von Graphen im zweidimensionalen Raum. Allgemeine Zielsetzungen sind dabei beispielsweise eine minimale Anzahl von Kantenüberkreuzungen oder eine räumlich nahe Anordnung von strukturell zusammenhängenden Knoten. Eine umfangreiche Übersicht von 2D-Layoutalgorithmen findet sich in [91].

Die Darstellung von Graphen im dreidimensionalen Raum ist erheblich aufwendiger. Einfaches Erweitern der zweidimensionalen Layout-Algorithmen um eine extra Dimension führt oft nicht zum gewünschten Ziel. Die zusätzliche Dimension ermöglicht eine viel flexiblere Darstellung. So können Kantenüberkreuzungen im dreidimensionalen Raum prinzipiell vermieden werden. Allerdings sind heutige Ausgabemedien nur zweidimensional, so dass durch die Projektion auf das zweidimensionale Ausgabemedium für den Betrachter wieder Kantenüberkreuzungen entstehen und so für den Betrachter die Struktur großer Graphen nur schwer zu erfassen ist. Diese Nachteile können teilweise durch das Nutzen von Navigationsmöglichkeiten wie Drehen und Zoomen ausgeglichen werden. Eine geeignete Wahl des Betrachtungspunktes (*Viewpoint*) kann darüber hinaus das Problem der Kantenüberkreuzung weiter minimieren. Landgraf [109] gibt eine Beschreibung von Layout-Verfahren im dreidimensionalen Raum und die Wahl geeigneter Viewpoints.

Eine weitere Methode um die Visualisierung großer Graphen übersichtlich zu halten, ist die Verwendung von Cluster-Verfahren. Diese werden in Abschnitt 3.2.5 vorgestellt.

Daneben ist auch die Verwendung von inkrementellen Verfahren, die eine gegebene Zeichnung so gut wie möglich erhalten und die Änderungen schonend einbauen, eine hilfreiche Unterstützung für den Benutzer, um die Übersicht bei kleinen Änderungen des Datenbestands nicht zu verlieren. Man spricht in diesem Zusammenhang auch von der Erhaltung der „Mental

Map“ des Benutzers. Die in BibRelEx verwendeten kräftegesteuerten Layout-Methoden lassen sich leicht zu inkrementellen Verfahren erweitern, indem die Kräfte nur auf neu hinzukommende Knoten angewandt werden und die einzelnen Iterationsschritte animiert angezeigt werden. Der Benutzer kann so den Einfluss der entsprechenden Knoten (Dokumente) leicht nachvollziehen. Eine Übersicht des Stands der Forschung im Bereich dynamischer Layout-Methoden gibt Branke [19].

Im Folgenden werden nur die Algorithmen beschrieben, die in BibRelEx verwendet werden. Dabei werden folgende Notationen verwendet: Ein Graph $G = (V, E)$ hat eine endliche Menge $V = \{v_1, \ldots, v_n\}$ von *Knoten* und eine endliche Menge $E = \{e_1, \ldots, e_m\}$ von *Kanten*, wobei jede Kante $e \in E$ ein geordnetes oder ungeordnetes Paar von Knoten ist. Ein geordnetes Paar $e = (u, v) \in V \times V$ ist eine gerichtete Kante (Pfad), ein ungeordnetes Paar von Knoten $e = \{u, v\} \subseteq V \times V$ eine ungerichtete Kante. $d_G(v)$ bezeichnet den Knotengrad eines Knoten v, d.h. die Anzahl der ein- und ausgehenden Kanten von v. Ein Graph-Layout ist ein Tupel (G, p), wobei $G = (V, E)$ der zu visualisierende Graph ist und durch $p : V \to R$ mit $R = \mathbb{R}^2$ bzw. $R = \mathbb{R}^3$ eine 2 bzw. 3 dimensionale Position für jeden Knoten $v \in V$ bestimmt wird.

3.2.1 Der Spring-Embedder-Algorithmus

Kräftegesteuerte Methoden existieren schon lange, z.B. Tutte [193], und sind in vielen Varianten publiziert worden. Als grundlegend für das Anwenden von kräftegesteuerten Methoden im Bereich Zeichnen von Graphen wird die Arbeit von Eades [49] angesehen, die den *Spring-Embedder*-Algorithmus enthält. Der Spring-Embedder-Algorithmus basiert auf einem physikalischen Modell, bei dem die Knoten des Graphen als elektrisch geladene Partikel betrachtet werden, die sich gegenseitig abstoßen. Die Kanten des Graphen werden als Federn modelliert, die die durch sie verbundenen Knoten anziehen. Ausgehend von einer zufälligen Anordnung der Knoten im Raum strebt ein solches System einen stabilen energiearmen Zustand an. In diesem „Ruhezustand“ hat das System eine einheitliche und ausgewogene Struktur, eventuell enthaltene Symmetrien sind gut zu erkennen. Diese Struktur erfüllt damit wesentliche Eigenschaften, die man von einem *ästhetischen* Layout verlangt. Zusätzlich hat die resultierende Darstellung den Vorteil, dass in Beziehung stehende Knoten räumlich nah zueinander angeordnet werden, wobei ein Minimalabstand der Knoten sichergestellt ist.

Um das Verhalten des kräftegesteuerten Systems zu simulieren, werden die Teilchenpositionen iterativ berechnet. Algorithmus 1 gibt eine prinzipielle Beschreibung des Spring Embedders wieder, wobei $f_{\mathsf{rep}}(u, v)$ die elektrische Abstoßung zwischen den Knoten u und v, $f_{\mathsf{spring}}(u, v)$ die anziehende Kraft

der Feder zwischen den Knoten u und v, und $F_v(t)$ die daraus resultierende Kraft, die zum Zeitpunkt t auf einen Knoten v wirkt, bezeichnet. Um übermäßige Bewegungen beim synchronen Bewegen aller Knoten nach der Kräfteberechnung zu vermeiden, wird die Bewegung durch die Verwendung der multiplikativen Konstante δ gedämpft.

Eingabe : verbundener ungerichteter Graph $G = (V, E)$
zufällige Startpositionen $p = \left(p_v^{(0)}\right)_{v \in V}$
Anzahl n_{iter} der auszuführenden Iterationen
Ausgabe : Layout $L = (G, p)$ wobei p eine Platzierung mit (lokal) minimaler Spannung ist

for $t \leftarrow 1$ ***to*** n_{iter} **do**
  for $v \in V$ **do**
    $$F_v(t) \leftarrow \sum_{u\,:\,\{u,v\} \notin E} f_{\mathsf{rep}}(p_u^{(t-1)}, p_v^{(t-1)}) + \sum_{u\,:\,\{u,v\} \in E} f_{\mathsf{spring}}(p_u^{(t-1)}, p_v^{(t-1)})$$
  for $v \in V$ **do** $p_v^{(t)} \leftarrow p_v^{(t-1)} + \delta \cdot F_v(t)$

Algorithmus 1: Spring Embedder nach Eades [49]

Die einzelnen Varianten des Spring Embedders unterscheiden sich in der Berechnung der Art der Kräfte und der Verfahren das Ende der Iterationen zu erkennen.

Bei Eades [49] werden zwei Knoten u und v, die mit den Abstand $d(p_u, p_v)$ platziert sind, durch Federkräfte proportional zu $\log(\frac{d(p_u,p_v)}{\delta_{u,v}})$ angezogen, wobei $\delta_{u,v}$ die natürliche[1] Länge der Feder zwischen u und v bezeichnet. Die Knoten werden nur dann angezogen, wenn $d(p_u, p_v) > \delta_{u,v}$. Nicht verbundene Knoten werden durch Kräfte proportional zu $\frac{1}{d^2(p_u,p_v)}$ abgestoßen. In jeder Iteration werden die Kräfte für alle Knoten neu berechnet und anschließend alle Knoten synchron bewegt. Bei dem Verfahren von Eades wird immer nach einer festen Anzahl von Iterationen terminiert.

Kamada und Kawai [90] betrachten dagegen die Energie zwischen zwei Knoten

$$E(u, v) = k \cdot \frac{(d(p_u, p_v) - \delta_{u,v})^2}{\delta_{u,v}^2}$$

wobei sie für $\delta_{u,v}$ die Anzahl der Kanten auf dem kürzesten Pfad zwischen u und v wählen und k eine Skalierungskonstante ist. In jeder Iteration wird der

[1] d.h. Länge der Feder, wenn keine Kraft auf die Feder wirkt.

Knoten verschoben, der bei festen Positionen der anderen Knoten die größte Verminderung der globalen Energie bewirkt. Die Terminierung erfolgt, wenn die maximale Energie zwischen zwei Knoten unter eine gegebene Schranke fällt.

Fruchterman und Reingold [62] haben die Kräfte so geändert, dass sie sich schneller berechnen lassen. Im Gegensatz zu Eades benutzen sie abstoßende Kräfte

$$f_{\mathsf{rep}}(p_u, p_v) = \frac{l^2}{d^2(p_u, p_v)} \cdot (p_v - p_u)$$

zwischen allen Knotenpaaren und zusätzlich anziehende Kräfte

$$f_{\mathsf{spring}}(p_u, p_v) = \frac{d(p_u, p_v)}{l} \cdot (p_u - p_v)$$

zwischen Nachbarknoten, wobei l ein gewählter Idealabstand der Knoten ist. Zusätzlich wird ein Temperaturparameter in ähnlicher Weise wie beim Simulated Annealing (Abschnit 3.2.2) verwendet, um die Knotenbewegungen mit fortlaufender Iteration abzubremsen.

Ein wesentlicher Nachteil des Spring Embedders ist das schlechte Laufzeitverhalten (je Iterationsschritt $O(n^2)$ für die Berechnung aller Kräfte), so dass er nur für kleine Graphen gut anwendbar ist. Für große Graphen bietet Clustern eine Möglichkeit, die Komplexität zu reduzieren [22, 77]. Daneben gibt es verschiedenen Ansätze, das Laufzeitverhalten durch Heuristiken zu verbessern, z.B. [62]. Sugiyama und Misue [186] verwenden zusätzlich magnetische Felder, um weitere Layoutkriterien modellieren zu können. So ermöglicht ihr Algorithmus z.B. gerichtete Graphen so zu zeichnen, möglichst viele Kanten ein eine Richtung, z.B. abwärts, verweisen.

Im Basisalgorithmus besteht neben der Berücksichtigung des Minimalabstands von Knoten keine Möglichkeit, weitere Ästhetikkriterien wie Minimalabstand von Kanten oder die Anzahl von Kantenüberkreuzungen einzubringen. Auch hier gibt es verschiedene Verbesserungsvorschläge, z.B. [20, 50, 186].

Eine gute Übersicht bzgl. vieler Verfeinerungen und Erweiterungen des ursprünglichen Algorithmus findet sich in [47] und [91]. Alle Varianten und der Basisalgorithmus haben allerdings den Nachteil, dass sie dazu neigen nur lokale Minima zu finden. Dieses Problem vermeidet das *Simulated Annealing*, das zufällige Knotenbewegungen verwendet, um ggf. ein lokales Minimum wieder zu verlassen.

In BibRelEx wird die Spring Embedder Variante von Fruchterman und Reingold [62] genutzt, da sie von der verwendeten Bibliothek LEDA angeboten wird. Ein Vorteil des Spring Embedders in Hinblick auf die Qualitätskriterien für ein „schönes“ Layout ist, dass er die Knoten gleichmäßig

verteilt. Für die Darstellung von Beziehungsgeflechten stellt dies allerdings eher einen Nachteil dar, da gemeinsam referierte Dokumente symmetrisch um die referierenden Dokumente angeordnet werden. Dies führt zu zusätzlichen Kantenüberkreuzungen, vgl. Abschnitt 5.6.2. Daher wurde für BibRelEx ein eigener Layout-Algorithmus basierend auf dem GEM-Algorithmus (Abschnitt 3.2.3) entwickelt, bei dem gemeinsam referierte Dokumente zwischen den referierenden Dokumenten angeordnet werden. Um dies zu erreichen, werden zusätzlich zu den anziehenden und abstoßenden Kräften noch Rotationskräfte ähnlich denen der Variante von Sugiyama und Misue [186] verwendet. Durch diese Kräfte werden möglichst alle ausgehenden Kanten eines Knotens in dieselbe Richtung gezeichnet. Durch die Verwendung des GEM-Algorithmus als Basis erreicht der BibRelEx Embedder eine schnellere Konvergenz und reduziert das Problem der Konvergenz zu lokalen Minima.

Ein wesentlicher Vorteil der kräftegesteuerten Verfahren, wie der in diesem Abschnitt vorgestellte Spring Embedder oder das im Folgenden beschriebenen Verfahren Simulated Annealing und GEM ist, dass sie sich leicht zu inkrementellen Verfahren erweitern lassen, indem die Kräfte nur auf neu hinzukommende Knoten angewandt werden und die einzelnen Iterationsschritte animiert angezeigt werden. Der Benutzer kann so den Einfluss der entsprechenden Knoten (Dokumente) leicht nachvollziehen.

3.2.2 Der Simulated Annealing Algorithmus

Simulated Annealing [42, 134, 174] basiert auf dem physikalische Verhalten von Teilchen bei der Abkühlung von Flüssigkeiten zu Kristallen. Bei genügend langsamer Abkühlung erreicht das Gesamtsystem einen Zustand minimaler Energie. Zur Simulation des Prozesses wird wieder iteriert, wobei in jedem Schritt die Temperatur T reduziert wird und man für jeden Knoten eine neue Position durch eine kleine zufällige Bewegung testet. Wird durch diese Bewegung die globale Energie U reduziert, akzeptiert das Verfahren die neue Position. Erhöht sich dagegen die Energie um ΔU, so wird die neue Position nur mit einer Wahrscheinlichkeit $p = e^{\frac{\Delta U}{T}}$ akzeptiert, andernfalls bleibt der Knoten an seiner alten Position. Das zufällige Akzeptieren von Bewegungen, die eigentlich zu einer Verschlechterung der Energiebilanz des Systems führen, ermöglicht dem Algorithmus lokale Minima zu verlassen und so im globalen Minimum oder zumindest in einem lokalen Minimum, das näher am globalen Minimum liegt, zu terminieren. Je mehr das System abkühlt, um so unwahrscheinlicher werden diese energievergrößernden Bewegungen. Algorithmus 2 fasst den Ablauf des Verfahrens nochmal zusammen.

Der große Vorteil von Simulated Annealing ist, dass sich die verschiedenen Ästhetikkriterien für das Layout leicht mit Hilfe der Energiefunktion in

Eingabe : Graph $G = (V, E)$
zufällige Startpositionen $p = (p_v)_{v \in V}$
Starttemperatur T_0
Endtemperatur T_E
Ausgabe : Layout $L = (G, p)$ wobei p eine Platzierung mit (lokal) minimaler Energie $U(p)$ ist

$T \leftarrow T_0$
while $T > T_E$ **do**
  for $v \in V$ **do**
    $p^{\text{new}} \leftarrow p + \Delta_{\mathsf{random}}$
    $\Delta U = U(p) - U(p^{\text{new}})$
    if $(\Delta U > 0) \vee (\mathsf{random} < e^{\frac{\Delta U}{T}})$ **then**
      $p \leftarrow p^{\text{new}}$
      $U(p) \leftarrow U(p^{\text{new}})$
  dekrementiere T

Algorithmus 2: Simulated Annealing

den Algorithmus einbringen lassen. Dazu betrachte man als zu minimierende Energiefunktion

$$\eta = \lambda_1 \eta_1 + \lambda_2 \eta_2 + \cdots + \lambda_k \eta_k,$$

wobei λ_i eine Gewichtung der einzelnen Energiefunktionen η_i als Modelle für die Ästhetikkriterien erlaubt. Davidson und Harel [42] haben beispielsweise folgende Energiefunktionen verwendet:

$$\begin{aligned}
\eta_1 &= \sum_{u,v \in V} \frac{1}{(d(p_u, p_v))^2} && \text{(Abstoßung von Knoten)} \\
\eta_2 &= \sum_{u \in V} \left(\frac{1}{r_u^2} + \frac{1}{l_u^2} + \frac{1}{t_u^2} + \frac{1}{b_u^2} \right) && \text{(Begrenzung der Zeichenfläche)} \\
\eta_3 &= \sum_{(u,v) \in E} (d(p_u, p_v))^2 && \text{(kurze Kanten)} \\
\eta_4 &= \text{Zahl der Kantenkreuzungen}
\end{aligned}$$

mit $d(p_u, p_v)$ Abstand der Knotenpositionen p_u und p_v, und r_u, l_u, t_u, b_u Abstände zum rechten, linken, oberen und unteren Rand der Zeichenfläche.

Simulated Annealing ist sehr flexibel, hat aber den Nachteil, dass die Abkühlung sehr langsam erfolgen muss, um ein ansprechendes Layout zu erhalten. Im Allgemeinen benötigt Simulated Annealing zehnmal mehr Ite-

rationen als der Spring Embedder. Versuche haben gezeigt, dass eine Kombination beider Verfahren sinnvoll ist: Man bewegt die Knoten in Richtung der Kräfte (Spring Embedder), fügt aber zusätzlich noch eine zufällige Kraft hinzu und weist die Bewegung mit einer gewissen Wahrscheinlichkeit zurück, wenn durch sie die globale Energie vergrößert wird (Simulated Annealing).

Ein Algorithmus, der auf dieser Kombination beruht, ist der im nächsten Abschnitt vorgestellte GEM (Graph-EMbedder) Algorithmus von Frick u. a. [61]. Er enthält eine Reihe von Heuristiken, um die Konvergenz zu beschleunigen, z.B. lokale Temperaturen, Knotenanziehung in Richtung des Barycenters der Knotennachbarn und die Entdeckung von Oszillationen und Rotationen.

3.2.3 Der GEM-Algorithmus

Frick u. a. [61] haben die beiden Verfahren Spring Embedder und Simulated Annealing kombiniert und dabei die Berechnung der Kräfte und das Iterationsverfahren modifiziert, um bessere Laufzeiten und eine höhere Layoutqualität im zweidimensionalen Raum zu erreichen. Anziehende und abstoßende Kräfte werden wie beim der Spring Embedder Variante von Fruchterman und Reingold [62] so definiert, dass keine Quadratwurzeln berechnet werden müssen, und alle Berechnungen werden mit Integer-Arithmetik ausgeführt. Zusätzlich werden die anziehenden Kräfte mit $\Phi = 1 + \frac{d_G(v)}{2}$ gewichtet und so dafür gesorgt, dass Knoten mit hohem Grad langsamer bewegt werden. Um zu verhindern, dass die Komponenten eines nicht zusammenhängenden Graphens beliebig weit auseinander driften, wird zusätzlich eine Gravitationskraft

$$f_{\mathsf{grav}}(p_u, p_v) = \Phi(v) \cdot \gamma \cdot \left(\frac{\zeta}{|V|} - p_v \right)$$

eingeführt, die die Knoten in Richtung auf das Barycenter $\zeta = \sum_{w \in V} p_w$ aller Knoten bewegt. Der Einfluss der Gravitation wird dabei über die Gravitationskonstante γ gesteuert. Außerdem wird wie beim Simulated Annealing eine zufällige Kraft f_{random} berücksichtigt, um nicht in lokalen Minima stecken zu bleiben.

Weiterhin hat jeder Knoten seine eigene, unabhängige Temperatur, mit der die Stärke der Bewegung des Knotens kontrolliert wird. Die Iteration wird in GEM solange ausgeführt, bis die durchschnittliche lokale Temperatur aller Knoten unter einer vorzugebenden Minimaltemperatur gesunken ist oder bis die vorgegebene maximale Anzahl von Iterationsschritten erreicht wurde. In jedem Iterationsschritt werden die Knotenpositionen in zufälliger Reihenfolge der Knoten neu berechnet, wobei die oben beschriebenen Kräfte f_{rep}, f_{spring}, f_{grav} und f_{random} die Richtung der Bewegung, und die lokale Temperatur des Knotens die Länge der Bewegung bestimmen.

Eingabe : Graph $G = (V, E)$
zufällige Startpositionen $p = \left(p_v^{(0)}\right)_{v \in V}$
lokale Temperatur $T = \left(T_v^{(0)}\right)_{v \in V}$
Maximaltemperatur T_{max}
Minimaltemperatur T_{min}
Wichtung Oszillation $\sigma_O \geq 1$
Wichtung Rotation $\sigma_R \in (0, 1]$
Anzahl n_{iter} der auszuführenden Iterationen
Ausgabe : Layout $L = (G, p)$

for $v \in V$ **do**
  $dir_v^{(0)} = \vec{0}$ // Schiefmaß

for $t \leftarrow 1$ ***to*** n_{iter} **do**
  if $\sum_i T_v^{(t-1)} \leq T_{\mathsf{min}}$ **then**
    break
  for $v \in V$ **do**

$$F_v(t) \leftarrow \sum_{u \in V} f_{\mathsf{rep}}(p_u^{(t-1)}, p_v^{(t-1)}) + \sum_{u\,:\,\{v,w\} \in E} f_{\mathsf{spring}}(p_u^{(t-1)}, p_v^{(t-1)}) + f_{\mathsf{grav}}(p_v^{(t-1)}) + \Delta_{\mathsf{random}}$$

    $p_v^{(t)} \leftarrow p_v^{(t-1)} + F_v(t) \cdot \frac{T_v^{(t-1)}}{\|F_v(t)\|}$
    $\alpha \leftarrow \angle(F_v(t-1), F_v(t))$
    switch α **do**
      case $\cos\alpha \approx 1$ *(Verstärkung)*
        $T_v^{(t)} \leftarrow T_v^{(t-1)} + \cos\alpha \cdot \sigma_O$
      case $\cos\alpha \approx -1$ *(Oscillation)*
        $T_v^{(t)} \leftarrow T_v^{(t-1)} + \cos\alpha \cdot \sigma_O$
      case $\cos\alpha \approx 0$ *(Rotation)*
        $dir_v^{(t)} = dir_v^{(t-1)} + \sigma_R \cdot sgn(\sin\alpha)$
    $T_v^{(t)} \leftarrow \min\left(T_{\mathsf{max}}, T_v^{(t)} \cdot (1 - |\mathsf{dir}_v^{(t-1)}|)\right)$

Algorithmus 3: Graph Embedder Algorithmus von Frick u. a. [61]

Um die Zahl der Iterationen zu reduzieren, wird die Knotentemperatur gesenkt, wenn GEM eine uneffektive Bewegung wie Oszillation oder Rotation des Knotens entdeckt. Eine Oszillation wird dabei angenommen, wenn sich die Bewegungsrichtung eines Knotens zwischen zwei Iterationen in etwa umkehrt, d.h. wenn $\cos\alpha = \cos\angle(F_v(t-1), F_v(t)) \approx -1$. Eine Rotation lässt sich daran erkennen, dass aufeinanderfolgende Bewegungsrichtungen in etwa senkrecht zueinander sind ($\cos\alpha \approx 0$). Bleibt dagegen die Bewegungsrichtung eines Knotens zwischen zwei Iterationen in etwa gleich ($\cos\alpha \approx 1$), wird die Knotentemperatur erhöht und so eine (angenommene) zielgerichtete Bewegung verstärkt, s. Algorithmus 3.

GEM-3D [28] ist die dreidimensionale Version von GEM. Da die Kräfteberechnung in GEM keine wesentlichen Eigenschaften des zweidimensionalen Raumes nutzt, kann die Erweiterung einfach durch Zufügen einer weiteren Komponente für die dritte Dimension erfolgen. Dagegen muss die Erkennung von Oszillation und Rotation wesentlich geändert werden. Beispielsweise muss zum Erkennen der Änderung der Bewegungsrichtung das Modell des öffnenden Winkels auf ein Modell eines öffnenden Kegels erweitert werden. Besonders schwierig ist es, die Rotation im dreidimensionalen Raum zu erkennen, da diese in unendlich vielen Ebenen möglich ist. In GEM3D werden zur Vermeidung von Rotationen 3 Ansätze vorgestellt:

1. Es werden nur die Rotationen in den Projektionsebenen betrachtet,

2. man zählt die Anzahl der 90° Winkel,

3. Verwendung eines globalen Abkühlungsschemas anstatt expliziter Rotationserkennung.

Leider findet sich in [28] kein Vergleich der Qualität der drei Ansätze.

3.2.4 Hierarchische Layoutverfahren

Die ursprüngliche Idee stammt von Sugiyama u. a. [187]. Seitdem sind zahlreiche Variationen entwickelt worden. Das Grundprinzip ist in Algorithmus 4 veranschaulicht. Bei allgemeinen gerichteten Graphen, die gerichtete Kreise enthalten, müssen diese erst durch Änderung der Richtung von möglichst wenigen Kanten azyklisch gemacht werden[2]. Ungerichtete Graphen müssen in einen gerichteten azyklischen Graphen überführt werden, bevor der Algorithmus angewendet werden kann.

[2] ein NP-schwieriges Problem

Eingabe : gerichteter azyklischer Graph $G = (V, E)$
Ausgabe : hierarchisches Layout $L = (G, p)$

1 Schichtung: Zuweisung der Knoten zu Schichten → Vertikale Koordinatenzuweisung

2 Kreuzungsminimierung: Bestimmung von Knotenpermutationen innerhalb der Schichten, so dass wenige Kreuzungen entstehen

3 Horizontale Koordinatenzuweisung

Algorithmus 4: Hierarchisches Layout

Die Schichtenzuweisung ist in BibRelEx implizit durch das Erscheinungsjahr gegeben, so dass für BibRelEx nur die in den folgenden Unterabschnitten dargestellten Phasen 2 und 3 relevant sind. Dokumente ohne Angabe des Jahres müssen gesondert berücksichtigt werden, vgl. Abschnitt 5.6.

Kreuzungsminimierung

Aufgabe der Kreuzungsminimierung ist, die Knoten so innerhalb der Schichten anzuordnen, dass sich möglichst wenige Kantenkreuzungen ergeben. Dabei erhalten die Knoten relative Positionen. Das Problem der Kreuzungsminimierung ist selbst für Graphen mit nur zwei Schichten NP-vollständig [66]. Daher werden häufig Heuristiken zur Kreuzungsminimierung verwendet. Das Grundprinzip der gängigen Heuristiken ist der *Layer-by-Layer-Sweep* bei dem die Schichten rauf und runter traversiert werden. Dabei werden stets nur zwei benachbarte Schichten betrachtet. Die Permutation der zuvor besuchten Schicht wird fixiert und auf der aktuellen Schicht die Knoten mit dem Ziel weniger Kreuzungen permutiert.

BibRelEx bietet alle Heuristiken zur Kreuzungsminimierung an, die die verwendete Bibliothek AGD (Algorithms for Graph Drawing) [128] zur Verfügung stellt: Barycenter-, Median-, gewichtete Median-, Split- und Sifting-Heuristik. Exemplarisch soll hier nur die *Barycenter Heuristik* [187] vorgestellt werden. Sie basiert auf der intuitiven Annahme, dass in einer Darstellung eines Graphen mit wenigen Kreuzungen jeder Knoten nah zu seinen adjazenten Knoten angeordnet ist. Die Barycenter Heuristik erfreut sich großer Beliebtheit, da sie einfach zu implementieren ist, schnell läuft und gute Ergebnisse liefert.

Bei dieser Heuristik wird die Position eines Knoten als das Barycenter (Durchschnitt) der Positionen seiner Nachbarn bestimmt. Die Position $pos(v)$ eines Knotens v kann entweder durch die Knotenreihenfolge (üblicherweise

bei Verwendung in Phase 2) oder durch die x-Koordinaten der Knoten (Phase 3) festgelegt werden. Abschließend werden die Knoten nach diesen Werten sortiert. Die Laufzeit des Verfahrens beträgt $O(|E| + |V_2| \cdot \log |V_2|)$, wobei V_2 die Menge der Knoten in der Schicht sind, die in Algorithmus 5 permutiert wird.

Eingabe : gerichteter azyklischer Graph $G = (V, E)$
Knotenmengen V_1, V_2 zweier benachbarte Schichten
Ausgabe : Reihenfolge der Knoten in V_2, so dass minimale Zahl Kreuzungen zwischen V_1 und V_2

1 Berechne für alle $u \in V_2$:

$$avg(u) = \frac{1}{|N(u)|} \sum_{v \in N(u)} pos(v)$$

mit $N(u) = \{v \in V_1 | (v, u) \in E\}$ Menge der Nachbarn von u
und $pos(v) = index(v)$ bei Verwendung der Knotenreihenfolge
bzw. $pos(v) = \pi_1(v)$ bei Verwendung von x-Koordinaten.

2 Sortiere die $u \in V_2$ mit Sortierschlüssel $avg(u)$, bei Gleichheit beliebige Reihenfolge.

Algorithmus 5: Barycenter-Heuristik

Horizontale Koordinatenzuweisung

Nach der Schichtung und Kreuzungsminimierung steht die Topologie der Zeichnung fest. Die Aufgabe der 3. Phase ist nun, den Knoten so absolute horizontale Positionen zuzuweisen, dass die Knoten sich nicht überdecken, das Layout möglichst balanciert ist und dass vorzugsweise keine Kante durch einen Knoten hindurch verläuft. Das Problem der optimalen Koordinatenzuweisung ist ebenfalls NP-vollständig. Zur seiner Lösung können ähnliche Heuristiken wie in Phase 2 verwendet werden, beispielsweise die Barycenter-Heuristik angewandt auf absolute Koordinaten. Da in BibRelEx auch hier wieder die Lösung aus AGD übernommen wurde, soll dies nicht weiter ausgeführt werden.

3.2.5 Cluster-Verfahren

Mit Hilfe von Cluster-Verfahren können große Graphen übersichtlich dargestellt werden. Dabei werden Gruppen von zueinander in Beziehung stehenden

Objekten bzw. die sie repräsentierenden Knoten, zu einem „Super“-Knoten (Cluster) zusammengefasst und so der Detaillierungsgrad der Darstellung des Graphen reduziert. Der Benutzer bekommt eine „Zusammenfassung“ des Graphen angezeigt.

Eine Clusterung ist formal wie folgt definiert:

Definition 3.1 Clusterung

Eine Clusterung eines Graphen $G = (V, E)$ ist eine Menge $C = \{C_1, \ldots, C_k\}$ mit $C_i \subseteq V \ \forall i : 1 \leq i \leq k$ für die gilt $\bigcup_{1 \leq i \leq k} C_i = V$. Gilt darüber hinaus $C_i \cap C_j = \emptyset \ \forall i, j : 1 \leq i, j \leq k, i \neq j$ so spricht man von einer disjunkten Clusterung oder Partitionierung.

Dabei muss die Anzahl k der resultierenden Cluster je nach Verfahren vorgegeben werden oder wird während des Verfahrens automatisch bestimmt.

Bei der Clusterung können zwei unterschiedliche Prinzipien verfolgt werden. Zum einen können semantische Eigenschaften der den Knoten zugeordneten Objekte benutzt werden, um die zugehörigen Knoten zu Clustern zusammenzufassen. Zum anderen können bestimmte Eigenschaften des Graphen genutzt werden, um Knoten zusammenzufassen.

Semantikbasierte Clusterbildung

Üblicherweise erfolgt die Bildung von semantikbasierten Clustern in 3 Teilschritten, die in Algorithmus 6 dargestellt sind.

Eingabe : Graph $G = (V, E)$
Ausgabe : Clusterung C

1. Berechnung der Ähnlichkeit der Objekte untereinander.
2. Erstellung einer Ähnlichkeitsmatrix für alle Paare aus der Menge der zu clusternden Objekte.
3. Fusionierung der Objekte zu Clustern auf der Basis dieser Ähnlichkeit.

Algorithmus 6: Semantikbasierte Clusterbildung

Der erste Schritt ist sehr stark von der Art der Objekte, bzw. deren formaler Repräsentation abhängig. In jedem Fall muss ein Proximitätsmaß definiert werden.

Im Fusionierungsschritt (3) sind n Objekte in k Klassen aufzuteilen, wobei in der Regel k nicht bekannt ist. Die Klassenbildung soll so stattfinden, dass

die Cluster die zugrundeliegende Struktur der Objektmenge repräsentieren, d.h. die Beziehungen von Objekten innerhalb einer Klasse ist enger als die zwischen Objekten aus verschiedenen Klassen.

Die resultierenden Cluster hängen in erster Linie von der Aufbereitung der Daten, dem verwendeten Proximitätsmaß, und der Wahl des Fusionierungsalgorithmus ab.

Man unterscheidet zwei Arten der Proximitätsmaße. Ähnlichkeitsmaße spiegeln die Ähnlichkeit zwischen zwei Objekten wieder. Je größer das Ähnlichkeitsmaß zwischen zwei Objekten ist, desto ähnlicher sind sie. Abstandsmaße spiegeln die Unähnlichkeit zwischen zwei Objekten wieder. Je kleiner das Abstandsmaß zwischen zwei Objekten ist, desto ähnlicher sind sie. Es gibt eine Vielzahl von Proximitätsmaßen, die - abhängig von der Art der Objekte - in Cluster-Verfahren verwendet werden. Die Adäquatheit des Proximitätsmaßes ist wichtig für die Qualität der Clusterung.

Textbasierte Objekte, wie sie in BibRelEx verwaltet werden, werden üblicherweise durch Termvektoren repräsentiert. Im binären *Dokumenten-Vektormodell* wird das Vorhandensein oder Fehlen eines Terms durch die Einträge 1 bzw. 0 angezeigt. Im kontinuierlichen Modell werden diese durch positive ganze Zahlen (Termgewichte) ersetzt, die die Wichtigkeit des Terms für das Objekt reflektieren sollen. Für die Termgewichtung sind in der Vergangenheit zahlreiche Gewichtsfunktionen vorgeschlagen worden, wie zum Beispiel die Häufigkeit des Auftretens eines Terms, die Termspezifität oder die inverse Dokumentenfrequenz (TFIDF). Bevor nun Cluster gebildet werden können, muss ein geeignetes Proximitätsmaß auf den Dokumentenvektoren definiert werden. Ein sehr einfaches Maß ist der sogenannte *Simple-Matching Koeffizient*, der die Anzahl der gemeinsamen Terme zwischen zwei textbasierten Objekten misst [158]. Da die Länge des Dokumentenvektors stark das Proximitätsmaß beeinflusst, wurden erweiterte Maße entworfen, die die Größe des textbasierten Objekts bzw. die Länge des Dokumentenvektors berücksichtigen. Dazu gehören unter anderem der Dicesche Koeffizient, der Jaccard Koeffizient und der Kosinuskoeffizient [158]. Darüber hinaus gibt es Proximitätsmaße, die auf einem probabalistischen Modell basieren. Sie messen die Ähnlichkeit zwischen zwei Objekten über das Ausmaß der Abweichung der Verteilung der Objekte von einer stochastischen Unabhängigkeitsverteilung, z.B. [125].

Anhand ihrer Vorgehensweise unterscheidet man zwei große Gruppen von Fusionierungsverfahren: hierarchische und nicht-hierarchische. Hierarchische Verfahren erzeugen durch fortgesetztes Teilen oder Zusammenfügen von Clustern eine Hierarchie von Clustern. Nicht-hierarchische Verfahren unterteilen dagegen die Knotenmenge in einem einzigen Schritt in Cluster. Wenn keine Überschneidungen von Clustern erlaubt sind, nennt man die nicht-

hierarchischen Verfahren auch partitionierende Verfahren. Daneben gibt es z.B. noch Graphentheoretische Verfahren, die im nächsten Abschnitt vorgestellt werden, Fuzzy-Verfahren und optimierende Verfahren, die hier nicht weiter betrachtet werden, da sie keine Anwendung in BibRelEx finden. Eine Übersicht über verschiedene Cluster-Verfahren und neuere Entwicklungen findet sich in [57].

Sowohl die partitionierenden als auch die hierarchischen Cluster-Verfahren können algorithmisch gesehen auf *agglomerative* Art arbeiten: Ausgehend von einer Menge von Clustern, die aus jeweils einem Objekt bestehen, werden in jedem Schritt zwei Cluster zusammengelegt. Das Verfahren endet spätestens, wenn alle Objekte zu einem Cluster zusammengeschmolzen sind. Die *divisiven* (zerteilenden) Clusterverfahren beginnen demgegenüber mit einem Cluster, der alle Objekte enthält und spalten diese schrittweise in kleinere Cluster auf. Das Verfahren endet spätestens wenn alle Cluster nur noch einelementig sind.

Die bei hierarchisch agglomerativen Clusteralgorithmen resultierenden Cluster können einen unterschiedlichen Detaillierungsgrad aufweisen, je nach der Art mit der der Abstand zweier Cluster voneinander bestimmt wird:

- **Single Linkage:** Nimmt die kürzeste Distanz zwischen den Objekten zweier Cluster als Abstand der Cluster. Dieses Verfahren erzeugt in der Regel eine geringe Anzahl großer Cluster, deren Elemente sich teilweise erheblich unterscheiden können, weil es zur Aufnahme eines weiteren Elementes in den Cluster genügt, dass ein Element des Clusters in der Nähe dieses Elements liegt [168].

- **Complete Linkage:** Interpretiert die größte Distanz zwischen den Objekten zweier Cluster als Abstand der Cluster. Complete Linkage resultiert in der Regel in vielen kleinen Clustern, deren Elemente sehr ähnlich sind. Die Berechnung ist allerdings aufwendiger als bei Single Linkage [44].

- **Average Linkage:** Der Abstand zweier Cluster wird durch das arithmetische Mittel der Distanzen zwischen allen Paaren von Objekten des einen Clusters zu denen des anderen Clusters bestimmt. Die hiermit erzeugten Cluster sind in der Regel gleichmäßig verteilt, von mittlerer Größe und stabil gegenüber Veränderungen wie das Einfügen neuer oder das Löschen alter Objekte [195].

- **Centroid-Methode:** Die Cluster werden durch ihren Zentroiden bestimmt, einen künstlichen Repräsentanten, gebildet durch Mittelung über alle Objekte des Clusters [182].

- **Ward-Verfahren:** Vereinigt diejenigen Objekte/Cluster, die die Fehlerquadratsumme[3] in einem Cluster möglichst wenig erhöhen. Das Ward-Verfahren neigt dazu, möglichst gleich große (homogene) Cluster zu bilden und ist nicht in der Lage, langgestreckte Gruppen oder solche mit kleiner Elementzahl zu erkennen [52].

Die hierarchischen Cluster-Verfahren haben den Nachteil, dass alle paarweisen Ähnlichkeiten bzw. Abstände bekannt sein müssen, was in der Regel bei der Clusterung von n Objekten mit einem Aufwand der Größenordnung $O(n^2)$ verbunden ist. Demgegenüber verwenden die nicht-hierarchischen Verfahren Heuristiken zur Aufteilung der Objektmenge und sind im allgemeinen schneller als die hierarchischen Verfahren. Dafür garantieren sie nicht die Vorzüge gleichmäßiger Clusterbelegung und der Stabilität gegenüber Änderungen.

Bei den nicht-hierarchischen (partitionierenden) Verfahren werden die Daten *auf einmal* in Cluster eingeteilt und diese eventuell noch verändert. Ein Beispiel für ein nicht-hierarches Verfahren ist die *Single Pass* Methode. Bei dieser Methode wird zunächst ein beliebiges Objekt ausgewählt, das den ersten Cluster bildet. Die weiteren Objekte werden dann in willkürlicher Reihenfolge mit den bestehenden Cluster verglichen und dem Cluster mit der größten Ähnlichkeit zugewiesen, wenn ein bestimmter Schwellwert überschritten wird. Andernfalls wird mit dem Objekt ein neuer Cluster gebildet. Der Nachteil dieses Vorgehens liegt darin, dass am Anfang tendenziell größere Cluster gebildet werden als im späteren Stadium des Durchlaufs. Außerdem hängt die Clusterbildung von der Reihenfolge der Dokumente ab [169].

Graphentheoretische Verfahren

Die graphentheoretischen Cluster-Verfahren verwenden nur die Struktur des Graphen und nicht die den Kanten und Knoten zugeordnete Information. Sie versuchen Eigenschaften des Graphen wie den Kantenzusammenhang oder vollständige Untergraphen (Clique) zu identifizieren.

Stein und Niggemann [183] haben ein neues Strukturmaß, die sogenannte *gewichtete partielle Konnektivität* eingeführt, dessen Maximierung zu Clustern führt, die die „natürliche Struktur“ des Graphen wiederspiegeln.

[3]Fehlerquadratsumme: Summe der quadrierten Abstände der Objekte des Clusters zum Zentroiden des Clusters.

Definition 3.2 Gewichtete partielle Konnektivität

Sei $G = (V, E)$ ein ungerichteter Graph und $C = (C_1, \ldots, C_n)$ eine Zerlegung von G. Die gewichtete partielle Konnektivität von C, $\Lambda(C)$, ist definiert durch

$$\Lambda(C) := \sum_{i=1}^{n} |C_i| \cdot \lambda_i$$

wobei λ_i die gewichtete Kantenkonnektivität des durch C_i induzierten Subgraphen $G(C_i)$ ist.

Dabei bezeichnet die gewichtete Kantenkonnektivität eines Graphen die minimale Summe der Kantengewichte derjenigen Kanten, die man aus dem Graphen entfernen muss, um einen nicht zusammenhängenden Graphen zu erhalten.

Abbildung 3.2 zeigt ein Beispiel für die Berechnung der gewichteten partiellen Konnektivität.

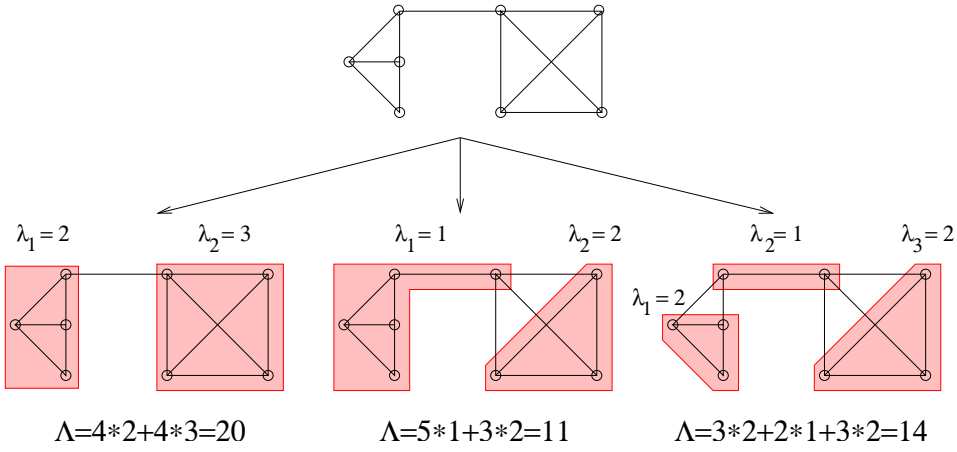


Abbildung 3.2: Beispiel für eine Clusterung mit dem MajorClust-Verfahren von Stein und Niggemann [183]

Leider ist bisher kein schnelles Verfahren bekannt, mit dem eine Graphzerlegung mit maximalem Λ berechnet werden kann. In [183] wird aber ein schnelles Verfahren, MajorClust, vorgestellt, das eine lokale Heuristik zur Maximierung von Λ verwendet, siehe Abbildung 3.2.

Für ungewichtete Kanten geht MajorClust folgendermaßen vor: Anfangs wird jedem Knoten sein eigener Cluster zugewiesen. Anschließend werden die Cluster schrittweise umorganisiert. In jedem Umorganisationsschritt wird für jeden Knoten derjenige Cluster bestimmt, der die meisten zu ihm adjazenten Knoten enthält und der Knoten ggf. in diesen verschoben. Gibt es für einen

Knoten mehrere Cluster mit maximaler Zahl zu ihm adjazenter Knoten, so wird einer davon zufällig ausgewählt. Der Algorithmus terminiert, wenn in einem Umorganisationsschritt kein Knoten mehr verschoben wird. Wichtig ist, dass die Knoten in jedem Umorganisationsschritt in zufälliger Reihenfolge durchlaufen werden. Der Algorithmus kann leicht auf gewichtete Kanten erweitert werden, vgl. Algorithmus 7.

Eingabe : Graph mit Kantengewichtung $G = (V, E, w)$
Ausgabe : Eine Funktion $c : V \rightarrow N$ die jedem Knoten eine Clusternummer zuordnet.

$n = 0$, $t =$ false
for $v \in V$ **do**
  $n = n + 1$
  $c(v) = n$
while $t =$ *false* **do**
  $t =$ true
  $V' = V$
  for $i \leftarrow 1$ ***to*** $|V|$ **do**
    wähle zufälliges $v \in V'$
    $C = \{i | \sum_{\{u,v\} \in E, c(u)=i} w_{u,v}$ ist maximal $\wedge\, i \neq c(v)\}$
    if $C \neq \emptyset$ **then**
      wähle zufälliges $c^* \in C$
      $c(v) = c^*$
      $t =$ false
    $V' = V' \setminus \{v\}$

Algorithmus 7: MajorClust [143]

MajorClust terminiert spätestens nach $O(|V||E|)$ Schritten [143]. Aufgrund dieses guten Laufzeitverhaltens ist MajorClust gut geeignet für große Graphen, wie sie in BibRelEx vorkommen, und ist in diesem Fall Cluster-Verfahren, die auf dem Minimum-Cut und Nearest-Neighbor Strategien basieren, überlegen [183]. Darüber hinaus haben Experimente in verschiedenen Anwendungsbereichen wie Visualisierung des Verkehrs in Rechnernetzen, Visualisierung von Wissensbasen zur Konfiguration technischer Systeme, Visualisierung tabellarischer Daten und verschiedenen Aufgabenmodellen in Unternehmensnetzwerken, gezeigt, dass MajorClust den Problemen angepasste Partitionierungen findet [131, 143].

3.3 Die bibliographische Datenbasis GeomBib

Die bibliographische Datenbasis GeomBib enthielt ca. 8700 bibliographische Einträge aus dem Bereich Algorithmische Geometrie, als unser Projekt startete. Sie referenziert Zeitschriftenartikel, Tagungsbeiträge und Technische Berichte. Als Format wird BibTEX verwendet; dies trägt mit zur großen Beliebtheit von GeomBib bei, denn die Produktion eigener Literaturverzeichnisse in LaTeX-Dokumenten wird hierdurch sehr erleichtert.

GeomBib wird gegenwärtig von B. Jones an der University of Saskatchewan unterhalten [55] und in gemeinschaftlicher Aktivität aktualisiert. Üblicherweise wird die Datenbasis, die aus einer einzelnen Datei besteht, von den Benutzern via FTP geladen. Zu einer Kopie dieser lokalen Version können die Benutzer nun neue Einträge verschiedener Publikationen zufügen oder unvollständige oder fehlerhafte Einträge korrigieren. Nach vier Monaten wird die so entstandene Datei mit der Originaldatei mit dem Unix-Befehl *diff -c* verglichen und die Unterschiede via Email an den Administrator gesendet. Aus allen so eingegangenen Aktualisierungsvorschlägen wird eine neue Version von GeomBib erstellt. Geänderte Einträge werden dabei mit einem *update*-Feld versehen, das das Datum und den Absender der Änderung enthält. Etwa einen Monat später wird die neue GeomBib-Version den Nutzern über FTP zur Verfügung gestellt. Dieser Aktualisierungs-Zyklus ist in Abbildung 3.3 schematisch dargestellt.

Dank der Aktivität der Computational Geometry Community ist so in den letzten Jahren eine sehr aktuelle und reichhaltige Literaturdatenbank entstanden.

In GeomBib wird jeder Eintrag über einen Schlüssel identifiziert, der nach festen Regeln aus den Zunamen der Autoren, dem Titel und dem Erscheinungsjahr generiert wird. Die genauen Regeln können der README-Datei, die jeder GeomBib-Distribution beigefügt ist, entnommen werden.

In GeomBib sind bereits für jedes Dokument optional die Felder *cites, precedes, succeeds* und *annote* vorgesehen. In das *cites*-Feld werden die Schlüssel der zitierten Dokumente eingetragen. Diese Liste braucht nicht vollständig zu sein. Vollständige Listen werden durch anhängen von ZZZ als solche markiert. In das *annote*-Feld können direkt Bemerkungen angegeben werden.

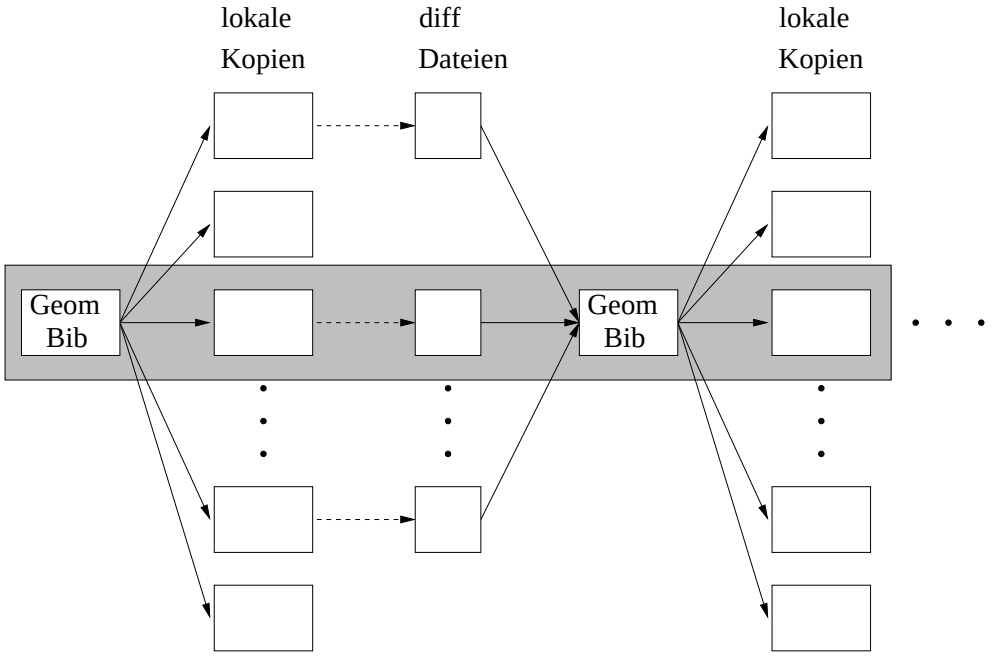


Abbildung 3.3: Periodische Aktualisierung von GeomBib

Nachdem wir uns in den bisherigen Kapiteln mit dem Stand der Forschung und den Grundlagen für BibRelEx beschäftigt haben, wollen wir nun im nächsten Kapitel an konkreten Anwendungsbeispielen die Möglichkeiten von BibRelEx aufzeigen.

Kapitel 4

Nutzungsbeispiele

Um die Möglichkeiten von BibRelEx deutlich zu machen, stellen wir in diesem Kapitel einige Nutzungsbeispiele vor. Prinzipiell lassen sich zwei verschiedene Anwendungsgebiete für BibRelEx ausmachen: Die Darstellung von Wissensgeflechten führt eher zu kleinen, dichten Netzen. Als Beispiele hierfür sind im Folgenden die Organisation einer Lehrveranstaltung (Abschnitt 4.1), der Einsatz in Lernumgebungen (Abschnitt 4.2), der Begutachtungsprozess (Abschnitt 4.3) und die Literaturverwaltung einer wissenschaftlichen Arbeit (Abschnitt 4.4) ausgeführt. Das zweite Anwendungsgebiet ist die Recherche in großen Datenbanken. Hier ergeben sich eher große, dünne Geflechte.

4.1 Organisation eines Seminars

Ein wissenschaftlicher Mitarbeiter soll ein Seminar über ein bestimmtes Gebiet organisieren. Im Idealfall ist das Wissen darüber, welche Dokumente in das Gebiet einführen, schon in der Datenbasis vorhanden und kann für die Organisation des Seminars wieder verwendet werden.

Ist dieses Wissen nicht in der Datenbasis enthalten, so muss der Mitarbeiter mittels Recherche geeignete Arbeiten suchen und ggf. in die Datenbasis einfügen. Bei der Durchsicht der Arbeiten ordnet er die Arbeiten nach verschiedenen Kriterien, z.B. „Introductory“, „Experienced“, „Advanced“, um sie nach dem zum Verständnis benötigten Wissen zu kategorisieren, und „Recommended as Primary Reading“, „Background Reading“ und „Useless“, um ihre Qualität in Bezug auf das Seminarthema zu beurteilen. Dieses Wissen kann auf drei Arten in die Datenbasis eingebracht werden:

1. **Wissensaggregation mittels Annotationen:** Für jedes Kriterium wird eine Annotation angelegt und diese an dem entsprechenden Dokument angebracht.

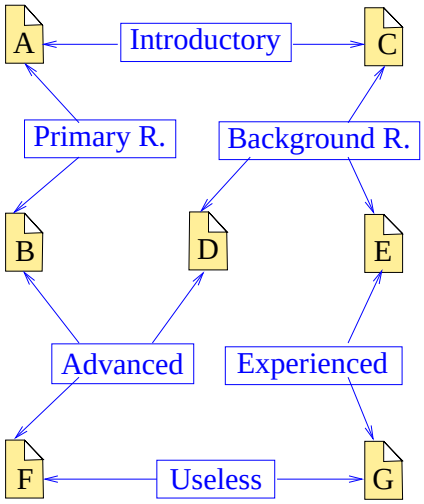
2. **Wissensaggregation mittels typisierten Links:** Für das Seminar wird ein Dokument angelegt, das die Seminarbeschreibung enthält und die zugehörigen bibliographischen Daten in die Datenbasis eingegeben. Anschließend werden Links von der Seminarbeschreibung zu den Arbeiten erzeugt und den Links das entsprechende Kriterium als Typ zugewiesen.

3. **Wissensaggregation mittels annotierten Links:** Wie im zweiten Fall wird ein Dokument für das Seminar erzeugt und über Links mit den Arbeiten verknüpft. An die Links werden dann Annotationen mit den entsprechenden Kriterien angebracht.

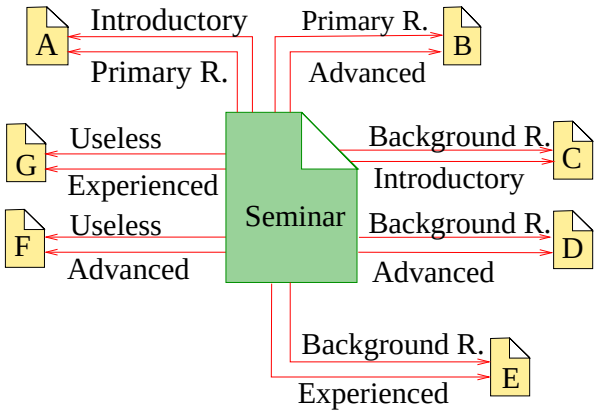
Es kann auch Arbeiten geben, auf die mehrere der Kriterien zutreffen. Beispielsweise kann eine Arbeit, die als „Recommended as Primary Reading“ kategorisiert wurde, den Wissenstand „Advanced“ erfordern. Bei der Wissensaggregation mittels Annotationen erhält eine solche Arbeit zwei Annotationen, bei der Wissensaggregation mittels typisierten Links verweisen zwei Links auf sie und bei der Wissensaggregation mittels annotierten Links verweist ein Link mit 2 Annotationen auf sie. Die Möglichkeiten der Wissensaggregation sind in Abbildung 4.1 exemplarisch dargestellt. Die Dokumente sind mit den Buchstaben A bis G gekennzeichnet. Die Annotationen werden durch blaue Rechtecke repräsentiert. Links, die Annotationen mit Dokumenten verbinden (AnnotationLinks), sind durch blaue Kanten dargestellt, (typisierte) Links durch rote Kanten.

Mit Hilfe von Anfragen an die Datenbasis und der Speichermöglichkeit von Suchfunktionen kann so abschließend das Material für jedes Thema extrahiert werden und an die entsprechenden Studenten verteilt werden.

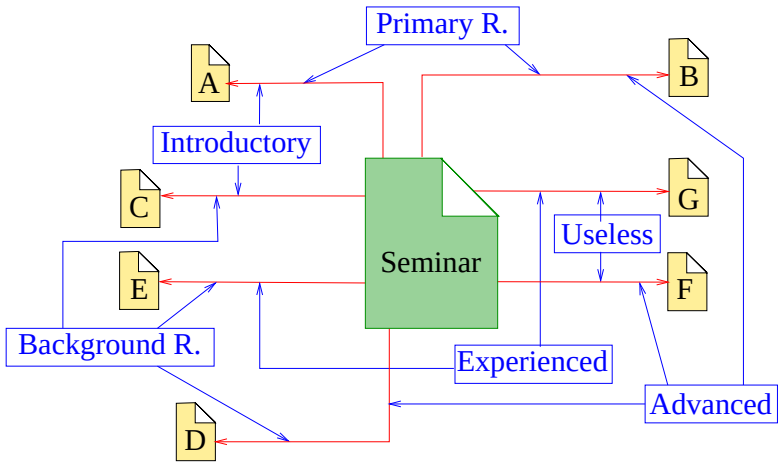
Betrachtet man Anfragen an die Datenbasis, um z.B. alle einführenden Arbeiten für das Seminar zu finden, zeigt sich, dass bei der Wissensaggregation mittels Annotationen die Zuordnung zum Seminar fehlt. Man würde alle Dokumente mit einer Annotation „Introductory“ als Ergebnis erhalten, auch wenn diese gar nicht zu dem geplanten Seminar gehören. Im Fall der Wissensaggregation mittels typisierten Links kann man nach Links vom Typ „Introductory“ und Ziel „Dokument Seminarbeschreibung“ suchen und findet so die gewünschten Arbeiten. Das Problem im ersten Fall kann man durch das Anbringen einer Annotation „gehört zu Seminar“ an allen Arbeiten zum Seminar lösen. Man kann nun nach allen Dokumenten suchen, auf die eine Annotation „Introductory“ und eine Annotation „gehört zu Seminar“ verweist. Da für den Fall der typisierten Links auch ein Dokument erzeugt werden musste (Dokument Seminarbeschreibung) sind beide Lösungen bzgl. des Aufwands zum Einbringen des Wissens gleichwertig. Die dritte Möglichkeit, die Organisation des Seminars in die Datenbasis zu integrieren, ist sowohl in



(a) mittels Annotationen



(b) mittels typisierten Links

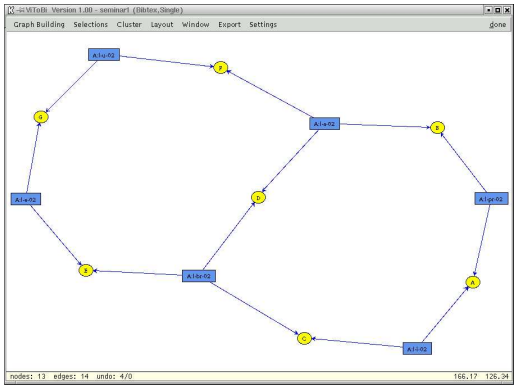


(c) mittels annotierten Links

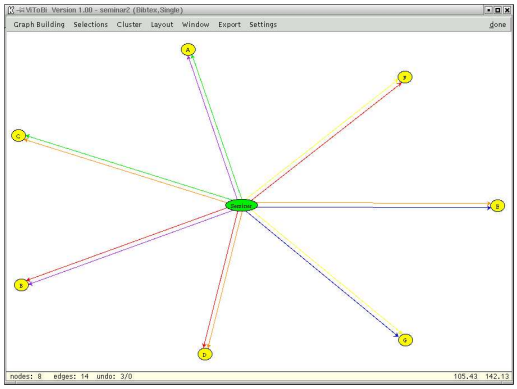
Abbildung 4.1: Aggregation von Wissen

Bezug auf das Einbringen als auch in Bezug auf Anfragen aufwendiger als die ersten beiden Möglichkeiten.

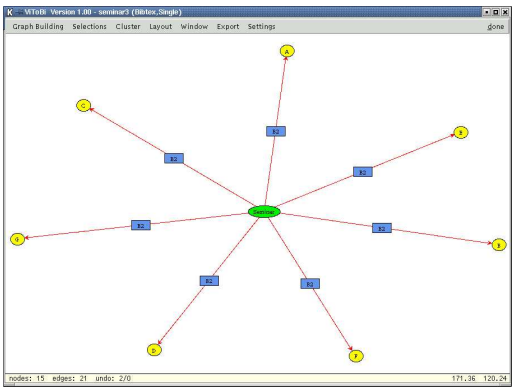
Abbildung 4.2 zeigt, wie die entsprechenden Beziehungsgeflechte in BibRelEx dargestellt werden. Die Dokumente sind durch Kreise bzw. Ellipsen dargestellt und mit $A, \ldots, G$ bzw. *Seminar* für das Dokument mit der Seminarbeschreibung beschriftet. Den Kanten und Dokumenten können in BibRelEx abhängig vom Typ unterschiedliche Farben zugeordnet werden. So sind beispielsweise AnnotationLinks in der Abbildung blau dargestellt und „Introductory"-Links grün. Annotationen an Links werden direkt auf der zugehörigen Kante positioniert und nicht untereinander verbunden. So geht zwar einerseits die Zuordnung verloren, welche Kanten dieselbe Annotation tragen, dafür ist die Darstellung aber wesentlich übersichtlicher, wie man leicht durch Vergleich der beiden Abbildungen 4.1(c) und 4.2(c) sieht. Für eine bessere Übersichtlichkeit werden darüberhinaus bei Kanten, die mehr als eine Annotation tragen, die Annotationen nicht einzeln dargestellt, sondern die Anzahl der zugehörigen Annotationen in die Knotenbeschriftung aufgenommen. Durch Doppelclick mit der Maus kann sich der Benutzer alle Annotationen anzeigen lassen.



(a) mittels Annotationen



(b) mittels typisierten Links



(c) mittels annotierten Links

Abbildung 4.2: Aggregation von Wissen in BibRelEx

4.2 Lernumgebungen

Das Vorgehen aus dem letzten Abschnitt bei der Organisation eines Seminars kann für beliebige Lehrveranstaltungen eingesetzt werden. Ebenso eignen sich die Kriterien für den Einsatz in Lernumgebungen, um die einzelnen Lerninhalte zu kategorisieren und durch entsprechende Filterbedingungen dem Lernenden nur die Inhalte zu präsentieren, die seinem Wissensstand und Lehrziel entsprechen. Beispielsweise können einem Studienanfänger, der sich erstmal nur einen groben Einblick in einem Wissensgebiet verschaffen möchte, alle Arbeiten mit den Kriterien „Introductory“ und „Primary Reading“ präsentiert werden.

In einer Lernumgebung ist es außerdem wichtig, die Lehrinhalte in eine logische Reihenfolge zu bringen und auch unterschiedliche Lehrpfade anzubieten. Hierfür sind Beziehungen der Art

X **need** Y**:** X setzt Wissen um Y voraus, bzw. Y wird gebraucht um X zu erlernen;

X **define** Y**:** Lernziel X verlangt vom Studenten Lehrinhalt Y definieren zu können.

nützlich. Diese Beziehungen können mit BibRelEx wieder leicht mit Hilfe von Annotationen und/oder Links realisiert werden.

Beziehungen können auch genutzt werden, um den Lernenden unterschiedliche Blickwinkel durch Guided Tours zu bieten. Möchte ein Dozent etwa, dass bestimmte Lerneinheiten in einer von ihm gewählten Reihenfolge für seine Vorlesung verwendet werden, kann er dazu Querverweise vorsehen, die die Lernmodule in der von ihm gewünschten Anordnung verbinden. Durch die Trennung der Dokumente bzw. ihrer Beschreibungsdaten und der inhaltlichen Beziehungen und Annotationen, siehe Abschnitt 5.4.2, können leicht kontextabhängige Touren realisiert werden und wird die Wiederverwendung von Lerneinheiten beispielsweise in verschiedenen Vorlesungen unterstützt. Die Trennung privater und öffentlicher Informationen in BibRelEx ermöglicht darüber hinaus den einzelnen Lernenden die Vorlesung mit privaten Notizen zu versehen.

Die einzelnen Lerneinheiten können in BibRelEx als Hypertexte organisiert werden, da BibRelEx direkt das Laden von Hypertexten mit Hilfe von Netscape unterstützt. Der Lernende kann dazu zunächst aus dem Wissensgeflecht die für ihn relevante Lehreinheit auswählen und das zugehörige Fenster mit den Metadaten (Dokumentbeschreibung) öffnen. Mit Hilfe des Kontextmenüs kann er dann direkt die Lerneinheit laden. Da damit die volle

Funktionalität des Browsers zur Verfügung steht, kann die bisher in BibRelEx noch fehlende Komponente des verteilten Arbeitens beispielsweise mit Hilfe von Javascript ergänzt werden. BibRelEx kann damit auch einfach auf Lernumgebungen, die bereits auf Hypertextbasis realisiert sind, aufgesetzt werden.

Das Beispiel soll hier nicht weiter ausgeführt werden, sondern nur einen ersten Einblick vermitteln, wie man mit Hilfe geeigneter Kriterienkataloge und der BibRelEx zugrunde liegenden Ideen Lernumgebungen gestalten kann. Durch den modularen Aufbau von BibRelEx kann es leicht in Lernumgebungen integriert werden bzw. eignet sich selbst als Grundlage für die Entwicklung einer Lernumgebung, kann aber auch - wie wir eben gesehen haben - direkt mit existierenden Hypertext-Lernumgebungen kombiniert werden. Die zur Realisierung der Benutzungsoberfläche verwendete Bibliothek Qt, siehe Abschnitt 6.6, bietet darüber hinaus die Möglichkeit Netscape-Plugins zu erstellen. Damit kann die Benutzungsoberfläche von BibRelEx auch ohne grösseren Aufwand zu einer WWW-Applikation erweitert werden.

4.3 Begutachtung

Unter Peer-Review-Verfahren versteht man den Begutachtachtungsprozess durch unabhängige Experten für einen bestimmten Bereich, den sogenannten „Peers“. Peer-Review-Verfahren werden beispielsweise zur Begutachtung von Manuskripten vor Ihrer Publikation, zur Auswahl von Konferenzbeiträgen oder für die Entscheidung über die Allokation von Forschungsmitteln eingesetzt. Sie dienen also der qualitativen Evaluation von Forschungsarbeiten und sind eine Form von organisationeller Wissensgenerierung.

BibRelEx unterstützt sowohl anonyme Begutachtung als auch offenes Reviewing. Im ersten Fall werden die Bewertungen von einzelnen anonymen Begutachtern unabhängig voneinander und nicht öffentlich geschrieben. Sie können in BibRelEx mit Hilfe private Beiträge umgesetzt werden, die von einer zentralen Stelle, z.B. eine Zeitschriftenredaktion oder einem Programmkomitee für eine wissenschaftliche Konferenz, gesammelt werden. Die automatische Aktualisierung von Datenbasen in BibRelEx vereinfacht diesen Prozess. Im zweiten Fall werden die Bewertungen öffentlich geschrieben und die Autoren und Gutachter kennen sich. Die Beiträge können durch öffentliche Annotationen und Beziehungen realisiert werden. Durch annotieren von Annotationen werden sowohl Diskussionen als auch ein mehrstufiges Reviewing unterstützt.

Die unterschiedlichen Disziplinen und Publikationsarten (Zeitschriftenartikel, Konferenzbeitrag, etc.) verlangen jeweils eigene Kriterienkataloge. In

BibRelEx können jedem Begutachter die Bewertungskriterien als Annotationen oder Linktypen (wie im Beispiel 4.1 *Organisation eines Seminars*)) zur Verfügung gestellt werden. Die Linktypen werden dabei über eine Konfigurationsdatei festgelegt.

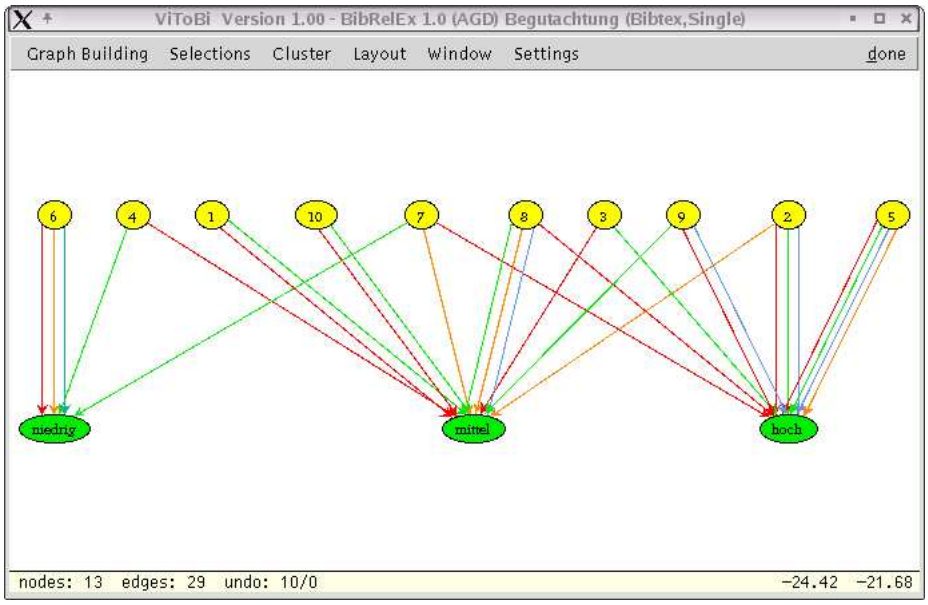
Beispiele für formale allgemeine Kriterien sind

- Informationsniveau und -gehalt,
- wissenschaftlichen Bedeutung,
- Originalität,
- mögliche Anwendungsrelevanz der erzielten Ergebnisse,
- Qualität der Darstellung (übersichtlich, angemessen Länge),
- inhaltliche Erschließung (Breite und Tiefe der Erschließung),
- Aufbereitung der Information (sachlich, korrekt, aktuell, angemessen, vollständig),
- Zusatz-/Hintergrundinformationen.

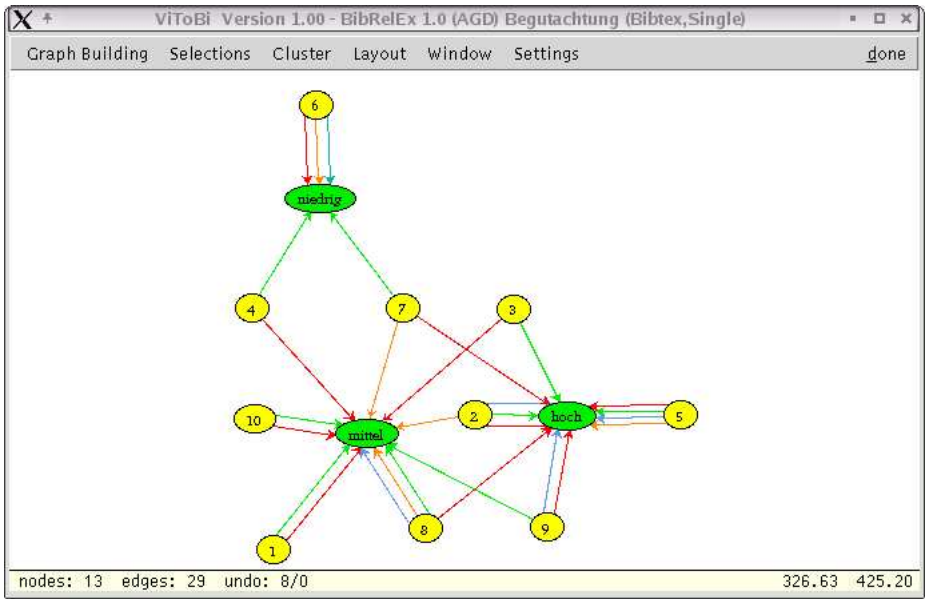
Alternativ kann man die Attribute einer Bewertung, wie *niedrig, mittel, hoch* als Dokumente anlegen und die zu bewertenden Dokumente mit Links, deren Typen die Bewertungskriterien wie *Informationsgehalt* wiedergeben, mit den Bewertungen verknüpfen. Durch Annotieren der Links können die jeweiligen Bewertungen zusätzlich näher erläutert werden. Die Dokumente werden dann in der Visualisierung sowohl bei den hierarchischen Layout Verfahren als auch bei den kräftbasierten Verfahren entsprechend Ihrer Bewertung angeordnet, siehe Abbildung 4.3. Das hierarchische Layout hat hier den Vorteil, dass bedingt durch die Mittelwertbildung bei der Kreuzungsminimierung, siehe Abschnitt 3.2.4, die Dokumente entsprechend ihrer durchschnittlichen Bewertung linear angeordnet werden und somit die resultierende Bewertung leicht zu erkennen ist.

Häufig wird ein Manuskript von mehreren Gutachtern unabhängig voneinander begutachtet. In BibRelEx können die verschiedenen Gutachten mit Hilfe der automatischen Aktualisierung zusammengeführt werden. Allerdings ist es in BibRelEx (noch) nicht möglich, die einzelnen Kriterien selbst oder abhängig von den Gutachtern (je nach dessen Kenntnisstand im Gebiet der begutachteten Arbeit) unterschiedlich zu gewichten oder explizit ein Gesamtqualitätsmass aus den einzelnen Bewertungen zu bilden[1].

[1]Implizit ist dies durch die hierarchische Darstellung, vgl. Abbildung 4.3(a), möglich.



(a) hierarchisches Layout



(b) Spring Embedder

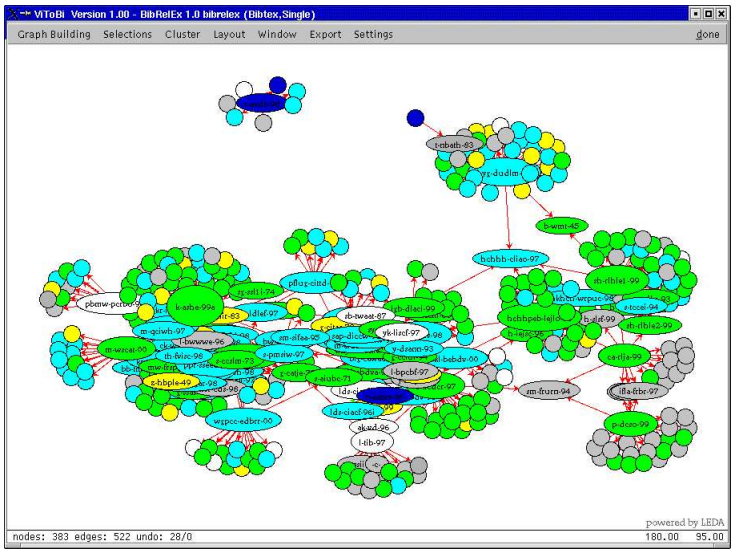
Abbildung 4.3: Begutachtung

4.4 Literaturverwaltung für wissenschaftliche Arbeiten

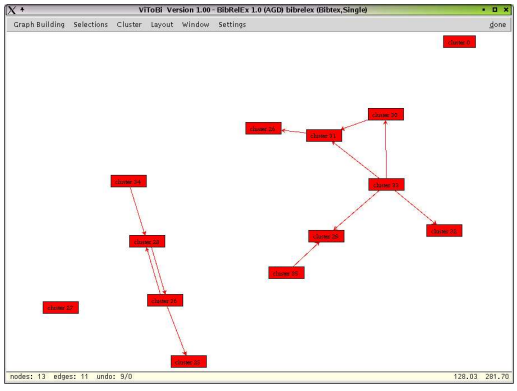
Bei der Erstellung von wissenschaftlichen Arbeiten oder der Durchführung von Forschungsprojekten ist häufig eine Vielzahl an Literatur zu lesen und zu verwalten. So beinhaltet beispielsweise die bibliographische Sammlung zum BibRelEx-Projekt mehr als 500 Literaturquellen. Diese wurden im Laufe des Projekts mit Notizen versehen und miteinander in Beziehung gebracht. BibRelEx war dabei eine große Hilfe, sowohl bei der Verwaltung der Literaturquellen, bei Ihrer Einordnung in den wissenschaftlichen Kontext als auch beim Schreiben dieser Arbeit. Durch die Notizen war ein schneller Zugriff auf die Literaturquellen möglich und stand ihre jeweilige Essenz ohne erneutes Anlesen zur Verfügung. Das konstruierte Wissensgeflecht war insbesondere bei der Beschreibung des Stands der Forschung nützlich, da mit Hilfe des Wissensgraphen zentrale und wichtige Arbeiten leicht zu lokalisieren waren und eine Übersicht des Forschungsgebietes zur Verfügung stand. Abbildung 4.4 zeigt einen Auszug aus dem Wissensgeflecht zu BibRelEx und die Möglichkeit der Strukturierung des Wissensraumes mit Hilfe der Clusterung.

Da das System erst im Laufe der Arbeit entstanden ist, konnte leider noch nicht auf vorhandenes Wissen zurückgegriffen werden. Hier sehen wir einen großen Nutzen für zukünftige Arbeiten und auch für den Einsatz in der Lehre. Das System kann Studierenden beispielsweise beim Erstellen von Seminar- oder Studienarbeiten helfen und unterstützt Gruppenarbeit. So kann z.B. aus den verschiedenen Seminararbeiten ein großes Wissensgeflecht erzeugt werden oder den Studierenden zur Einarbeitung in ein für sie neues Gebiet das zugehörige Wissensgeflecht an die Hand gegeben werden.

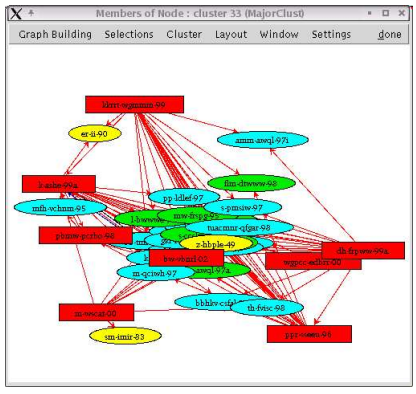
Beim Lesen von (neuen) Forschungsarbeiten hilft BibRelEx die Arbeit in den vorhandenen Wissensraum einzuordnen und verwandte Arbeiten zu finden oder Arbeiten, die das Problem von einem anderem Standpunkt aus beschreiben. Es können dabei auch neue interessante Zusammenhänge zu Tage treten, die bei der Suche nach weiterer Literatur hilfreich sein können. Bisher standen dem Benutzer dafür in den meisten Systemen nur textbasierte Suchmöglichkeiten zur Verfügung. In einigen wenigen Systemen konnte er noch direkte Zitierlinks verfolgen. Mit BibRelEx hat er zusätzliche beziehungsbasierte Suchmöglichkeiten, vgl. Abschnitt 5.5. Für die Einordnung neuer Arbeiten in die bestehende Datenbasis bietet die graphische Benutzeroberfläche von BibRelEx vielfältige Unterstützung, die die Eingabe von Beziehungen und Notizen stark vereinfachen. Zwei Dokumente können beispielsweise leicht mittels Drag&Drop zu einer Beziehung zusammengefügt werden. Darüberhinaus können Annotationen und Beziehungen auch inter-



(a) Auszug aus dem Wissensgeflecht zu BibRelEx



(b) nach Themen geclustert



(c) Cluster zum Thema Link Analyse

Abbildung 4.4: Wissensgeflecht zu BibRelEx

aktiv in der Visualisierung eingegeben werden.

Kapitel 5

Konzepte und Algorithmen

In diesem Kapitel erfolgt die Darstellung der Konzeption der verschiedenen Teilbereiche von BibRelEx, die sich grob in Datenhaltung, Recherche und Visualisierung einteilen lassen.

Bei der Datenhaltung spielt insbesondere die Sicherstellung der Konsistenz der Daten eine große Rolle. Da bisher viele Benutzer Daten in die Datenbasis einbringen konnten und kein System zur Verfügung stand, das den Benutzer dabei unterstützt hat, kam es in der Vergangenheit zu fehlerhaften und doppelten Einträgen. Um diese zu erkennen und zukünftig fehlerhafte Eingaben zu vermeiden, haben wir das Programm BibConsist entwickelt und in BibRelEx integriert. BibConsist wird in Abschnitt 5.1 vorgestellt.

Eine wesentliche Vereinfachung für den Benutzer bedeutet auch, die Aktualisierung von GeomBib weitestgehend zu automatisieren. Dies haben wir in BibRelEx mit Hilfe einer Dreiteilung des Datenbestandes erreicht, die in Abschnitt 5.2 beschrieben wird. Dieses Konzept lässt sich auch auf andere Datenbasen anwenden und bietet sich immer dann an, wenn innerhalb einer Community Daten gemeinsam verwaltet werden sollen.

Jeder Benutzer kann Dokumente, Anmerkungen und Beziehungen in die Datenbasis einbringen. Um die Qualität der Beiträge auch ohne einen separaten Begutachtungsprozess sicherstellen zu können, müssen Richtlinien festgelegt werden. Mögliche Regeln werden in Abschnitt 5.3 diskutiert.

BibRelEx unterscheidet sich von den bisher existierenden Informationsvisualisierungssystemen insbesondere durch seine Möglichkeit der Wissensaggregation. Im Abschnitt 5.4 wenden wir uns der Fragestellung zu, wie das Wissen in BibRelEx repräsentiert werden kann und so in die Datenbasis eingebracht werden kann, dass sowohl eine einfache Wissenswiederverwendung als auch eine Trennung des öffentlichen und privaten Datenbestands ermöglicht wird.

Nachdem nun die wesentlichen die Datenhaltung betreffenden Aspekte

behandelt wurden, ist der nächste Schritt die Nutzung des zusätzlich eingebrachten Wissens. Im Abschnitt 5.5 gehen wir insbesondere auf die strukturbasierten Suchmöglichkeiten von BibRelEx ein.

Abschließend wird in Abschnitt 5.6 die Visualisierung in BibRelEx behandelt. Hier werden die verschiedenen Layoutmöglichkeiten vorgestellt und gezeigt, wie mit Hilfe der Visualisierung die Recherche unterstützt werden kann.

5.1 Konsistenzprüfung

Aus der Entscheidung, eine existierende Datenbasis für BibRelEx zu verwenden, ergaben sich direkt zwei Konsequenzen für die Entwicklung von BibRelEx. Zum einen muss GeomBib in seiner ursprünglichen Form mit BibRelEx koexistieren können. Zum anderen müssen ausreichend viele Zitierinformationen und Annotationen bereitgestellt werden, damit die Geometriegemeinschaft die neue Erschließungsmethode vorteilhaft nutzen kann. In GeomBib sind bereits für jedes Dokument optional die Felder *cites, precedes, succeeds* und *annote* vorgesehen. Bis zum Projektbeginn von BibRelEx wurden aber nur in weniger als 10 Prozent der Datensätze diese Felder genutzt.

Wir haben deshalb GeomBib um Literaturverweise ergänzt, die in den Proceedings der großen Tagungen (SoCG, CCCG, etc.) vorkommen. Für die meisten Arbeiten existierten bereits Datensätze in GeomBib; ein Großteil der zitierten Arbeiten war dagegen bisher nicht in GeomBib berücksichtigt. Wir haben bis zur GeomBib-Version vom Juli 2001 über 3000 neue Einträge und 13000 Verweise beigesteuert.

Beim Einfügen neuer Datensätze stießen wir auf folgendes Problem: In GeomBib wird jeder Eintrag über einen Schlüssel identifiziert, der nach festen Regeln aus den Zunamen der Autoren, dem Titel und dem Erscheinungsjahr generiert wird. GeomBib selbst kann zwar neue Einträge zurückweisen, wenn dadurch ein doppelter Schlüssel entstünde; es kommt aber vor, dass Eingabefehler zu Duplikaten führen, deren Schlüssel verschieden sind, etwa dann, wenn die Autorennamen vertauscht wurden, oder bei falscher Schreibweise des Titels.

Um solche Inkonsistenzen erkennen zu können, haben wir das Programm *BibConsist* [106] entwickelt. Es gibt zu einem gegebenen Eintrag alle ähnlichen Einträge aus, d.h. alle solchen, bei denen die Mehrheit der korrespondierenden Felder ähnlich ist. Zwei Felder des gleichen Typs werden dabei als ähnlich angesehen, wenn die Mehrzahl der Worte in diesen Feldern phonetisch ähnlich sind. Numerische Felder wie *year*, *volume*, etc. werden auf Gleichheit verglichen. Die Felder *author* und *title* gehen doppelt gewichtet ein.

Die phonetische Ähnlichkeit von zwei Worten wird mit Hilfe eines modifizierten Soundex-Code bestimmt. Der Soundex-Algorithmus wurde ursprünglich von Margaret K. Odell und Robert C. Russell entwickelt und bereits im Jahre 1900 bei der amerikanischen Volkszählung eingesetzt, um Nachnamen für die leichtere Recherche zu codieren [206].

Die Beschreibung des Soundex-Algorithmus von Knuth [97] in seinem Buch 'The Art of Computer Programming, vol.3: Searching and Sorting' hat zu seiner „Wiederentdeckung“ geführt. Er wird heutzutage auch eingesetzt, um allgemein Suchen zu vereinfachen. Einige Datenbankmanagementsysteme wie beispielsweise MySQL aber auch die Skriptsprachen Perl und PHP stellen dafür den Soundex-Algorithmus zur Verfügung.

Der Soundex-Algorithmus basiert auf der Annahme, dass Worte, die ähnlich klingen, auch von der Semantik her ähnlich sind. Jedes Wort wird auf einen eindeutigen maximal vier Zeichen (1 Großbuchstabe + 3 Ziffern) langen Code nach folgendem Verfahren reduziert.

Der Anfangsbuchstabe des zu verarbeitenden Wortes bleibt in jedem Fall erhalten. Im Rest des Wortes werden alle Vokale, sämtliche h, w und y (Dehnungszeichen) aus dem Wort gestrichen und die Konsonanten nach folgender Tabelle in Ziffern übersetzt, wobei nur die ersten drei Ziffern in den Code übernommen werden:

- b, f, p, v = 1
- c, z, s, g, j, k, q, x = 2
- d, t = 3
- l = 4
- n, m = 5
- r = 6

Stehen nach der Umwandlung zwei oder mehr gleiche Ziffern hintereinander, wird nur eine davon beibehalten. Abschließend wird der Code auf 3 Ziffern gekürzt oder bei weniger als 3 Ziffern mit Nullen auffüllt.

Über den invertierten Index der Soundex-Codes ist dann die Suche nach Worten, die ähnlich zu einem Suchwort sind, realisierbar. Beispielsweise würden bei einer Suche nach Einträgen mit Name=Meier auch Einträge mit Name=Meyer oder Name=Mayer zum Suchergebnis gehören, siehe Beispiel 5.1(a). Will man allerdings Worte nach ihrem Klang mit Hilfe des Russel-Soundex sortieren, versagt dieser schnell, wenn sich schon deren erster Buchstabe unterscheidet, vgl. Beispiel 5.1(b). Durch das Zusammenstreichen ganzer Worte auf lediglich 4 Stellen, kommen außerdem auf einen einzigen Soundex-Code Dutzende passende Worte, die nicht unbedingt einen ähnlichen Klang haben. Ein weiteres Problem ist, dass die Ersetzungstabelle aufgrund ihrer Herkunft auf den englischsprachigen Raum ausgerichtet ist und so bei deutschen (oder anderssprachigen) Wörtern oft merkwürdige Ergebnisse liefert. Beispielsweise steht B625 sowohl für Bruckner als auch für

Bergmann oder H562 für Heinrich und Hammerschlag (siehe Beispiel 5.1(c)), obwohl ihr Klang völlig verschieden ist [206].

Beispiel 5.1 Umwandlung nach dem Soundex-Algorithmus

	Umzuwandeln	Vokale streichen	Konsonanten ersetzen	Doppelte streichen	Auffüllen/ streichen
(a)	Meier	Mr	M6	M6	M600
	Meyer	Mr	M6	M6	M600
	Mayer	Mr	M6	M6	M600
(b)	Fischer	Fscr	F226	F26	F260
	Wischer	Wscr	W226	W26	W260
(c)	Bruckner	Brcknr	B62256	B6256	B625
	Bergmann	Brgmnn	B62555	B625	B625
	Heinrich	Hnrc	H562	H562	H562
	Hammerschlag	Hmmrsclg	H5562242	H56242	H562

Aus diesen Gründen wurde das Verfahren modifiziert. Wer diese neue Variante, die unter den Begriffen *refined soundex* oder auch *extended soundex* bekannt ist, entwickelt hat, ist leider nicht klar. Gelegentlich wird aber auch R.C. Russel in diesem Zusammenhang genannt [205].

Das eigentliche Vorgehen ist dem des Original sehr ähnlich. Statt drei Stellen werden nun vier verwendet und es werden neun verschiedene Laute statt früher sechs unterschieden:

- b,p = 1
- c,k,s = 3
- m,n = 8
- d,g,t = 6
- l = 7
- r = 9
- f,v = 2
- g,j = 4
- q,x,z = 5

Sieht man sich die Wörter aus dem vorhergehenden Beispiel an, die beim ursprünglichen System Probleme bereiteten, erkennt man, dass sie sich nun unterscheiden: Bruckner B9389, Bergmann B9688, Heinrich H8930 und Hammerschlag H8937.

Es gibt einige Weiterentwicklungen des Soundex-Codes für die phonetische Textumwandlung in anderssprachigen Datenbeständen, z.B. der Algorithmus von Daitch und Mokotoff [133] für osteuropäische Namen. Auch bei diesem Algorithmus wird ein Name auf eine bestimmte Anzahl von Zahlen abgebildet, allerdings nicht buchstabenweise, sondern abhängig von der Position der Buchstaben und in welcher Kombination sie auftreten.

Die Berücksichtigung des Kontextes der Buchstaben ist auch die grundlegende Idee des Metaphone-Algorithmus von Philips [151, 152]. Er arbeitet

ähnlich dem Soundex-Verfahren, d.h. Vokale werden nach dem ersten Zeichen ignoriert und verschiedene Laute zusammengefasst. Im Gegensatz zum Soundex-Verfahren werden Buchstaben jedoch nicht isoliert kodiert, sondern in verschiedenen Kontexten betrachtet, z.B. wann 'c' als 's' oder 'k' ausgesprochen wird[1].

Das Verfahren Buchstaben durch Ziffern zu ersetzten, kann man auch auf Wortteile bzw. ganze Wörter ausdehnen und so wie in einer Lautschrift suchen. Solche Verfahren, z.B. [132], basieren auf langen Ersetzungstabellen, die angeben, wie bestimmte Buchstabenkombinationen unter Berücksichtigung des Kontextes umgesetzt werden [204].

Die Umwandlung nach solchen Verfahren ist sehr kompliziert und aufwendig. Daher haben wir uns für die Verwendung des Soundex-Codes in BibConsist entschieden. Dies bedeutet keine Einschränkung, da durch den Einsatz geeigneter Entwurfs- und Implementierungstechniken (Strategiemuster, vgl. Abschnitt 6.1.1) der Algorithmus zur Laufzeit austauschbar gehalten werden kann. Damit lässt sich auch eine Sprachunabhängigkeit erreichen, indem der Benutzer zur Laufzeit wählen kann, ob der deutschsprachige oder englischspachige Soundex verwendet werden soll.

Um die phonetische Darstellung beliebig langer Zeichenfolgen, die auch Ziffern enthalten können, miteinander vergleichen zu können, wird in BibConsist der modifizierter Soundex-Code weiter abgewandelt. Er unterscheidet sich im wesentlichen in zwei Punkten vom oben beschriebenen modifizierten Soundex-Code:

- Der Code wird nicht auf 4 Zeichen gekürzt,
- Damit Ziffern bei der Umwandlung erhalten bleiben, werden die Konsonanten durch Kleinbuchstaben anstatt durch Ziffern ersetzt.

Um die Ähnlichkeit von Zeichenketten zu bestimmen, werden in BibConsist zwei verschiedene Methoden verwendet:

- *Listenvergleich:* Es wird - unabhängig von der Position der einzelnen Worte in der Zeichenkette - gezählt, wieviele Worte gleich bzw. phonetisch ähnlich sind und wieviele Worte verschieden sind.
- *gemeinsame Zeichenfolgen:* Dieses Maß basiert auf der Anzahl aufeinanderfolgender Zeichen, die in beiden Zeichenketten übereinstimmen: Für jedes gemeinsame Zeichen in Folge wächst die Ähnlichkeit um das Quadrat der Anzahl gemeinsamer aufeinanderfolgender Zeichen. Ein

[1] Der Metaphone-Algorithmus ist ebenfalls für die englische Sprache entworfen worden.

> gemeinsames Zeichen erhöht die Ähnlichkeit um 1, ein direkt folgendes gemeinsames zweites Zeichen um 4, ein drittes um 8, ein viertes um 16, usw. Die Ähnlichkeit wird anschließend so auf einen Bereich von 0 bis 100 normiert, dass die Länge der Zeichenketten ausgeglichen wird: Je kürzer die zu vergleichenden Zeichenketten sind, um so weniger gemeinsame aufeinanderfolgende Zeichen sind notwendig, um einen hohen Wert für die Ähnlichkeit zu erzielen [120].

Die erste Methode wird für Felder verwendet, in denen Worte vertauscht auftreten können, ohne dass dies in Konflikt zur Ähnlichkeit steht. Beispielsweise sind zwei *author*-Felder, in denen die gleichen Autoren in unterschiedlicher Reihenfolge angegeben sind, ähnlich. Felder, in denen Wortvertauschungen kritisch sind, d.h. veränderte Reihenfolge der Worte führt zu unterschiedlichen, semantisch nicht ähnlichen, Zeichenketten, werden mit der zweiten Mehtode verglichen. Diese Methode wird in BibConsist beispielsweise auf die *journal*-Felder angewandt.

BibConsist wurde ursprünglich als eigenständiges Programm konzipiert, um die Arbeit mit GeomBib zu erleichtern. Inzwischen ist BibConsist in BibRelEx integriert. Neben dem Vergleich mit Hilfe des Soundex-Codes werden weitere Inkonsistenzprüfungen vorgenommen. Beispielsweise ob Datensatzschlüssel mehrfach vergeben sind, ob alle Schlüssel in den Feldern *precedes*, *succeeds* und *cites* definiert sind, d.h. es gibt Einträge mit diesen Datensatzschlüsseln, und ob kein Schlüssel in den Feldern precedes, succeeds und cites auf den Eintrag selbst verweist. Darüber hinaus meldet BibConsist, wenn bei Büchern der Buchtitel sowohl im Feld *booktitle* als auch im Feld *title* definiert ist.

Die folgenden Beispiele für verschiedene Arten von Inkonsistenzen, die mit BibConsist gefunden werden können, sind ein Auszug aus den Ergebnissen der Konsistenzprüfung von GeomBib in der Version von März 1997. In dieser Version haben wir dabei insgesamt 69 Paare von inkonsistenten ähnlichen Einträgen (ohne Berücksichtigung von technischen Berichten, Diplomarbeiten, etc.) und 49 Fehler bei den Datensatzschlüsseln gefunden. Beispiel 5.2 zeigt einen doppelten Eintrag aufgrund unterschiedlicher Angabe des Titel und daraus resultierenden verschiedenen Datensatzschlüsseln. In Beispiel 5.3 führt die unterschiedliche Reihenfolge der Autorennamen zu verschiedenen Datensatzschlüsseln. Ebenso führen fehlende oder unterschiedliche Angaben des Erscheinungsjahres zu verschiedenen Datensatzschlüsseln, vgl. Beispiel 5.4. In Beispiel 5.5 kam es gleich durch mehrere Fehler zu einem doppelten Eintrag; es fehlte ein Autorenname, der Satztyp war verschieden und in dem zweiten Eintrag fehlte die Angabe des Buchtitels.

Beispiel 5.2 Inkonsistenz aufgrund unterschiedlicher Angabe des Titels

```
@book{s-asds-90
, author =      "H. Samet"
, title =       "Applications of Spatial Data Structures"
, year =        1990
... }

@book{s-asdsc-90
, author =      "H. Samet"
, title =       "Applications of Spatial Data Structures: Computer
                 Graphics, Image Processing, and {GIS}"
, year =        1990
... }
```

Beispiel 5.3 Inkonsistenz durch unterschiedliche Reihenfolge der Autoren

```
@incollection{fs-amgfe-72
, author =      "J. Fukuda and J. Suhara"
, title =       "Automatic Mesh Generation for Finite Element Analysis"
, year =        1972
... }

@incollection{sf-amgfe-72
, author =      "J. Suhara and J. Fukuda"
, title =       "Automatic Mesh Generation for Finite Element Analysis"
, year =        1972
... }
```

Beispiel 5.4 Inkonsistenz aufgrund falscher bzw. fehlende Angabe des Jahres

```
@article{ngv-begs-
, author =      "M. H. Nodine and M. T. Goodrich and J. S. Vitter"
, title =       "Blocking for External Graph Searching"
, note =        "To appear"
... }

@article{ngv-begs-96
, author =      "M. H. Nodine and M. T. Goodrich and J. S. Vitter"
, title =       "Blocking for External Graph Searching"
, year =        1996
... }
```

Beispiel 5.5 Inkonsistenz durch mehrere Fehler

```
@inproceedings{dl-cvdrp-91
, author =      "H. Djidjev and A. Lingas"
, title =       "On computing the {Voronoi} diagram for restricted planar
                 figures"
, booktitle =   "Proc. 2nd Workshop Algorithms Data Struct."
, year =        1991
, pages =       "54--64"
... }

@incollection{d-cvdrp-91
, author =      "H. Djidjev"
, title =       "On computing the {Voronoi} diagram of restricted planar
                 figures"
, booktitle =   "??"
, year =        1991
, pages =       "54--64"
... }
```

Die Konsistenzüberprüfung unterstützt zum einem den Benutzer bei der Bearbeitung der Datenbasis und hilft so die Eingabe von doppelten oder fehlerhaften Datensätze von vornherein zu vermeiden. Im speziellen Anwendungsfall von GeomBib kann zum anderen der Administrator von GeomBib beim Einmischen aller Akualisierungsvorschläge Duplikate aufspüren und so die Konsistenz der neuen GeomBib-Version sicher stellen. Welche weiteren Mittel zur Unterstützung der GeomBib Aktualisierung in BibRelEx verwendet werden, beschreiben wir im folgenden Abschnitt.

5.2 Unterstützung der GeomBib Aktualisierung

Aufgrund der periodischen Aktualisierung von GeomBib, siehe Abschnitt 3.3, lassen sich die Einträge in der Datenbasis logisch drei Gruppen zuordnen:

global Originaleinträge aus GeomBib.

update Einträge, die dem GeomBib-Verwalter zugeleitet werden sollen, weil sie Korrekturen an bereits in GeomBib vorhandenen Einträgen enthalten oder bisher in GeomBib unbekannt sind.

local Einträge, die nur für den einzelnen Nutzer von Interesse sind und entweder gar nicht oder erfolglos der GeomBib-Verwaltung zur Aufnahme vorgeschlagen wurden.

In unserer Arbeitsgruppe hat es sich bewährt, für jede dieser drei Gruppen eine Datei zu verwalten, vgl. [140]. Diese Aufteilung wird in BibRelEx beibehalten. Für den Benutzer selbst soll die Dreiteilung der Datenbasis möglichst transparent sein. Dafür werden folgende Regeln festgelegt:

- *Überdeckung*: Existiert für einen bibliographischen Eintrag ein Datensatz in der lokalen Datenbasis, so überdeckt dieser eventuell existierende Einträge in der update/globalen Datenbasis. Analog überdeckt ein Eintrag in der update Datenbasis den entsprechenden Eintrag in der globalen Datenbasis.

- *Editieren*: Die globale Datenbasis kann nicht verändert werden. Falls ein Eintrag aus der globalen Datenbasis geändert wird, wird automatisch ein entsprechender Eintrag in der update Datenbasis erzeugt und die Änderungen in ihm abgelegt.

- *Löschen*: Falls ein globaler Eintrag gelöscht werden soll, wird wie beim Editieren ein entsprechender update Eintrag erzeugt und zusätzlich für die zentrale Datenverwaltung ein Löschhinweis im *note*-Feld des Eintrags erzeugt.

- *Neueinträge*: Der Benutzer kann für Neueinträge festlegen, ob diese in der lokalen oder update Datenbasis abgelegt werden sollen. Eine entsprechende Voreinstellung ist über die Konfigurationsdatei bzw. im Optionen-Menü möglich, um größtmögliche Transparenz für den Benutzer bzgl. der Dreiteilung zu erlangen.

Annotationen und Links werden in BibRelEx im selben Format wie bibliographische Einträge verwaltet, siehe Abschnitt 6.1.2. Damit kann die Aktualisierung des aggregierten Wissens mit der Aktualisierung von GeomBib in einheitlicher Form erfolgen. Durch die Übertragung der Dreiteilung der Datenbasis auf die Annotationen und Links lässt sich außerdem die Trennung von privater und öffentlicher Information mit der periodischen Aktualisierung des öffentlichen Bestandes vereinbaren.

Um den Benutzer beim Aktualisieren von GeomBib weitgehend zu unterstützen, ist folgende Automatisierung der Aktualisierung von GeomBib in BibRelEx integriert.

- *Erstellen der Aktualisierungsvorschläge*: Aus der globalen und update Version der Datenbasis wird die modifizierte Version der Datenbasis erzeugt und mit dem Unix-Befehl diff -c die Datei für die zentrale Verwaltung erzeugt[2].

[2]Systemabhängigkeit zur Zeit durch die zentrale Administration so vorgegeben

- *Einspielen einer neuen Version*: Für jeden Eintrag aus dem update-Teil der Datenbasis wird geprüft, ob die Aktualisierung in GeomBib aufgenommen wurde. Ist das der Fall, wird der Eintrag aus dem update-Bestand entfernt. Erfolgte eine Übernahme, die nicht vollständig identisch mit einem unterbreiteten Vorschlag ist, oder unterblieb die Übernahme ganz, so kann der Benutzer entscheiden, ob der vorgeschlagene Datensatz im update-Teil verbleiben soll (der Vorschlag wird also bei der nächsten Aktualisierung wieder eingereicht), aus dem update-Teil gelöscht werden soll oder ob er in die lokale Datenbasis übertragen werden soll. Weiterhin wird geprüft, ob Einträge aus dem lokalen Teil der Datenbasis aufgrund der Änderungsvorschläge anderer Nutzer nun in GeomBib sind. Diese Einträge werden aus dem lokalen Teil der Datenbasis gelöscht und bei Differenzen in den update Teil der Datenbasis verschoben.

Die Zuordnung der einzelnen Versionen eines Datensatzes erfolgt über den Schlüssel des Datensatzes. Daher muss der Fall der Schlüsseländerung gesondert betrachtet werden. In diesem Fall wird der „alte“ Eintrag im globalen Teil der Datenbasis gelöscht, d.h. ein entsprechender Eintrag mit *note*-Feld mit Löschhinweis in der update Version eingetragen. Zusätzlich wird der Datensatz mit dem neuen Schlüssel in den update Teil eingetragen.

Die Aktualisierung des Datenbestandes mit Hilfe der logischen Dreiteilung der Datenbasis erlaubt auch schnellere Updatezyklen bis hin zur Online-Aktualisierung. Hierfür kann man auf dem Web-Server die drei Dateien zu der jeweiligen Datenbasis halten und dem Benutzer erlauben, über eine CGI-Schnittstelle, Änderungen in die sich nun auf dem Web-Server befindende update-Datei einzubringen. Durch die logische Überdeckung sehen Benutzer bei Abfragen direkt die aktuellen Daten. Der Datenbank-Administrator kann dennoch jederzeit die Änderungen rückgängig machen, wenn sie z.B. gegen bestimmte Richtlinien verstoßen, da sie sich in einer separaten Datei, nämlich der update-Datei, befinden. Theoretisch könnte man sich über das Netz leicht als jemand anderes ausgeben und so negative Kommentare in GeomBib einbringen. Bleibt zu beurteilen, ob eine Kontrolle durch einen Administrator wirklich notwendig sein wird oder eine Selbstregulierung durch die Benutzer und/oder weitergehende Authentifizierungsmechanismen ausreichen. Auf die Richtlinien, die beim gemeinschaftlichen Arbeiten u.a. einer gewissen Qualitätssicherung dienen, gehen wir im nächsten Abschnitt ein.

5.3 Policies

Eine Policy ist eine Art Richtliniensammlung. Für das elektronische Publizieren werden Policies bereits in vielfältiger Weise verwendet, beispielsweise zur inhaltlichen Qualitätssicherung bei Zeitschriften und bei Tagungsbänden über das sogenannte Peer-Review, s. Abschnitt 4.3, zur Sicherstellung des Urheberrechts und zur Autorisierung.

Im gewissem Sinne stellen schon die README- und Authority-Dateien zu GeomBib Policies dar. In ihnen wird festgelegt, wie die Einträge zu formatieren sind, wie Zitierschlüssel gebildet werden, welche Schlüsselworte verwendet werden, wie Abkürzungen für Zeitschriften oder Konferenzen anzugeben sind und einige weitere Regeln. In BibRelEx bieten diese Dateien die Grundlage für die Eingabeunterstützung des Benutzers, indem beispielsweise der Zitierschlüssel automatisch nach diesen Regeln gebildet wird und die in der Authority-Datei angegebenen Schlüsselworte, Zeitschriften, Konferenzen und Herausgeber in Auswahllisten dem Benutzer zur Verfügung gestellt werden. So lässt sich eine einheitliche Eingabe erreichen und viele Eingabefehler von vornherein vermeiden. Für andere Anwendungsgebiete können diese Angaben mit Hilfe von Konfigurationsdateien überschrieben werden.

Jeder Benutzer kann Annotationen in den Datenbestand einbringen. Annotationen können selbst auch wieder annotiert werden, so dass eine einfache Diskussion entstehen kann. Um den Datenbestand vor Wildwuchs durch das Anbringen von Annotationen und neuer Dokumenteinträge zu schützen und eine inhaltliche Qualität der Beiträge sicherzustellen, müssen für das Arbeiten mit BibRelEx weitere geeignete Policies festgelegt werden.

Wir schlagen vor, dass bei Arbeiten, die einen Peer-Review Process durchlaufen, die Autoren selbst, zusammen mit ihrer Arbeit, einen Vorschlag für einen annotierten Eintrag in BibRelEx einreichen. Für die Gutachter der Arbeit ist es nach der Lektüre recht einfach festzustellen, ob der vorgeschlagene Eintrag korrekt und vollständig ist. Wird die Arbeit akzeptiert, so wird der Eintrag in BibRelEx übernommen. Dieser Ansatz, der auch von Cameron [30] vorgeschlagen worden ist, ist auch im Interesse der Autoren. Zusätzlich oder alternativ kann man auch Diskussionen am Rande der periodisch stattfindenden Tagungen/Workshops, bei denen zahlreiche Experten zusammen kommen, nutzen, um neue Arbeiten und Ideen in den Datenbestand einzubringen.

Darüber hinaus scheint uns eine Qualitätskontrolle auf dem Niveau eines Begutachtungsprozesses nicht machbar und nicht notwendig zu sein. So wie der Bestand nur durch den Beitrag vieler Nutzer aktuell gehalten werden kann, könnte auch die Kontrolle der Beiträge nur von vielen Experten erfolgen. Das würde einen unangemessenen Aufwand bedeuten und die Aktuali-

sierung des Datenbestandes unnötig verzögern. Zumal man davon ausgehen kann, dass jeder Wissenschaftler selbst in der Lage ist, fachlich relevante Informationen von banalen oder unsinnigen Beiträgen zu unterscheiden und so die kompetenten Benutzer selber für eine Qualitätssicherung des Datenbestandes sorgen.

Durch das Einbringen von öffentlichen Benutzerannotationen kann es im Prinzip auch zu Konflikten kommen, z.B. wenn ein Autor A eine öffentliche Annotation hinzufügt, die auf einen Fehler in einer Arbeit von Autor B hinweist. Hier schlagen wir folgende Verhaltensregel vor: Autor A ist verpflichtet sich zunächst von Autor B eine Zustimmung zu holen. Wenn dieser zustimmt, wird die Annotation in BibRelEx eingebracht und vermerkt, dass Sie von Autor A stammt. Grundsätzlich haftet ein Autor mit seinem Namen dafür, dass die von ihm eingebrachte Annotation in dem Sinne rechtmäßig ist. Wir glauben, dass nur in wenigen Ausnahmefällen eine Vermittlung durch Dritte, notwendig sein wird.

In GeomBib ist für jeden Eintrag ein *update*-Feld vorgesehen, in dem der Name desjenigen steht, der den Eintrag veranlasst hat. Dieses Vorgehen lässt sich leicht auf andere Datenbasen übertragen, um die verbürgte Autorenschaft für alle Beiträge namentlich auszuweisen. Dieses einfache Verfahren sollte später durch eine sichere Authentifizierung ersetzt werden.

Die Authentifizierung ermöglicht auch die Vergabe verschiedener Zugriffsrechte. Diese kann insbesondere für eine WWW-Schnittstelle zu BibRelEx sinnvoll sein. So kann man beliebigen Benutzern die Recherche erlauben, aber die Eingabe von Änderungen oder Neueinträgen auf registrierte Benutzer einschränken.

Zusätzlich könnte man die Möglichkeit der elektronischen Abstimmung anbieten. Ein solcher Ansatz findet sich im System ActivePerspective [145] zur Aushandlung gemeinsamer Schlüsselworte und ihrer Zuordnung zu Literaturquellen. Abstimmungsalternativen sind Akzeptieren, Ablehnen, Enthalten, Gegenvorschlag und die Kommunikation ohne Systemunterstützung. In BibRelEx könnte man so beispielsweise über den Inhalt von Annotationen abstimmen.

Eine Alternative zur elektronischen Abstimmung wäre der Diskurs über moderierte elektronische Foren, wie er beipspielsweise bei ENFORUM, einem virtuellen kollaborativen Fachwörterbuch [102], eingesetzt wird.

5.4 Aggregation von Wissen

5.4.1 Wissensrepräsentation

Eine der ersten Entscheidungen in BibRelEx war die Frage in welcher Form das Wissen in die Datenbasis eingebracht werden soll. In wissensbasierten Systemen finden sich die unterschiedlichsten Ansätze zur Wissensrepräsentation, angefangen von völlig unstrukturierten Informationen (Freitext), über semi-strukturierten Informationen, bei denen bestimmte Anteile der Information in einem fest vorgeschriebenen Format vorliegen müssen, bis hin zur formalen Repräsentation, die eine weitgehende automatische Interpretation ermöglicht. Bei der Festlegung einer geeigneten Repräsentation ist zu berücksichtigen, dass im Allgemeinen Formalisierung und Benutzerakzeptanz konkurrierende Ziele sind, da der Benutzer bei starker Strukturierung in seinen Kommunikationsmöglichkeiten eingeschränkt wird.

Freitext ermöglicht dem Benutzer Wissen in einer für ihn gut verständlichen Form einzubringen und erfordert keinen Einarbeitungsaufwand in irgendeine Beschreibungssprache. Dem gegenüber stehen als wesentliche Nachteile, dass keine Eindeutigkeit der getroffenen Aussagen vorliegt, Missverständnisse leicht möglich sind und (zumindest bis heute) kaum eine automatische Interpretation durch Rechner möglich ist.

Formale Ansätze wie die Verwendung von Dokumentationssprachen erreichen durch die Abbildung verschiedener Textformulierungen auf eindeutige Bezeichnungen eine höhere Eindeutigkeit und vermeiden bei der Recherche naturgemäß das Problem der Wahl geeigneten Vokabulars zur Anfrageformulierung. Der Benutzer braucht ein gesuchtes Konzept nur auf die entsprechende Bezeichnung in der Dokumentationssprache abbilden. Dafür muss er aber Kenntnis der zugrundeliegenden formalen Beschreibungssprache besitzen, was zu einem hohen Einarbeitungsaufwand führt. Für nur gelegentliche Benutzung ist ein solches System sicher nicht geeignet.

Da wir Wert auf eine einfache Benutzung von BibRelEx legen, aber auch leicht das Beziehungsgeflecht aus den Daten extrahieren können müssen, um eine effiziente Visualisierung zu ermöglichen, haben wir uns für einen semistrukturierten Ansatz entschieden. In BibRelEx erfolgt die Eingabe des Wissens durch Annotationen und Beziehungen (Links). Der Inhalt der Annotationen kann als Freitext eingegeben werden, die Eingabe von Beziehungen zwischen Arbeiten und das Verknüpfen von Annotationen mit den zugehörigen Dokumenten bzw. Beziehungen erfolgt in strukturierter Form, indem die zugehörige Information in vordefinierte Felder abzulegen ist und die Typen der Beziehungen (Linktypen) fest vorgegeben sind. Damit liegt die Struktur des Beziehungsgeflecht in maschinell interpretierbarer Form vor,

und der Benutzer hat genügend Freiheitsgrade bei der Formulierung seines Wissens. Die festgelegten Beziehungstypen kann er bei Bedarf durch Freitext-Annotationen ergänzen.

5.4.2 Trennung der Dokumentbeschreibungen von den Annotationen und inhaltsbasierten Beziehungen

Da in GeomBib bereits für jedes Dokument optional die Felder *cites, precedes, succeeds* und *annote* vorgesehen sind, könnte man mit Hilfe dieser Felder inhaltliche Beziehungen und Annotationen verwalten. Dieses Konzept ist aber zu unflexibel. Beispielsweise stehen damit nur die Zitierrelation und nicht beliebige Beziehungen zur Verfügung. Die Verwendung des *annote*-Feldes ermöglicht außerdem keine Trennung von Annotationen verschiedener Benutzer, da pro BibTEX-Eintrag jedes Feld nur einmal angegeben werden kann.

Daher werden in BibRelEx die inhaltlichen Beziehungen (Links) und Annotationen getrennt von den dazugehörigen Dokumenten verwaltet. Die Informationen über Start und Ziel aller Links werden dabei nicht in den Dokumentbeschreibungen integriert, sondern in spezialisierten Strukturen verwaltet. Die separate Linkverwaltung ist gerade bei großen Datenbeständen sinnvoll. So muss beispielsweise nicht die gesamte Dokumentenmenge analysiert werden, um das Beziehungsgeflecht zu extrahieren. Bei Änderungen, wenn z.B. ein Zieldokument verschoben (Zitierschlüssel ändert sich) oder gelöscht wird, kann der Link automatisch angepasst bzw. ebenfalls entfernt werden. Außerdem ist auch das Rückwärtsverfolgen der Links effizient möglich, denn es sind ja beide „Enden“ bekannt. Dies wird z.B. für beziehungsbasierte Anfragen und die graphische Aufbereitung benötigt. Zusätzlich kann damit beim Löschen von Einträgen geprüft werden, ob andere Einträge noch auf diesen verweisen. Werden Einträge gelöscht, obwohl noch Links auf diese verweisen, so sollten diese Links in der Visualisierung konsequenterweise nicht dargestellt werden.[3]

Um größtmögliche Flexibilität zu erhalten, wird ebenfalls die Information auf welches Dokument bzw. welche Beziehung eine Annotation verweist, in einem eigenständigen Link und nicht in der Annotation abgelegt.

[3]Da eine Dokumentenmenge in Bezug auf die Zitierrelation oder anderer inhaltlicher Beziehungen nicht abgeschlossen sein muss, kann es durchaus sinnvoll sein Dokumente anzuzeigen, die nicht in der Datenbasis vorhanden sind. So kann es beispielsweise vorkommen, dass eine Arbeit häufig zitiert wird, aber nicht in der verwendeten Fachdatenbasis vorkommt, da sie zu einem anderen Fachgebiet gehört. Eine solche wesentliche, da häufig zitierte Arbeit sollte in der Visualisierung sichtbar sein, damit Benutzer sie nicht übersieht. In BibRelEx ist die Anzeige solcher Dokumente optional möglich.

Die Trennung des Beziehungsgeflechts und des zusätzlichen Wissens über die Dokumente vereinfacht auch die Trennung von privaten und öffentlichen Datenbestand, da die entsprechenden Annotationen und Beziehungen einfach in den entsprechenden Teil der Datenbasis geschrieben werden können.

Weiterhin wird das Zusammenführen von Expertenwissen verschiedener Benutzer beim periodischen Update vereinfacht. Anstatt die entsprechenden Dokumentbeschreibungen ändern zu müssen, können die neuen Annotationen und Beziehungen einfach an den öffentlichen Teil der Datenbasis angefügt werden.

Zusammenfassend lässt sich festhalten, dass die getrennte Verwaltung von Dokumenten, Beziehungen und Annotationen einfach, effizient und nichtredundant ist, eine Trennung des öffentlichen und privaten Datenbestands ermöglicht, und das periodische Update vereinfacht.

5.4.3 Linktypen

In BibRelEx können beliebige inhaltliche Beziehungen durch typisierte und annotierte Links definiert und genutzt werden. Wie in Abschnitt 2.3 diskutiert, ist die geeignete Festlegung von Linktypen ein bis heute noch nicht zufrieden stellend gelöstes Problem. Einerseits ermöglicht die Verwendung beliebiger Linkbezeichnungen dem Benutzer eine möglichst genaue Charakterisierung von Beziehungen; andererseits entsteht dadurch die Gefahr inkonsistenter Typnamen.

In BibRelEx haben wir das Problem dadurch gelöst, dass dem Benutzer eine feste Menge an Linktypen präsentiert wird. Diese kann aber über die Verwendung von Konfigurationsdateien speziellen Bedürfnissen angepasst werden. So können beispielsweise gruppenspezifische Anforderungen berücksichtigt werden. Darüber hinaus hat der Benutzer die Möglichkeit Links zu annotieren. Somit kann er immer dann, wenn ihm ein Linktyp für eine bestimmte Beziehung nicht aussagekräftig genug ist, die Information durch Anheften einer Annotation ergänzen.

Linktypen können in BibRelEx auch dazu verwendet werden, um linkbasierte Recherchen weiter einzuschränken. Die Recherchemöglichkeiten von BibRelEx werden im folgenden Abschnitt besprochen.

5.5 Suchmöglichkeiten

Mit den bisher verwendeten Suchwerkzeugen für BibTEX-Datenbasen bibindex, biblook [54] und Ebiblook [140] war eine Suche nach Feldinhalten und Satzschlüsseln möglich. BibRelEx bietet die bekannte Suchfunktionalität mit

derselben Anfragesyntax an, damit der Umstieg für den Benutzer ohne großen Aufwand möglich ist. Darüber hinaus wurde die Suchfunktionalität erheblich erweitert. BibRelEx unterstützt folgende Anfragearten:

- **Spezifische Anfragen** beziehen sich auf ein scharf eingegrenztes, oft spezialisiertes Thema. Bei ihnen ist eher mit wenigen Anfrageergebnissen zu rechnen.
- **Allgemeine Anfragen** sind das Gegenstück zu den spezifischen Anfragen, d.h. diese Anfragen beziehen sich auf ausgedehnte Themen. Hier muss mit einer Flut von Anfrageergebnissen gerechnet werden.
- **„Ähnliches Dokument“ Anfragen:** Bezogen auf ein Ausgangsdokument wird nach Dokumenten mit ähnlichem Inhalt gesucht.
- **Beziehungsbasierte Anfragen** ermöglichen das Recherchieren der Struktur des Wissensgeflechts.

Für die ersten beiden Anfragetypen können die klassischen textbasierten Retrievalmöglichkeiten genutzt werden. Um die Menge der relevanten Anfrageergebnisse bei Anfragen bzgl. breiterer Themengebiete zu reduzieren, bietet sich die Anwendung des HITS-Algorithmus von Kleinberg, siehe Abschnitt 2.4, Seite 21, auf die Zitierbeziehung an. Authorities entsprechen vielzitierten und damit vermutlich wichtigen Arbeiten in einem Gebiet, Hubs entsprechen Arbeiten mit vielen für ein Gebiet relevanten Zitaten und damit Übersichtsarbeiten bzw. Review-Artikeln für dieses Gebiet. Viele Probleme, die bei der Anwendung des HITS-Algorithmus im WWW auftreten, spielen bei Zitiergraphen keine Rolle. Es gibt dort beispielsweise keine Werbe- oder Navigationslinks.

In BibRelEx kann der Benutzer bei der Suche angeben, ob er nach wesentlichen Dokumenten oder Übersichtsarbeiten zu einem Themengebiet sucht. Betrachten wir hierzu als Beispiel das Gebiet der Voronoi-Diagramme. Das Voronoi-Diagramm ist eine grundlegende Datenstruktur der Algorithmischen Geometrie. Daher findet man viele Arbeiten über Voronoi-Diagramme in der Datenbasis GeomBib. Mit Hilfe der beiden Anfragen

(a) hitsauth - voron*

(b) hitshubs - voron*

an die Datenbasis GeomBib, erhält man in BibRelEx wesentlichen Arbeiten (a) und Übersichtsarbeiten (b) im Gebiet der Voronoi-Diagramme. Die zugehörigen Ergebnisfenster findet man in der Abbildung A.6 im Anhang A.

Bei den wesentlichen Arbeiten liefert die Anfrage unter anderem die Arbeit von Shamos und Hoey [173], die grundlegend für Voronoi-Diagramme ist. Als Übersichtsarbeit wird das Kapitel *Voronoi Diagrams* von Aurenhammer und Klein [6] im Handbuch *Handbook of Computational Geometry* gefunden, das in der Tat einen sehr guten Überblick über Voronoi Diagramme in der Algorithmischen Geometrie gibt.

Zusätzlich wird bei den Suchergebnissen ihr Rank als Authority bzw. als Hub geliefert. Dieser Rank ist für den Benutzer aussagekräftiger als der Impakt Faktor, siehe Abschnitt 3.1, da er im Gegensatz zum Impakt Faktor alle Zitate berücksichtigt und nicht nur Zitate ausgewählter Zeitschriften über einen eingeschränkten Zeitraum. Darüberhinaus ist der HITS Algorithmus auch für die Bewertung von Literatur geeignet, die keiner Begutachtung unterzogen wurde.

In BibRelEx kann der Benutzer sich die Basismenge, auf die die weitere Analyse im HITS-Algorithmus aufbaut, darstellen lassen und erhält so einen Überblick über das Beziehungsgeflecht des Themengebiets und seiner Umgebung. Im Gegensatz zum Originalalgorithmus schränken wir bei der Bildung der Basismenge nicht die Zahl der hinzuzufügenden Verweise ein. Im Gegensatz zum WWW stehen ja in BibRelEx alle Verweise unmittelbar zur Verfügung und müssen nicht aus einem verteilten Informationsraum zusammen gesucht werden. Aus demselben Grund ist es uns möglich, nicht nur direkte Verweise zu verwenden, sondern auch längere Pfade zu betrachten, indem das Verfahren zur Bildung der Basismenge wiederholt auf die sich jeweils ergebende Basismenge angewendet wird. Schon ein einfaches Beispiel zeigt, dass dies sinnvoll ist: Man betrachte wie in Abbildung 5.1 dargestellt drei Arbeiten A, B, C, wobei B und C von A zitiert werden, d.h. in Kozitationsbeziehung stehen.

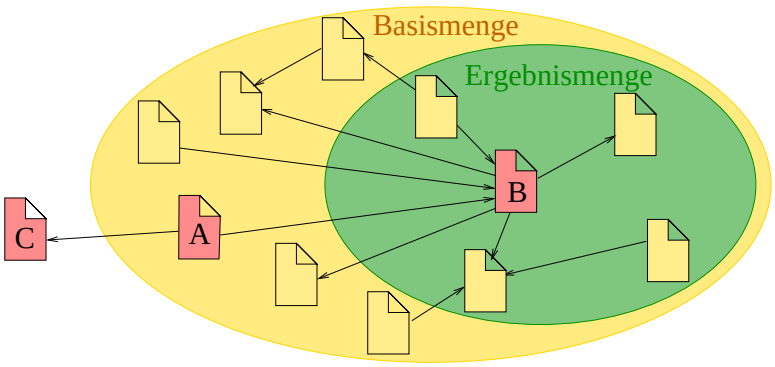


Abbildung 5.1: Einfluss der berücksichtigten Pfadlänge auf die Bildung der Basismenge beim HITS-Algrithmus

HITS baut die Basismenge ausgehend von der Ergebnismenge einer Anfrage zum gesuchten Thema auf. Aufgrund von unterschiedlichem Vokabular kann es leicht vorkommen, dass nur das Dokument B in der Ergebnissmenge dieser Anfrage liegt. Die Erweiterung zur Basismenge nach dem HITS-Algorithmus sieht nur die Aufnahme direkter Verweise vor. Daher würde zwar die Arbeit A in die Basismenge aufgenommen und bei der Linkanalyse berücksichtigt werden, aber nicht die Arbeit C. Da aber Kozitation auf thematische Ähnlichkeit hinweisen kann, wird so eine für die Anfrage eventuell relevante Arbeit nicht berücksichtigt. Gleiches gilt für Arbeiten, die bibliographisch miteinander gekoppelt sind. Wie man in Abbildung 5.2 sieht, kann es auch sinnvoll sein, Pfadlängen > 2 zu berücksichtigen. Die Dokumente C und D sind nicht bibliographisch gekoppelt, werden nicht gemeinsam zitiert (Kozitation) und auch ein Vergleich mit Hilfe des CCIDF liefert keinen Zusammenhang zwischen den beiden Dokumenten. Betrachtet man aber einen größeren Ausschnitt aus dem Zitiergraphen, lässt sich vermuten, dass zwischen den Dokumenten C und D der selbe Zusammenhang besteht wie zwischen den Dokumenten A und B. Diese Problematik wurde unabhängig von unserer Arbeit auch von Lu u. a. [118] erkannt. Sie haben den HITS-Algorithmus entsprechend erweitert, auf Zitiergraphen angewendet und mit empirischen Versuchen den Einfluss der berücksichtigten Pfadlänge untersucht. Kürzere Pfade lieferten eine höhere Präzision der Suchergebnisse. Bei der Verwendung kurzer Pfade werden direkt zusammenhängende Dokumente gefunden, während durch die Berücksichtigung längerer Pfade auch tiefere Verbindungen zwischen Dokumenten gefunden werden. Dafür ist der Rechenaufwand höher, da die Basismenge durch die mehrfache Erweiterung umfangreicher ist. In BibRelEx kann die zu berücksichtigende Pfadlänge vom Benutzer optional eingestellt werden. Man könnte sogar noch weiter gehen, indem man nicht nur die einzelnen Pfade berücksichtigt, sondern bestimmte *Muster* im Graphen sucht und zur Wissenserkennung nutzt.

Ähnliche Dokumente können ebenfalls durch Auswertung des Beziehungsgeflechts bestimmt werden. Dazu wird beispielsweise die CCIDF Gewichtung, die Übertragung der TFIDF Gewichtung auf die Zitierrelation berechnet. Da sich die TFIDF Gewichtung analog zur CCIDF Gewichtung auch auf beliebige inhaltliche Beziehungen übertragen lässt, kann die Bestimmung der Ähnlichkeit von Dokumenten flexibel nach benutzerdefinierten Kriterien erfolgen. Für die Berechnung der Ähnlichkeit basierend auf beliebigen inhaltsbasierten Beziehungen wird die vereinfachte Vorkommenshäufigkeit

$$nxf_{mi} = \begin{cases} 1 & \text{wenn zwischen Dokument } d_i \text{ und Dokument } d_m \\ & \text{eine Beziehung vom Typ } x \text{ vorliegt} \\ 0 & \text{sonst} \end{cases}$$

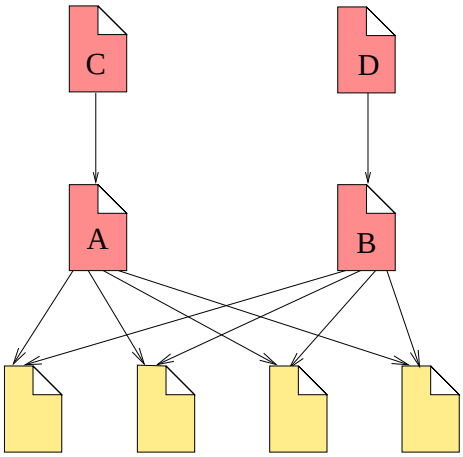


Abbildung 5.2: Indirekter Zusammenhang über bibliographische Koppelung [118]

verwendet (Bezeichnungen siehe Seite 35).

Zusätzlich ermöglicht BibRelEx beziehungsbasierte Anfragen, d.h. der Benutzer kann die Beziehungen zwischen den Dokumenten zur Recherche nutzen. Dies ist zum einen über das Verfolgen der Beziehungslinks möglich, zum anderen wurde die Abfragesprache um Recherchemöglichkeiten bzgl. Annotationen und Beziehungen erweitert. Neben den Suchmöglichkeiten nach Feldern in Annotationen und Beziehungen, kann man nach Dokumenten suchen, zu denen es Annotationen und Beziehungen mit bestimmten Eigenschaften gibt, z.B. finde alle Arbeiten, die von Autor *A* annotiert wurden. Konkrete Beispiele hierzu findet man im Anhang B.

Ein wesentlicher Nachteil der bisher verwendeten Werkzeuge war, dass sie nicht das Editieren von Datensätzen unterstützt haben. Bisher musste der Benutzer nach Änderungen mit Hilfe eines Editors oder dem X11 Frontend für BibTEX Datenbanken bibview [116] den Index der Datenbasis mit bibindex [54] neu erstellen, bevor er eine neue Suche mit biblook [54] starten konnte. BibRelEx unterstützt das Editieren von Einträgen und hält dabei den Index aktuell, so dass der lästige Aufwand für das Neuindizieren entfällt.

Das Ergebnis einer Suche ist wie die ursprüngliche Datenbasis eine Menge von bibliographischen Einträgen, Annotationen und Links. In BibRelEx werden daher Ergebnismengen wie Datenbasen behandelt. Das hat für den Benutzer den Vorteil, dass er auf eine Ergebnismenge dieselben Operationen anwenden kann wie auf die Datenbasis selbst. Sie wirken dann nur innerhalb der Ergebnismenge. So kann der Benutzer die Datenmenge schrittweise eingrenzen und auch eine Vorauswahl der zu visualisierenden Daten treffen, was gerade für große Datenbasen sehr hilfreich ist. Auf dieses Problem gehen wir

im Abschnitt 5.6.1 genauer ein.

Neben den text- und strukturbasierten Recherchemöglichkeiten unterstützt eine Visualisierung der Wissensstruktur die inhaltlichen Navigation im Dokumentenraum. Bei Verwendung geeigneter Layoutverfahren lässt sich die Struktur des Raumes leicht erkennen. Dabei bieten sich für verschiedene Fragenstellungen unterschiedliche Darstellungsmöglichkeiten an. Übersichtsarbeiten und zentrale Arbeiten lassen sich beispielsweise gut in Darstellungen, die mit kräftebasierten Verfahren erstellt werden, erkennen. Für die Geschichte der Lösung eines Problems bietet sich dagegen eine hierarchische Darstellung über die Zeit an. Die verschiedenen Darstellungsmöglichkeiten von BibRelEx werden in den folgenden Abschnitten vorgestellt.

5.6 Visualisierung

Eines der größten Probleme der Visualisierung großer Dokumentsammlungen ist die Handhabung der Informationsmenge. Die Visualisierung von semantischen Netzwerken versagt schnell, wenn eine bestimmte Anzahl von dargestellten Knoten und Links überschritten wird. Für die „Lesbarkeit“ der graphischen Darstellung ist es deshalb wesentlich, dass nicht zu viele Details dargestellt werden. Auf der anderen Seite darf durch die Reduktion des Detaillierungsgrades nicht zu viel von der strukturellen Information verloren gehen, damit weiterhin die wesentlichen Zusammenhänge gut zu erkennen sind und eine zielgerichtete Erforschung des Informationsraumes möglich ist.

Zur Reduktion der Komplexität der Darstellung und zur zusätzlichen Unterstützung des Benutzers werden in BibRelEx eine Reihe von Methoden eingesetzt:

- Vorauswahl der darzustellenden Dokumente mit Hilfe von Anfragen,
- geeignete Platzierungstechniken,
- dynamisches Layout bei möglichst weitgehender Wahrung der Mental Map
- Clusterung,
- Verwenden von Filtern,
- Link-Aggregation,

- Stufenloses Zoomen und einfache Navigation.

5.6.1 Vorauswahl der Darstellungsmenge

BibRelEx stellt zwei Möglichkeiten zur Verfügung, die darzustellende Dokumentenmenge vorauszuwählen. Als erstes kann der Benutzer Top Down vorgehen, indem er mit einem Übersichtsgraphen startet. Dieser kann theoretisch den gesamten Wissensraum umfassen. Sinnvoll ist jedoch gerade bei großen Datenmengen zunächst mit einer Anfrage die Menge der Dokumente, die überhaupt berücksichtigt werden sollen, festzulegen. Startet der Benutzer für die Ergebnismenge einer Anfrage die Visualisierung, werden alle Dokumente der Ergebnismenge dargestellt, wobei ggf. vorgeclustert wird, vgl. Abschnitt 5.6.4. Alle weiteren Operationen in der Visualisierung beschränken sich auf die Ergebnismenge. Die Menge kann nicht durch weitere Abfragen vergrößert werden, sondern nur weiter eingeschränkt werden.

Alternativ kann der Benutzer Bottom Up vorgehen. Hierzu startet er die Visualisierung für eine Datenbasis ohne vorherige Anfrage. Die Darstellungsfläche in der Visualisierung bleibt hierbei zunächst leer. Die Darstellung kann der Benutzer nun schrittweise mit Hilfe von Anfragen an die Datenbasis aufbauen. Zusätzlich hat er die Möglichkeit die Darstellung um Verweise von und/oder zu den dargestellten Dokumenten zu erweitern. Für die Erweiterungen stehen alle Dokumente der Datenbasis zur Verfügung.

Die erste Methode hat den Vorteil, dass man von vornherein die Datenmenge einschränken kann. Bei der zweiten Methode ist die Darstellungsmenge dagegen jederzeit erweiterbar. Diese Erweiterbarkeit ist wesentlich bei der Informationssuche basierend auf der strukturellen Information. Wie bereits bei der Analyse des HITS-Algorithmus (Abschnitt 2.4, Seite 21) beschrieben, kann es Dokumente geben, die hochrelevant für eine Anfrage sind, aber keines der Stichworte der Anfrage enthalten. Dieses Problem tritt noch gravierender auf wenn die zugrundeliegende Datenbasis keine Volltexte enthält sondern nur Metainformationen über Dokumente, wie es (zur Zeit) bei BibRelEx der Fall ist, da die Wahrscheinlichkeit, dass die Suchbegriffe in der Metainformation vorkommen geringer ist.

Daher lässt sich zusammenfassend feststellen, dass die erste Methode eher geeignet ist, sich einen Überblick über Teile des Wissensraumes zu verschaffen, z.B. über alle Arbeiten eines Autors. Zur gezielten Informationssuche hingegen sollte die zweite Methode verwendet werden.

5.6.2 Platzierung

Neben der Vorauswahl der Darstellungsmenge ist die Anordnung der Knoten und Kanten ein wesentlicher Faktor, der die Lesbarkeit der Darstellung beeinflusst. Für BibRelEx wurden verschiedene 2D- und 3D Darstellungstechniken implementiert und erprobt:

- Kräftegesteuerte Verfahren (Spring Embedder, BibRelEx Embedder, Graph Embedder von Frick u. a. [61]),
- minimaler Spannbaum basierend auf Co-Occurence-Analysen[4],
- Zeitdarstellung.

Zunächst stellen wir die einzelnen Verfahren kurz vor, um zu zeigen, was jeweils mit welcher Darstellung erreicht werden kann. Im Anschluß daran werden die einzelnen Darstellungsarten dann detailliert beschreiben.

Die kräftegesteuerten Verfahren sind in BibRelEx sowohl für zwei- als auch dreidimensionale Darstellungen realisiert worden. Bei den 2D Darstellungen bietet LEDA die Möglichkeit diese schrittweise zu animieren und auch einen Teil der Knoten festzuhalten. Dies ermöglicht eine dynamische Darstellung unter Wahrung der Mental Map und gibt so dem Benutzer optimale Möglichkeiten, gezielt den Einfluss einzelner Arbeiten zu verfolgen. Auf die Dynamik wird genauer im Abschnitt 5.6.3 eingegangen. Bei den 3D Darstellungen wird zunächst der gesamte Graph angeordnet und dann animiert. Dabei wird der Graph in Rotation gezeigt, wobei der Benutzer mit Hilfe der Maus Einfluss auf Drehgeschwindigkeit und -richtung nehmen kann. Über die Maus ist auch ein Ein- und Auszoomen der drehenden Darstellung möglich. Die 2D und 3D Darstellungen ergänzen sich somit: Mit Hilfe der 3D Darstellung gewinnt der Benutzer einen Eindruck der Gesamtstruktur, indem er das Geflecht in seiner Gesamtheit von allen Seiten betrachten kann. Die 2D Darstellung erlaubt ihm sich Einzelheiten genauer anzusehen. Beispielsweise kann er sich die zu einem Knoten gehörenden Dokumentdaten ansehen und alle Knoten können mit den Zitierschlüsseln der zugehörigen Dokumente versehen werden. Zur weiteren Unterstützung stehen dem Benutzer bei den 2D Darstellungen verschiedene Filter- und Selektionsmethoden zur Verfügung und mit Hilfe von Kontextmenüs kann er sich z.B. zu einem Knoten die referierenden Knoten hervorheben lassen.

[4]Bei der Co-Occurence-Analyse wird die Ähnlichkeit zweier Dokumente über gemeinsames Auftreten bestimmter Eigenschaften, z.B. gemeinsame Terme, gemeinsames zitiert werden oder gemeinsame Zitate bestimmt.

Die minimale Spannbaumdarstellung hilft Zusammenhänge in der Dokumentenmenge zu erkennen. In ihr lassen sich beispielsweise leicht zentrale Arbeiten zu einzelnen Themen erkennen. Die Berechnung des minimalen Spannbaums ist in BibRelEx für verschiedene Co-Occurence Maße realisiert, z.B. basierend auf gemeinsames zitiert[5] werden, gemeinsames zitieren[5] oder gemeinsame Terme. Bei dieser Darstellung werden die Knoten zunächst nicht neu angeordnet, sondern nur die Kanten neu gesetzt. So ist die Veränderung der Darstellung leicht für den Benutzer nachzuvollziehen. Anschließend kann der Spannbaum mit einem der kräftegesteuerten Verfahren animiert werden. Auch eine Kombination des Zitiergeflechts mit dem Spannbaum ist möglich, wobei sogar verschiedene Beziehungen zugrunde gelegt werden können.

Die Zeitdarstellung hilft die Entwicklung eines Gebietes nachzuvollziehen. Hier werden die Dokumente entlang einer Zeitachse angeordnet. Die Darstellung kann auch stufenweise aufgebaut werden, indem die jeweils nächsten referierenden/referierten Dokumente zu der Darstellung hinzugenommen werden.

Zusammenfassend lässt sich festhalten, dass die 3D kräftebasierten Verfahren der Übersicht über die Gesamtstruktur dienen, mit den 2D Verfahren Details des Beziehungsgeflechts erforscht werden können, die Spannbaumdarstellungen helfen Zusammenhänge im Wissensraum zu erkennen und die Zeitdarstellung die Entwicklung eines Gebietes verdeutlichen. Nachdem nun die Ziele der einzelnen Darstellungsmöglichkeiten klar sind, kommen wir zu der ausführlichen Beschreibung der einzelnen Darstellungen.

Kräftegesteuerte Darstellungen

In BibRelEx wird die Spring Embedder Variante von Fruchterman und Reingold [62] genutzt, da sie von der verwendeten Bibliothek LEDA angeboten wird. Ein Vorteil des Spring Embedders in Hinblick auf die Qualitätskriterien für ein „schönes“ Layout ist, dass er die Knoten gleichmäßig verteilt. Für die Darstellung von Beziehungsgeflechten stellt dies allerdings eher einen Nachteil dar, da gemeinsam referierte Dokumente symmetrisch um die referierenden Dokumente angeordnet werden. Dies führt zu zusätzlichen Kantenüberkreuzungen. Die Darstellung mit Hilfe des Graph Embedders von Frick u. a. [61] ist zwar schneller als mit dem Spring Embedder, hat aber dieselben Nachteile durch das symmetrische Layout. Um die Vorteile der Konvergenzbeschleunigung zu nutzen, wurde für BibRelEx der Graph Embedder weiterentwickelt. Damit gemeinsam referierte Dokumente zwischen den referierenden Dokumenten angeordnet werden, haben wir zusätzlich zu

[5] auch für beliebige Beziehungen möglich

den anziehenden und abstoßenden Kräften im Graph Embedder noch weitere anziehende Kräfte verwendet, die die Kantenrichtung berücksichtigen. Haben zwei Knoten gemeinsam referierte Nachbarknoten, so werden die zusätzlichen Anziehungskräfte so definiert, dass die gemeinsamen Nachbarknoten durch den Mittelpunkt der beiden referierenden Knoten angezogen werden, siehe Abbildung 5.3. Die Stärke w_a der zusätzlichen Anziehungskräfte kann eingestellt werden und darüber die „Spitzwinkeligkeit“ der Darstellung beeinflusst werden. Für $w_a = 0$ entspricht der BibRelEx-Embedder dem 2D Graph Embedder [61] bzw. GEM3D [28].

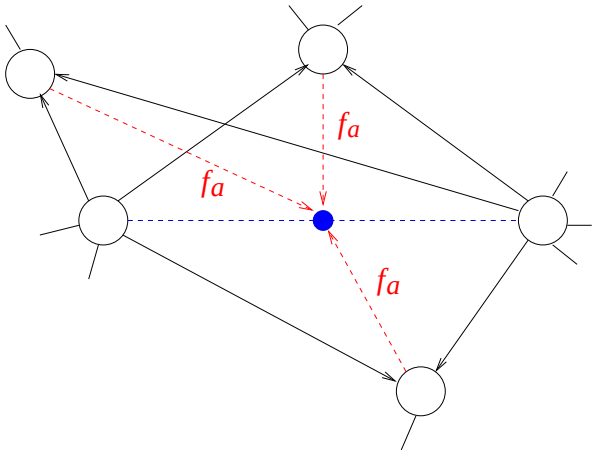
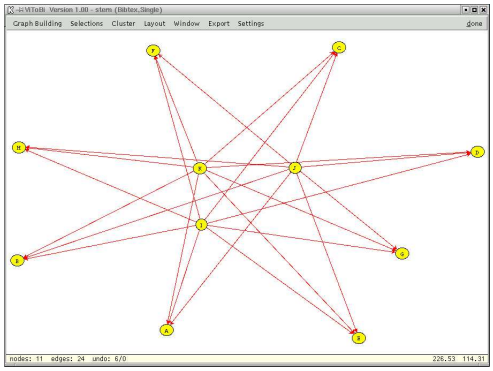


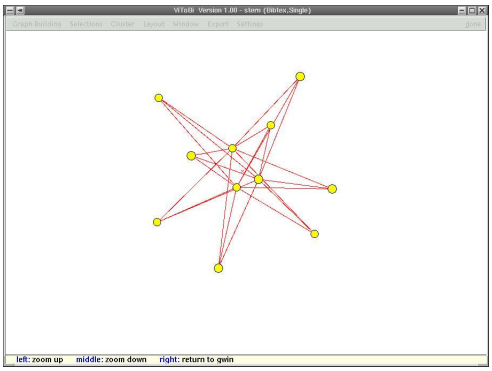
Abbildung 5.3: Zusätzliche Anziehungskräfte (f_a) zur Anordnung gemeinsam referierter Knoten

Abbildung 5.4 zeigt die Darstellung von drei Dokumenten mit gleichen Zitaten mit dem Graph Embedder und dem BibRelEx Embedder. Bei der Darstellung mit dem BibRelEx Embedder ist der Zusammenhang zwischen zitierenden und zitierten Dokumenten auf einen Blick gut zu erkennen, da die gemeinsam zitierten Dokumente zentral zwischen den zitierenden Dokumenten nah zueinander angeordnet werden. Diese Anordnung spiegelt die Koziationscluster wieder.

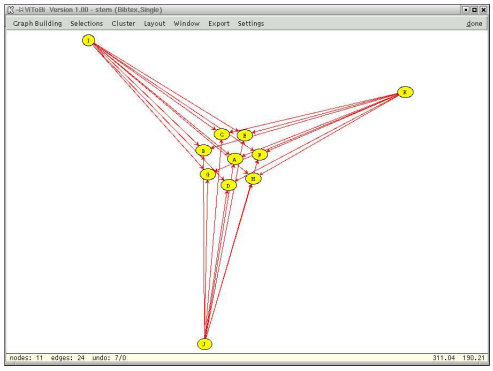
Für kleinere stärker vernetzte Geflechte liefert der BibRelEx Embedder ebenfalls besser lesbare Ergebnisse als der Spring Embedder, wie man bei der Gegenüberstellung der Darstellungen zum Nutzungsbeispiel 4.1 *Organisation eines Seminars* in Abbildung 5.5 sieht. In diesem Beispiel wurde die Literatur zu einem Seminar gesichtet und die Arbeiten nach verschiedenen Kriterien geordnet. Für jedes Kriterium wurde eine Annotation angelegt und an die entsprechenden Arbeiten angebracht. In der Abbildung 4.2 waren die Arbeiten zu einem Thema dargestellt. Für Abbildung 5.5 wurde das Beispiel auf 3 Seminarthemen ausgeweitet. Zu jeder Arbeit ist im *keyword*-Feld



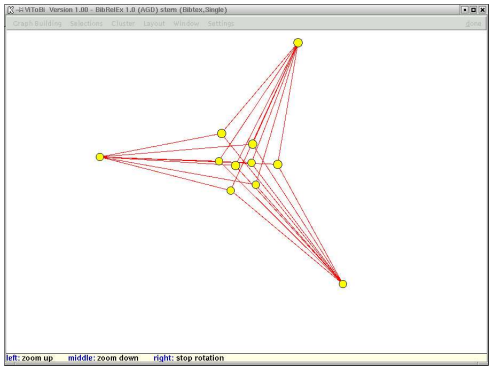
(a) 2D Graph Embedder



(b) 3D Graph Embedder



(c) 2D BibRelEx Embedder

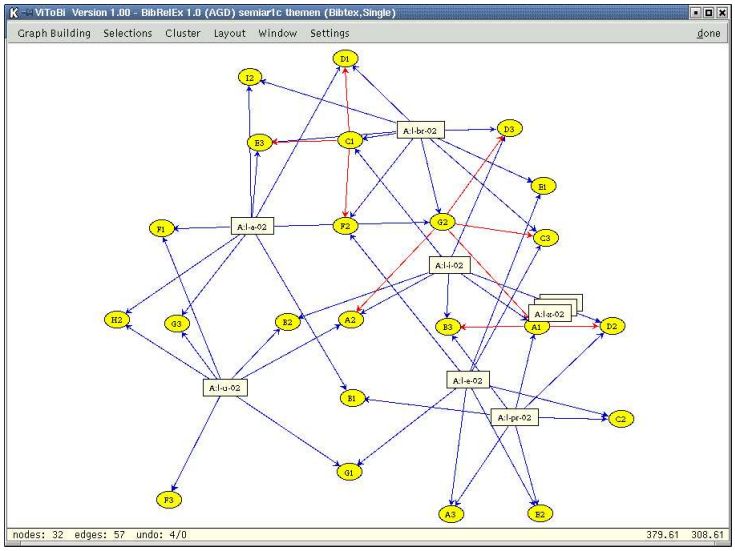


(d) 3D BibRelEx Embedder

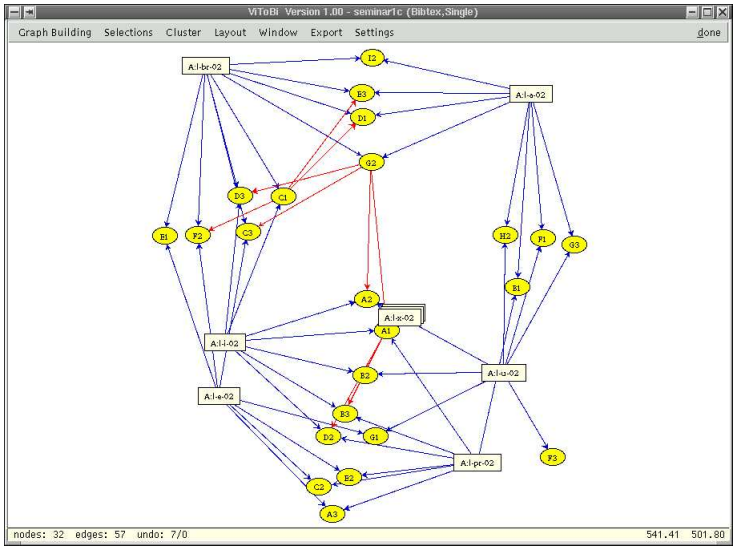
Abbildung 5.4: kräftegesteuerte Layoutmethoden in BibRelEx

vermerkt, auf welche Themen sie sich bezieht. Zusätzlich wurde die Zitierbeziehung zwischen den Arbeiten zum Seminar teilweise eingegeben und einige zusätzliche Annotationen gemacht.

In den Abbildungen 5.6(a), 5.6(b)und 5.6(c) ist ein umfangreicheres geclustertes Zitiernetz mit den verschiedenen 2D Layout-Verfahren dargestellt (Zitiergeflecht zu BibRelEx, vgl. Abschnitt 4.4). Auch hier ist der Zusammenhang zwischen zitierenden und zitierten Dokumenten bei der Darstellung mit dem BibRelEx Embedder am leichtesten zu erkennen. Die Abbildung 5.6(d), 5.6(e) und 5.6(f) zeigen dasselbe Geflecht nach Auflösung der Cluster. Hier ist deutlich zu sehen, dass der BibRelEx Embedder insbesondere bei höherem Detaillierungsgrad besser lesbare Darstellungen liefert.

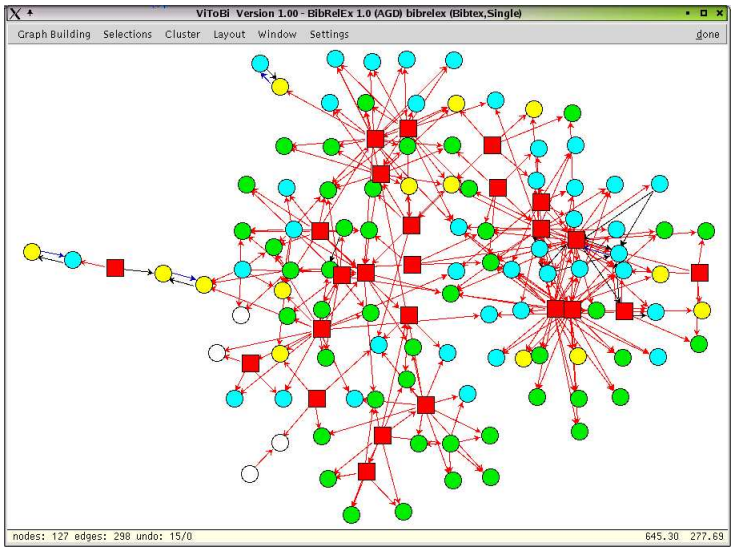


(a) 2D Spring Embedder

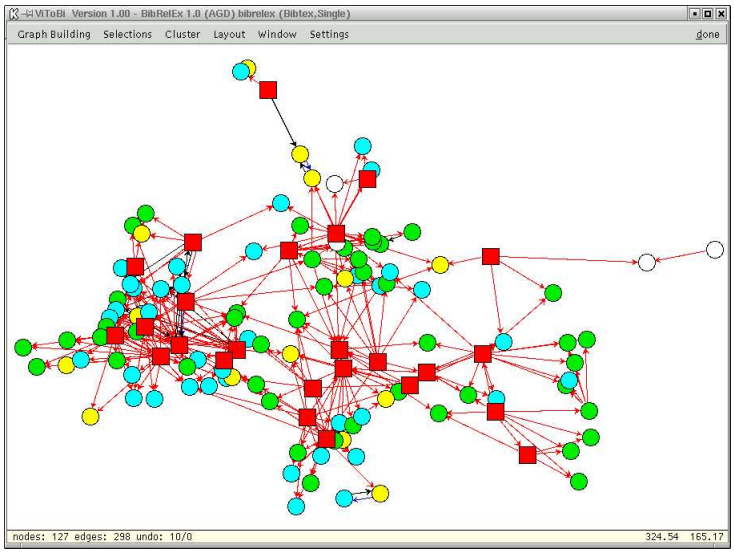


(b) 2D BibRelEx Embedder

Abbildung 5.5: Darstellungsmöglichkeiten in BibRelEx am Beispiel 4.1 *Organisation eines Seminars*

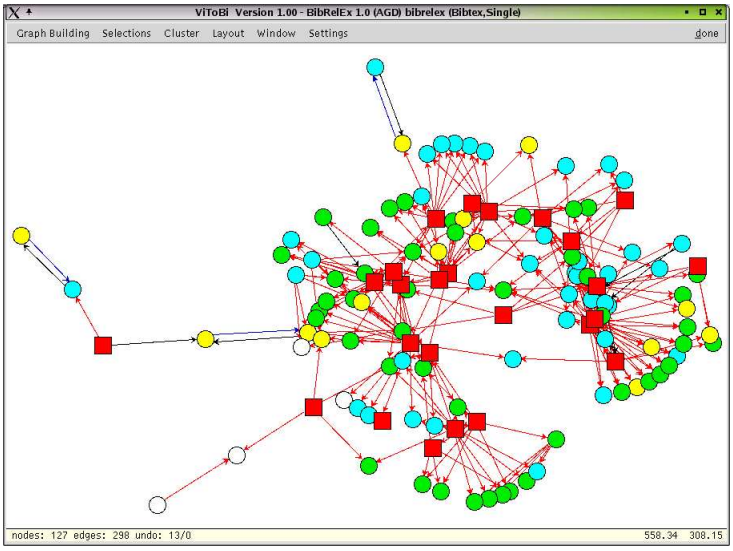


(a) 2D Spring Embedder

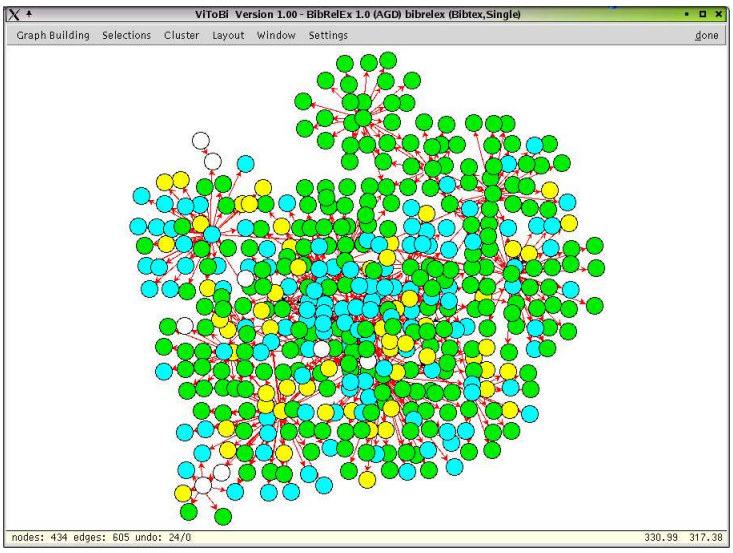


(b) 2D Graph Embedder

Abbildung 5.6: Darstellungsmöglichkeiten in BibRelEx

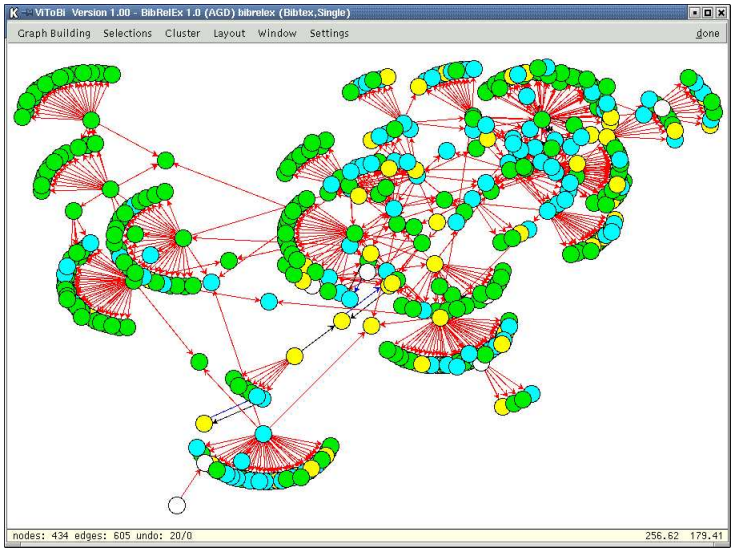


(c) 2D BibRelEx Embedder

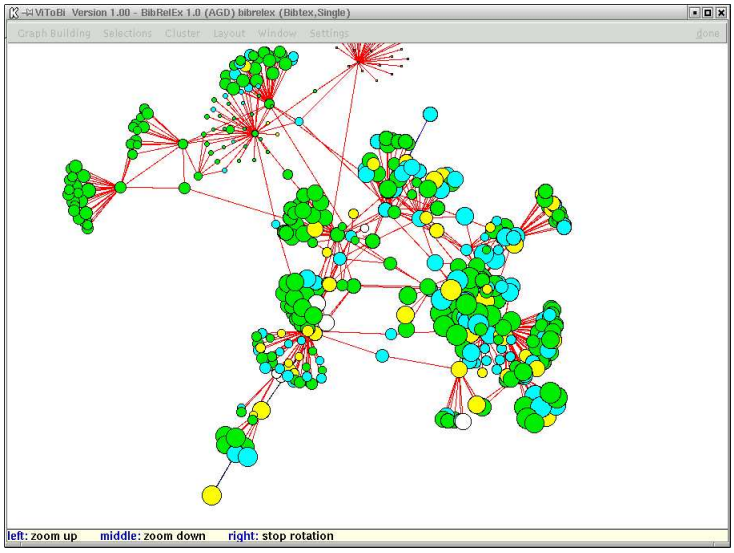


(d) 2D Spring Embedder

Abbildung 5.6: Darstellungsmöglichkeiten in BibRelEx



(e) 2D BibRelEx Embedder



(f) 3D BibRelEx Embedder

Abbildung 5.6: Darstellungsmöglichkeiten in BibRelEx

Darstellung minimaler Spannbäume

Die minimale Spannbaumdarstellung spiegelt die semantische Ähnlichkeit der Dokumente basierend auf dem gemeinsamen Vorkommen verschiedener Charakteristika wieder. Die Darstellung basiert auf einer Co-Occurence-Analyse: Tritt eine bestimmte Eigenschaft häufig zusammen bei zwei Dokumenten auf, werden diese als zusammengehörig in Bezug auf diese Eigenschaft angesehen und die Verbindung zwischen ihnen stärker gewichtet. Das heißt häufiges, gemeinsames Auftreten bestimmter Eigenschaften spiegelt semantische Nähe wieder. Beispielsweise kann man – wie bereits in Abschnitt 3.1 erläutert - annehmen, dass zwei Dokumente thematisch ähnlich sind, wenn sie häufig gemeinsam zitiert werden (Kozitation).

Mit Hilfe der Co-Occurence-Distanz wird zunächst ein gewichteter ungerichteter Graph erzeugt: Zwei Knoten werden genau dann in den Graphen eingefügt und mit einer Kante verbunden, wenn die zugehörige Co-Occurence-Ähnlichkeit einen Schwellwert übersteigt. Die Kante wird mit der zugehörigen Co-Occurence-Distanz gewichtet. Für den so entstandenen Graphen wird der minimale Spannbaum bestimmt. Durch die Bildung des minimalen Spannbaums werden in der Darstellung des Beziehungsnetzwerkes nur die wesentlichen Beziehungen wiedergegeben.

Das Verfahren kann auf verschiedene Eigenschaften angewandt werden. Neben der Zitierrelation sind für die Co-Occurence-Analyse gerade in BibRelEx beliebige andere Beziehungen nutzbar. Ebenso kann sie auf verschiedenen Objekten basieren und ist nicht nur auf die Analyse von Dokumenten beschränkt. Beispielsweise kann man das Netzwerk der sich zitierenden Autoren betrachten und auf diesem eine Kozitationsanalyse anwenden. Der entstehende Spannbaum spiegelt dann den Einfluss der Autoren in bestimmten Wissensgebieten wieder. Eine solche Darstellung wird bei dem System Starwalker von Chen [34, 35] benutzt.

Eine andere Möglichkeit ist, die Co-Occurence-Analyse auf die in den Dokumenten vorkommenden Begriffe anzuwenden, um ähnliche Begriffe zu bestimmen. Dieses Verfahren kann bei der Suche genutzt werden, um aus der Suchergebnismenge Begriffsgruppen abzuleiten, mit denen nachfolgende Suchanfragen durch das Hinzufügen oder Streichen von Suchbegriffen verbessert werden können. Dieses Vorgehen wird bei einigen Suchmaschinen genutzt.

In BibRelEx kann der minimale Spannbaum für die Co-Occurence-Ähnlichkeit basierend auf beliebigen Beziehungen zwischen Dokumenten erzeugt werden. Eine Ko-Autor-Darstellung wie bei Chen [34, 35] haben wir bisher nicht implementiert, da in BibRelEx der Schwerpunkt auf der Verwaltung und Recherche auf Dokumentenebene liegt. Eine entsprechende Erweiterung

wäre aber mit geringem Aufwand möglich.

Eine Darstellung basierend auf der Ko-Term-Analyse ist in BibRelEx möglich, allerdings nur für Graphen mit wenigen tausend Knoten. Eine bessere Skalierbarkeit wäre durch geeignetes Preprocessing und Ablegen der Terme mit den Dokumenthäufigkeiten in einer Datei/Datenbank möglich, siehe Kapitel 8. Da wir mit BibRelEx aber insbesondere den Nutzen inhaltlicher Beziehungen untersuchen wollen, wurde aus Zeitgründen auf eine besser skalierbare Implementierung zunächst verzichtet. Klassische Textretrieval-Funktionalität kann später als zusätzlicher „Service“ für den Benutzer immer noch optimiert bzw. ergänzt werden, spielt aber für die Fragestellung in dieser Arbeit eine untergeordnete Rolle.

Wir wollen nun die minimalen Spannbaumdarstellungen anhand der Literatur zu dieser Dissertation (Zitiergeflecht zu BibRelEx, vgl. Abschnitt 4.4) näher betrachten und bezüglich ihrer Nützlichkeit bei der Recherche in Wissensgeflechten untersuchen. Aufgrund der Größe der Datenbasis würden die resultierenden Darstellungen sehr viele Knoten enthalten und schnell unübersichtlich werden. Daher wurden in allen Darstellungen vor der Bestimmung des minimalen Spannbaums zunächst alle Dokumente ohne Verweise zu einem Cluster zusammengefasst und anschließend die Darstellung noch mit einem einfachen Verfahren (*simple clustering*) geclustert. Dieses Verfahren fasst ein Dokument mit denjenigen seiner Zitate zusammen, die sonst von keiner anderen Arbeit zitiert werden und die auch keine anderen Arbeiten zitieren. Es werden also diejenigen Knoten in Clustern „versteckt“, die keinen wesentlichen Beitrag zur Gesamtstruktur des Geflechts leisten. Cluster sind in den Abbildungen durch rote rechteckige Knoten dargestellt. In allen Spannbaumdarstellungen spiegelt die Kantendicke die Stärker der Co-Occurence-Ähnlichkeit zwischen zwei Dokumenten wieder. Je stärker die Ähnlichkeit ist, desto dicker werden die Kanten gezeichnet.

In Abbildung 5.7 ist der minimale Spannbaum basierend auf der Kozitation dargestellt.

Anhand der Abbildung kann man ein Problem dieser Darstellungsmethode erkennen: Zwischen den Dokumenten -c- (CompuScience) [59] und -ansii-96 (Ariadne) [63] (markierte Knoten in Abbildung 5.7) besteht kein wesentlicher Zusammenhang. Dennoch lässt die Darstellung gerade einen solchen vermuten. Ursache ist, dass die beiden Arbeiten nur einmal zitiert werden und das zusammen, womit sich eine Kozitationsähnlichkeit von 100% ergibt. Um eine Fehlinterpretation zu vermeiden, sollten nur stärker vernetzte Dokumente Einfluss auf die Spannbaumdarstellung nehmen dürfen. Daher nehmen wir nur solche Dokumente auf, die mit einer gewissen Häufigkeit zitiert werden. Um neuere Arbeiten nicht zu sehr zu benachteiligen, normieren wir diese Zitierrate mit der Anzahl der Arbeiten, die nach dem Erscheinen der jewei-

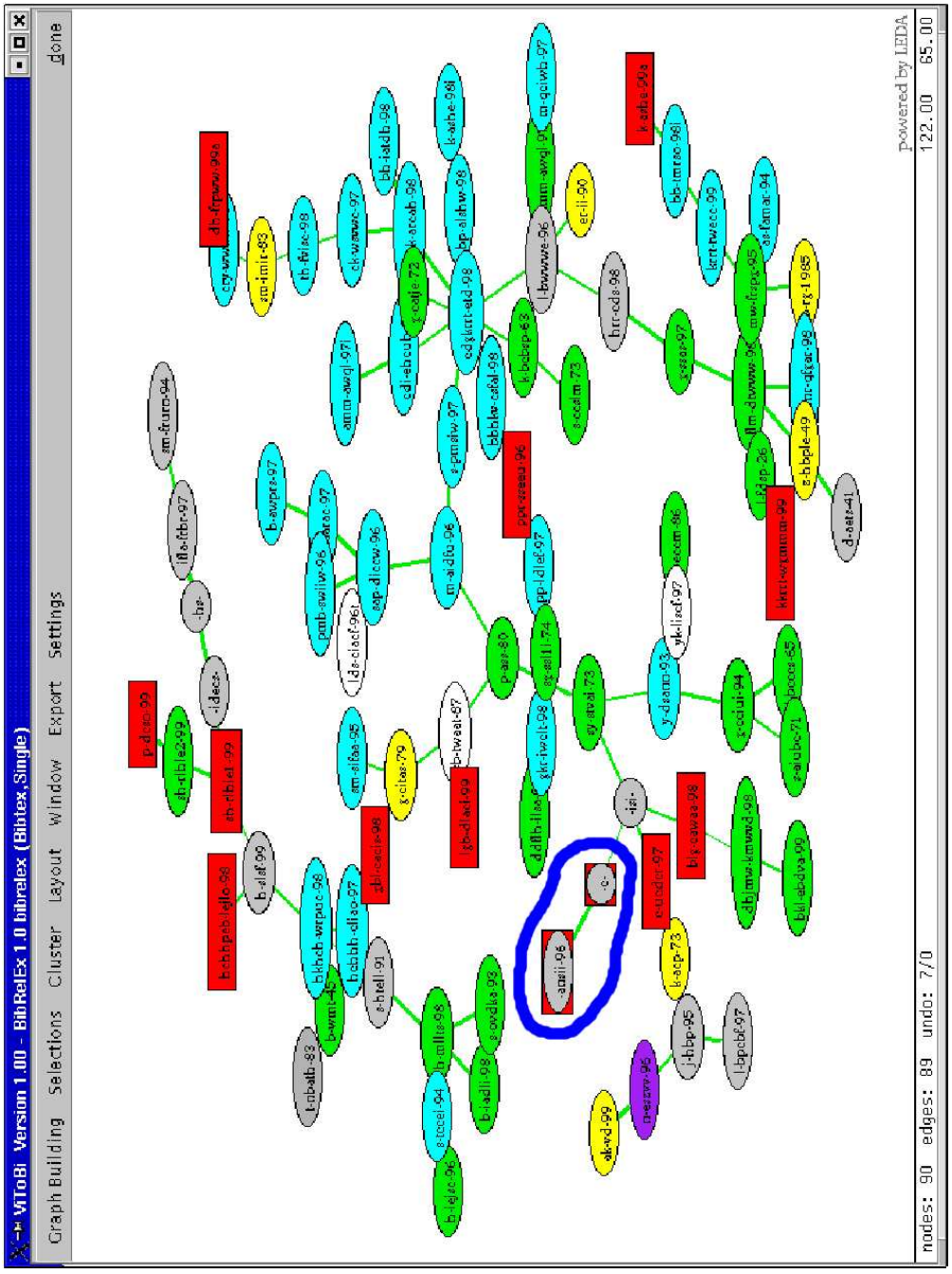


Abbildung 5.7: Minimaler Spannbaum ohne Normalisierung der Zitierrate

ligen Arbeit erschienen sind. Diese Vorgehensweise wenden auch Popescul u. a. [155] in ihrem heuristischen Clusterverfahren an, vgl. Abschnitt 5.6.4. Für Arbeiten, bei denen kein Erscheinungsjahr angegeben ist, wird weiterhin die nichtnormierte Zitierrate ausgewertet.

Abbildung 5.8 zeigt für das eben betrachtete Geflecht den minimalen Spannbaum basierend auf der Kozitation unter Berücksichtigung der normierten Zitierrate. Durch die Verwendung der normierten Zitierrate entfallen solche Knoten, die nur selten zitiert werden. Von diesen Arbeiten kann man vermuten, dass sie ohnehin keine wesentlichen Arbeiten sind[6]. Dieses Vorgehen reicht aber noch nicht aus, um erst kürzlich erschienene und damit noch kaum zitierte Arbeiten nicht zu vernachlässigen. In BibRelEx kann der Benutzer daher wählen, ob neuere Arbeiten auf jeden Fall mit in den Spannbaum übernommen werden sollen. Um wie in den beiden vorangegangenen Abbildungen auch Cluster mit in der Spannbaumbildung zu berücksichtigen, wird bei der Berechnung der normierten Zitierrate für diese das Jahr des Cluster-Zentroiden zugrunde gelegt.

Noel [144] hat vorgeschlagen, für die minimale Spannbaumdarstellung nicht nur Zwei-Tupel bei der Bestimmung der Ähnlichkeit anhand der Kozitation zu berücksichtigen, sondern auch Tupel höherer Kardinalität zur Ähnlichkeitsbestimmung heranzuziehen. So wird ein noch engerer Zusammenhang zwischen den Dokumenten verlangt, um in den Spannbaum aufgenommen zu werden und die Gefahr der Fehlinterpretation reduziert. Auf der anderen Seite werden so mehr Dokumente vernachlässigt. Damit werden neuere Arbeiten noch stärker (über einen grösseren Zeitraum!) benachteiligt, da eine höhere Zitierrate notwendig ist, um Berücksichtigung zu finden. Für die Recherche in Massendaten, wie den SCI [72, 73] mag das in Ordnung sein, nicht aber für Wissensgeflechte, die im Verhältnis dazu eher klein sind, oder wenn man bei der Recherche gerade an neueren oder sehr speziellen Arbeiten in einem Gebiet interessiert ist.

Ein weiteres Problem lässt sich bei den Dokumenten mit den Zitierschlüsseln k-ashe-98i [94] und k-ashe-99a [95] (selektierte Knoten in Abbildung 5.8) feststellen: Beide Dokumente sind im Prinzip zwei verschiedene Versionen der selben Arbeit, einmal in einem Proceedingsband und einmal als erweiterte Fassung in einer Zeitschrift erschienen. Im Allgemeinen ist das Zitierverhalten so, dass nur eine von beiden Arbeiten zitiert wird und damit die Kozitationsähnlichkeit der beiden Arbeiten *unerwartet* niedrig ist. Somit werden die beiden Arbeiten an ganz verschiedenen Stellen des Spannbaumes angeord-

[6]Dennoch kann eine selten zitierte Arbeit gerade für eine spezifische Anfrage relevant sein! Bei der Spannbaumdarstellung geht es aber eher darum eine Einsicht in die Entwicklung von Wissensgebieten zu gewinnen.

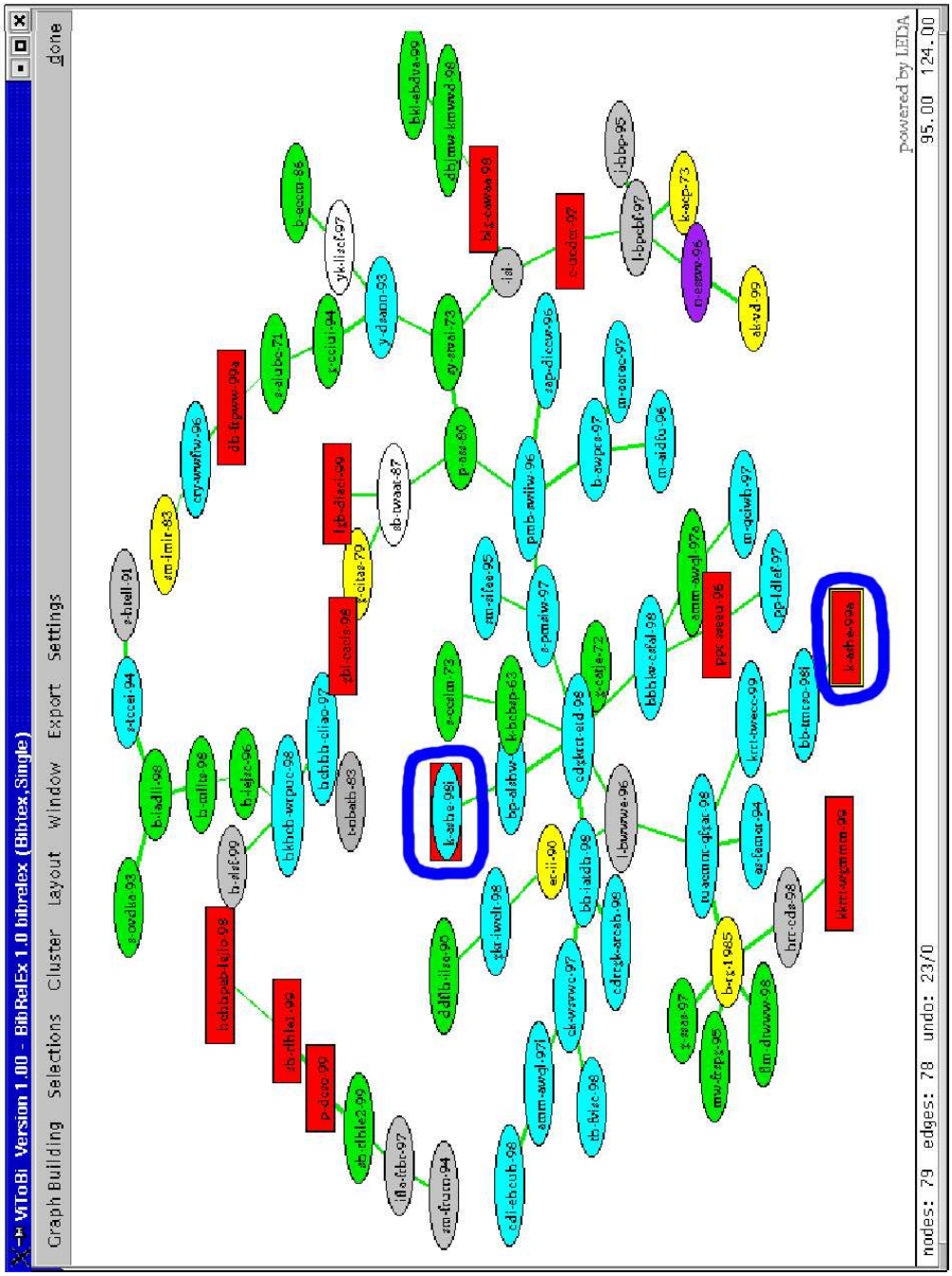


Abbildung 5.8: Minimaler Spannbaum mit Normalisierung der Zitierrate

net, obwohl der Zusammenhang zwischen beiden Arbeiten theoretisch sehr hoch ist. Außerdem ist gerade die Arbeit von Kleinberg zentral für BibRelEx, findet sich aber dennoch *nur* am Rand des Spannbaumes wieder, was daran liegt, dass es noch eine verhältnismäßig neue Arbeit ist. Dies zeigt, wie schnell eine *unzureichende* Verlinkung der Arbeiten zu einer Fehlinterpretation bei dieser Darstellung führen können.

Für eine gezielte Recherche ist die Spannbaumdarstellung ebenfalls nicht geeignet, da auch eine Arbeit, die wenig zur gesamten globalen Struktur des Zitiernetzes beiträgt, für einzelne Anfragen ausschlaggebend sein kann.

Dennoch kann diese Darstellung für den Benutzer sehr hilfreich sein, um sich einen Überblick über die Literatur zu einem Gebiet zu verschaffen. Man sollte sich aber stets der Gefahr der Möglichkeit der Fehlinterpretation bewusst sein. Zentrale Arbeiten, die gut vernetzt sind – wie in der Abbildung 5.9 das Dokument mit dem Zitierschlüssel cdgkrrt-etd-98 [32] – lassen sich leicht erkennen[7]. Sie befinden sich in der Nähe des Zentrums des Spannbaums. Sie sind sicher eine gute Wahl zur weiteren Lektüre, wenn man nach wesentlichen Arbeiten auf einem Gebiet sucht. BibRelEx ist in einem sehr aktuellen, aktiven Forschungsbereich angesiedelt, so dass sich das zugehörige Wissensgeflecht noch im Aufbau befindet und sich rasch ändert. Bei etablierten Themen dürfte die Gefahr der Fehlinterpretation geringer sein. Für Recherchen im Bereich der aktuellen Forschung ist die Spannbaumdarstellung basierend auf der Kozitation ohnehin nicht geeignet, da sich die neueren Arbeiten bestenfalls an dessen Rand wiederfinden. Diesen Nachteil hat die Darstellungsmethode aber mit allen Recherchemethoden gemeinsam, die auf der Kozitationsanalyse beruhen. Die Kozitationsanalyse trägt immer eine gewisse Verzögerung mit sich, da es eine gewisse Zeit dauert, bis Arbeiten zitiert werden.

Zusammenfassend lässt sich sagen, dass für die minimale Spannbaumdarstellung basierend auf der Kozitation eine ausreichend vernetzte Menge Daten vorliegen muss, da es sonst zu einer Beliebigkeit der Interpretierbarkeit der Darstellung kommt. Dennoch kann die Spannbaumdarstellung wertvolle Hinweise bei der Recherche geben, z.B. um zentrale Arbeiten zu finden.

BibRelEx ermöglicht auch eine Spannbaumdarstellung basierend auf der bibliographischen Koppelung. Wie in Abschnitt 3.1 beschrieben, verbindet die bibliographische Koppelung im Gegensatz zur Kozitation jüngere Arbeiten, so dass diese Darstellung eine Alternative ist, wenn man sich für neuere Arbeiten interessiert. Hierbei wird auch keine „Mindestzitierrate“ verlangt.

[7] Für eine noch besser Übersichtlichkeit der Darstellung kann man – wie bei Abbildung 5.9 – in BibRelEx *signifikante Knotenbeschriftung* verwenden, d.h. Knoten werden nur dann beschriftet, wenn der Knotengrad eine vorgegebene Schranke überschreitet.

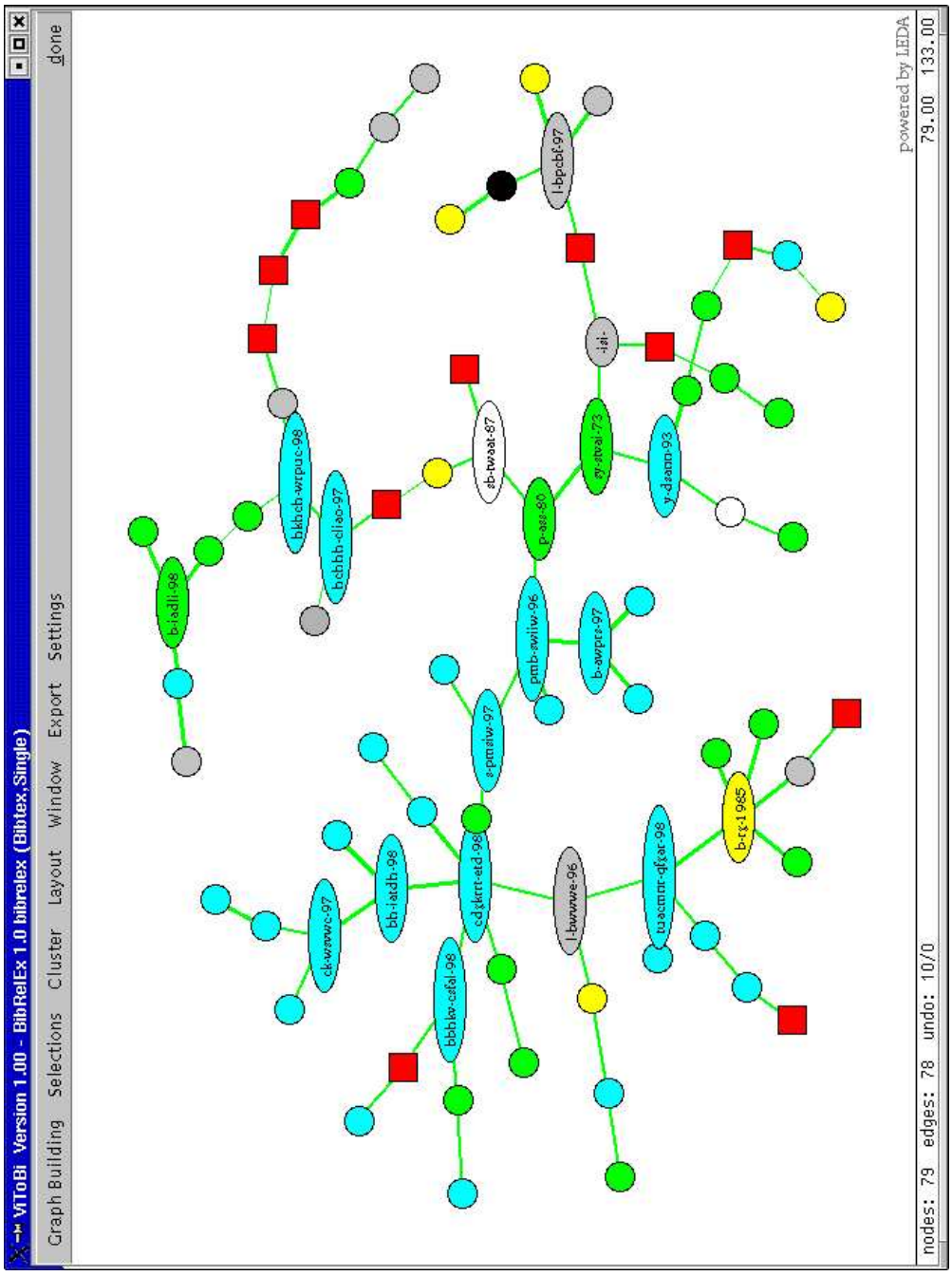


Abbildung 5.9: Minimaler Spannbaum mit Normalisierung der Zitierrate und signifikanter Knotenbeschriftung[7]

Somit werden bei dieser Darstellung neuere Arbeiten nicht benachteiligt. Allerdings ist zu berücksichtigen, dass die Darstellung insofern subjektiv ist, als dass sie auf den Angaben der Autoren der Arbeiten selbst beruht. Abbildung 5.10 zeigt den minimalen Spannbaum basierend auf der bibliographischen Koppelung des selben Zitiergeflechts wie in den Abbildungen 5.7 bis 5.9.

Eine Möglichkeit, die direkten Beziehungen wiederzugeben, ist die Darstellung des minimalen Spannbaums des Zitiernetzes bei direkter Gewichtung der Zitierkanten mittels Kozitation, bibliographischer Koppelung oder Wortgewichtung. Das heißt die Kanten des ursprünglichen Zitiergraphen werden – im Gegensatz zu den bisher betrachteten Spannbaumdarstellungen – beibehalten und entsprechend gewichtet. Die neue Spannbaumdarstellung ist allerdings ebenfalls subjektiv und verstärkt den Effekt von Zitiergemeinschaften und Selbstzitaten. Abhilfe kann man hier dadurch schaffen, dass man Selbstzitierungen und Mehrfachnennungen durch andere Autoren innerhalb eines Jahres als Indikator von Zitiergemeinschaften vor der Berechnung der Spannbaumdarstellung ausschliesst. Der Vorteil der neuen Darstellung ist, daß sie - im Gegensatz zu den bisher besprochenen Spannbaumdarstellungen - die direkten Beziehungen zeigt. In BibRelEx ist die Spannbaumdarstellung des Zitiernetzes bei Gewichtung der Zitierkanten mit Kozitation, bibliographische Koppelung und Wortgewichtung möglich. Die Abbildung 5.11 zeigt den minimalen Spannbaum des Zitiergeflechtes mit Kozitationsgewichtung.

Die Spannbaumdarstellung kann mit der Darstellung beliebiger Beziehungen kombiniert werden, z.B. Zitiergraph und Kotermbeziehung oder Zitiergraph und Kozitationsspannbaum, siehe Abbildung 5.12. Leider ist es bisher mit der für BibRelEx verwendeten Visualisierungbibliothek LEDA nicht möglich, gerichtete und ungerichtete Kanten in einer Darstellung gemeinsam zu verwenden. Daher werden bei der kombinierten Darstellung auch die Kanten des minimalen Spannbaums gerichtet dargestellt. Die Richtung hat für die Spannbaumkanten allerdings keine Bedeutung.

Eine andere Möglichkeit die Entwicklung eines Gebietes nachzuvollziehen bietet die im Folgenden beschriebene Zeitdarstellung.

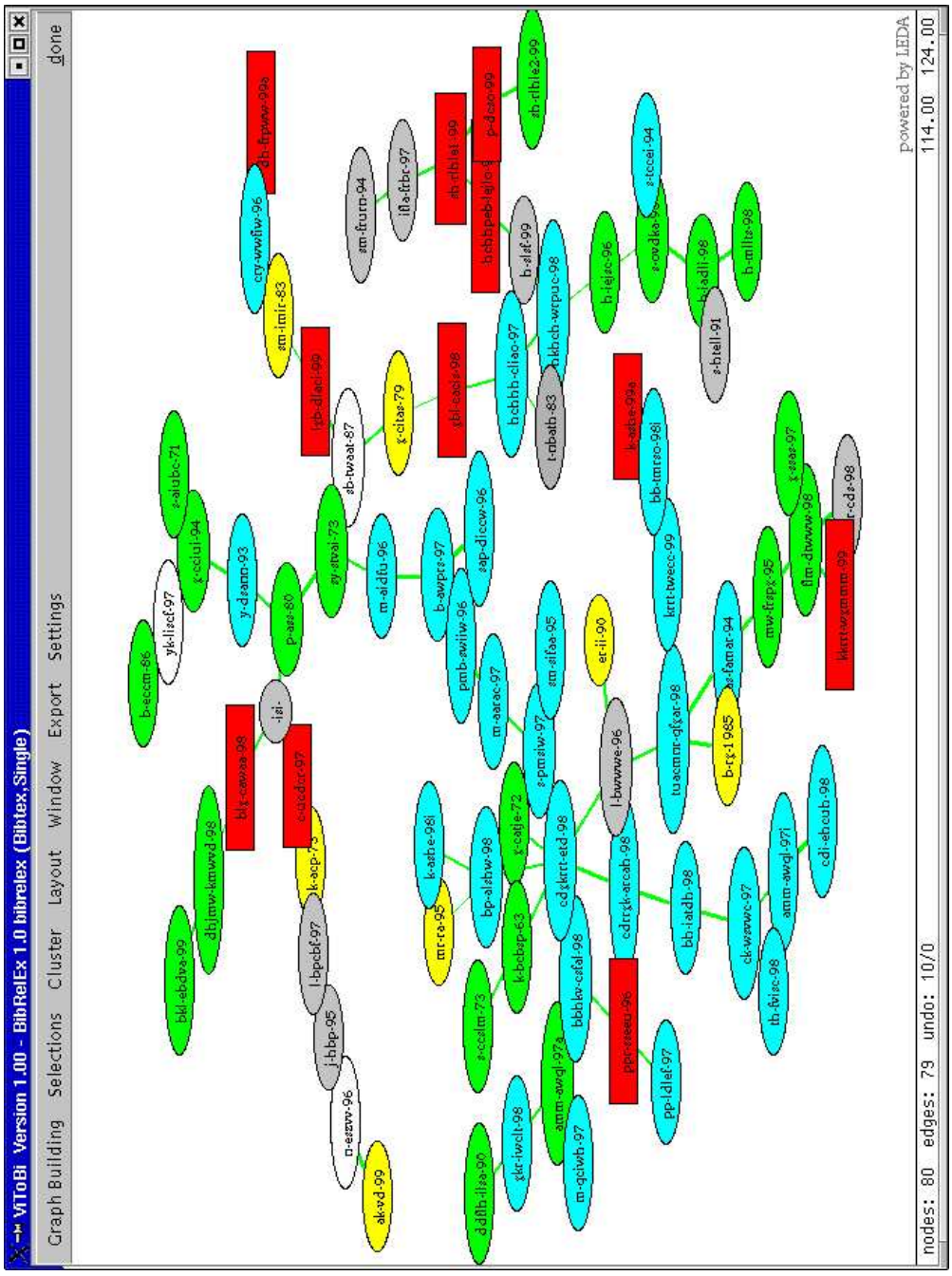


Abbildung 5.10: Minimaler Spannbaum (bibliographische Koppelung)

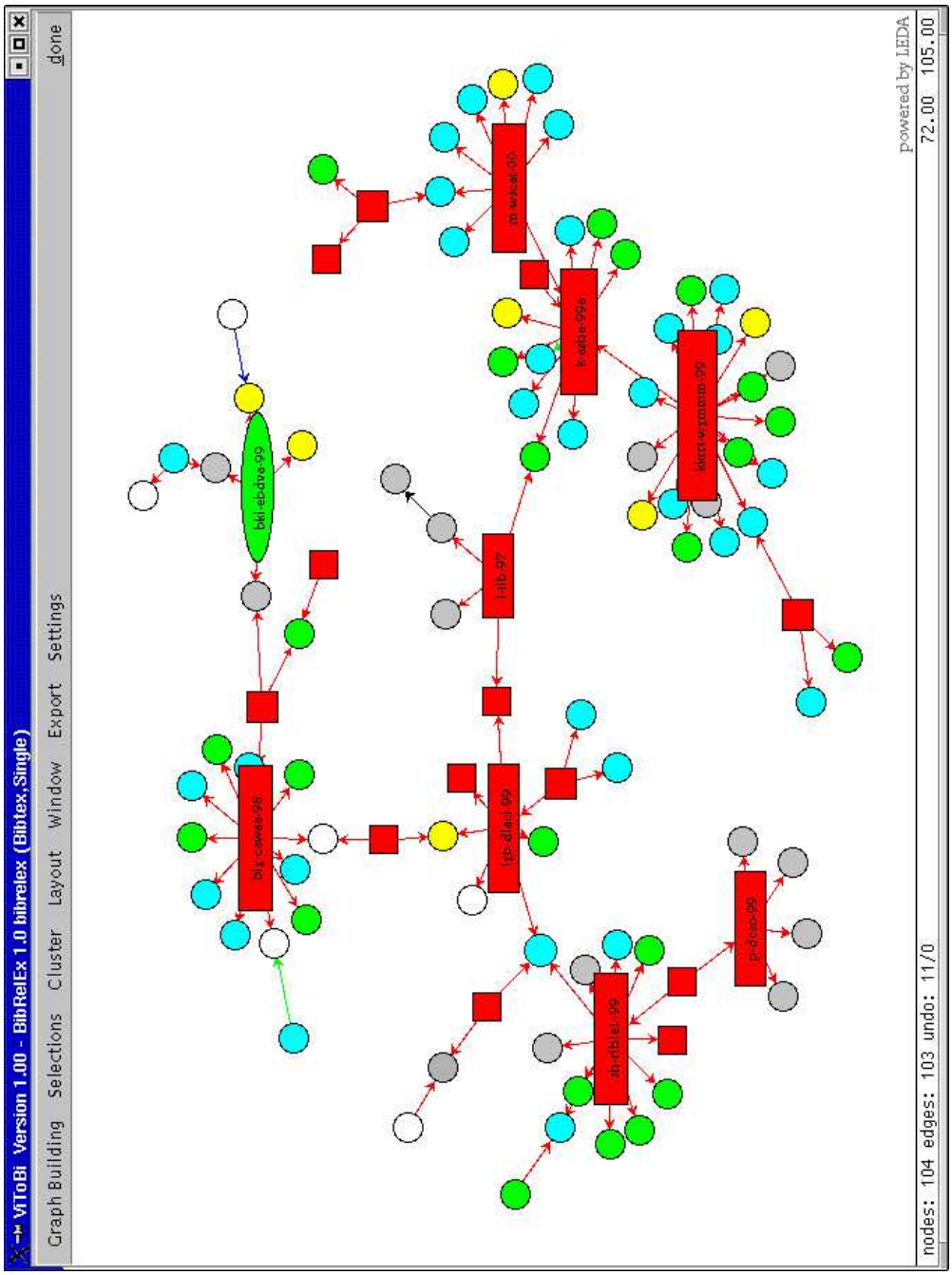


Abbildung 5.11: Minimaler Spannbaum basierend auf der Zitierbeziehung mit Kozitationsgewichtung

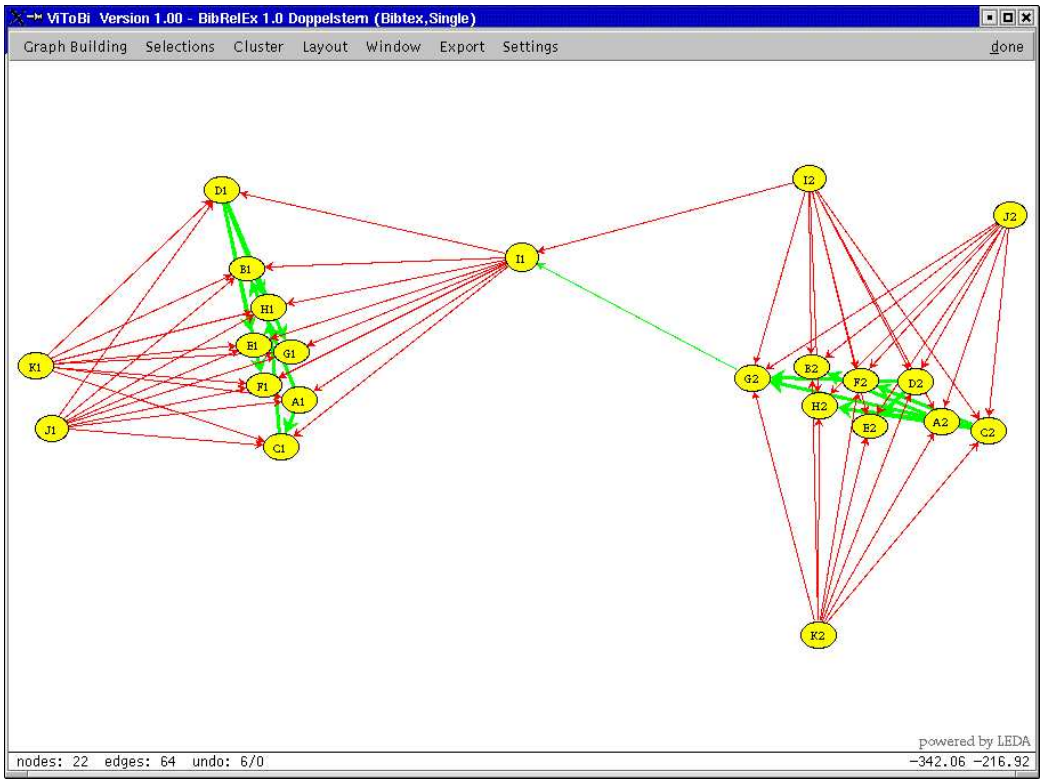


Abbildung 5.12: Kombinierte Darstellung von Zitiernetzwerk und minimalem Spannbaum

Zeitdarstellung

In der Zeitdarstellung wird der Zitiergraph entlang einer Zeitachse angeordnet. Das Layout wird mit Hilfe des Sugiyama-Verfahrens [187] bestimmt, wobei verschiedene Heuristiken zur Kreuzungsminimierung angeboten werden.

Dokumente ohne Angabe des Erscheinungsjahres werden in einer separaten Zeile am unteren Rand der Darstellung in der Jahresspalte des ältesten Dokuments, das dieses zitiert, angeordnet. Cluster bekommen die Jahresspalte des jüngsten in ihm enthaltenen Dokument zugeordnet, damit die *zitiert*-Verweise alle nach links verlaufen und nicht der Eindruck entsteht, dass ein Dokument in der Zukunft zitiert wird.

Abbildung 5.13 zeigt einen Ausschnitt des Zitiergeflechts von Geombib in der Zeitdarstellung unter Verwendung der Barycenter-Heuristik. Dargestellt sind diejenigen Arbeiten, die in irgendeinem Feld das Präfix „Voron“ enthalten.

Die Darstellung gibt deutlich die Geschichte des Voronoi-Diagramms in

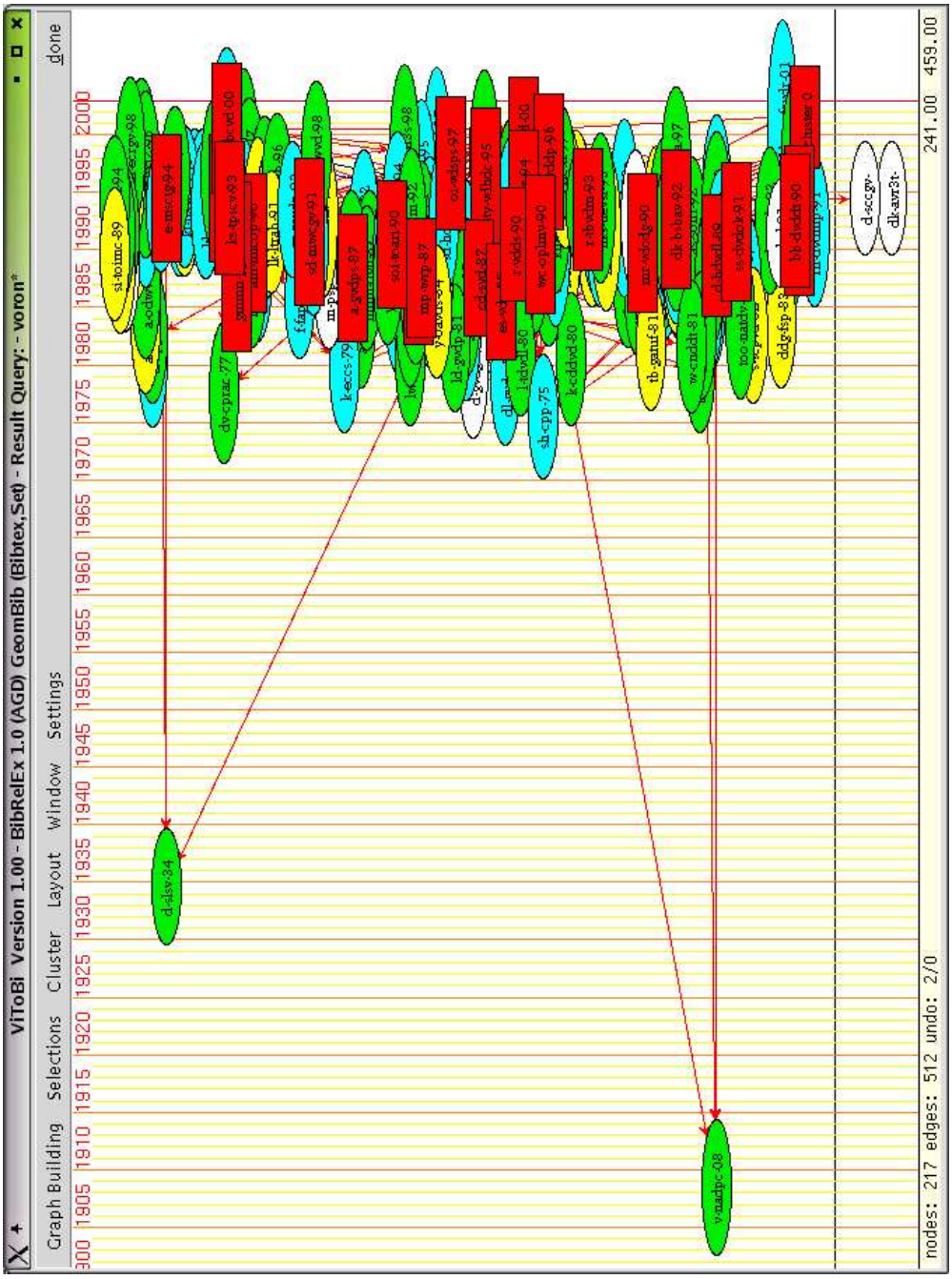


Abbildung 5.13: Zeitdarstellung

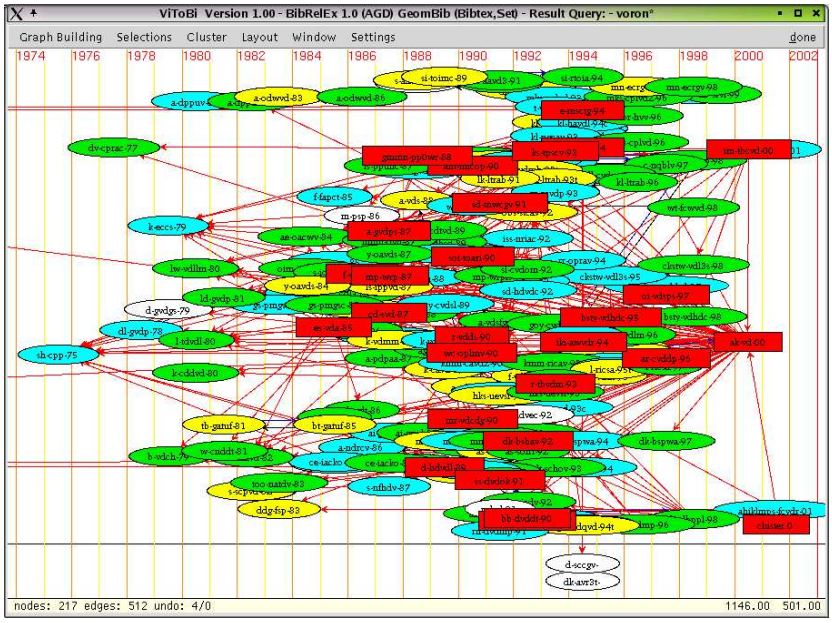
der Algorithmischen Geometrie wieder. Ein Voronoi-Diagramm unterteilt einen Raum für eine gegebene Menge von Punkten so, dass die entstehenden Teilbereiche genau jene Punkte enthalten welche zu den gegebenen Punkten am nächsten liegen. Voronoi-Diagramme werden in den verschiedensten Fachbereichen zur Lösung von Distanzprobleme eingesetzt.

Die beiden Arbeiten [196] und [45] links in der Abbildung sind die Arbeiten der Namensgeber des Voronoi-Diagramms und seiner dualen Struktur, der Delaunay-Triangulation. Die Algorithmische Geometrie entstand mit der Arbeit von Shamos und Hoey [173], die zugleich grundlegend für Voronoi-Diagramme ist. Seitdem sind Voronoi-Diagramme Gegenstand vieler Forschungsarbeiten in der Algorithmischen Geometrie wie ebenfalls gut in der Darstellung zu erkennen ist.

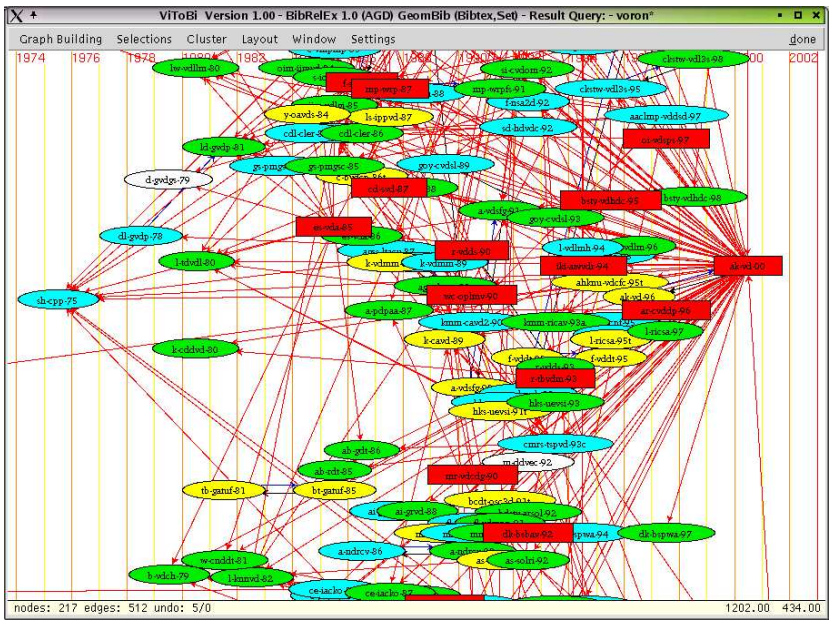
Da bedingt durch die beiden älteren Arbeiten der Namensgeber ein großer Zeitbereich dargestellt wird, sind weitere Details in dieser Abbildung schwer zu erkennen. Ein einfaches Zoomen in den Graphen hinein ist möglich, ist aber für die spezielle Darstellung in Zeitscheiben nicht ausreichend. BibRelEx bietet hier die Möglichkeit auch nur in einzelne Richtungen zu zoomen: in horizontaler Richtung, um die Anordnung entlang der Zeitachse zu entzerren und in vertikaler Richtung, um die Anordnung in den Zeitscheiben zu entzerren. In Abbildung 5.14 ist das selbe Geflecht wie in Abbildung 5.13 dargestellt, wobei zunächst nur in horizontaler Richtung (a) gezoomt wurde, um die Darstellung der neueren Arbeiten zu entzerren und anschließend in vertikaler Richtung (b), um die Arbeiten in einzelnen Zeitscheiben genauer betrachten zu können. Der jeweils interessante Ausschnitt kann mit Hilfe der Maus verschoben werden, so dass alle Teile des Zitiergraphen in der vergrößerten Form vom Benutzer betrachtet werden können.

In der gezoomten Zeitdarstellung erkennt man beispielsweise gut, dass die Arbeit mit dem Zitierschlüssel ak-vd-00 rechts in der Darstellung (*Voronoi Diagrams* von Aurenhammer und Klein [6] im Handbuch *Handbook of Computational Geometry*) eine aktuelle Übersichtsarbeit ist. Diese bietet sich damit als Lektüre an, wenn man sich in das Gebiet der Voronoi Diagramme einarbeiten möchte. Ebenso ist gut in der gezoomten Zeitdarstellung zu erkennen, dass die Arbeit mit dem Zitierschlüssel sh-cpp-75 links in der Darstellung fundamental für das Gebiet der Voronoi-Diagramme ist. Es handelt sich dabei um die bereits oben erwähnte Grundlagenarbeit von Shamos und Hoey [173]. Diese beiden Arbeiten hatten wir auch im Beispiel für die Recherche in BibRelEx mit dem HITS-Algorithmus als Übersichtsarbeit bzw. fundamentale Arbeit identifiziert, vgl. Seite 84.

Mit Hilfe der Zeitdarstellung kann auch leicht die Frage beantwortet werden, welche Folgearbeiten eine Publikation ausgelöst hat. Dazu kann in BibRelEx die Darstellung ausgehend von der entsprechenden Publikation um



(a) horizontales zoomen ...



(b) ... und anschließendes vertikales zoomen

Abbildung 5.14: Zoommöglichkeiten in der Zeitdarstellung

zitierende Arbeiten ergänzt werden. Als Beispiel zeigt die Abbildung 5.15 die Folgearbeiten der Publikation [93].

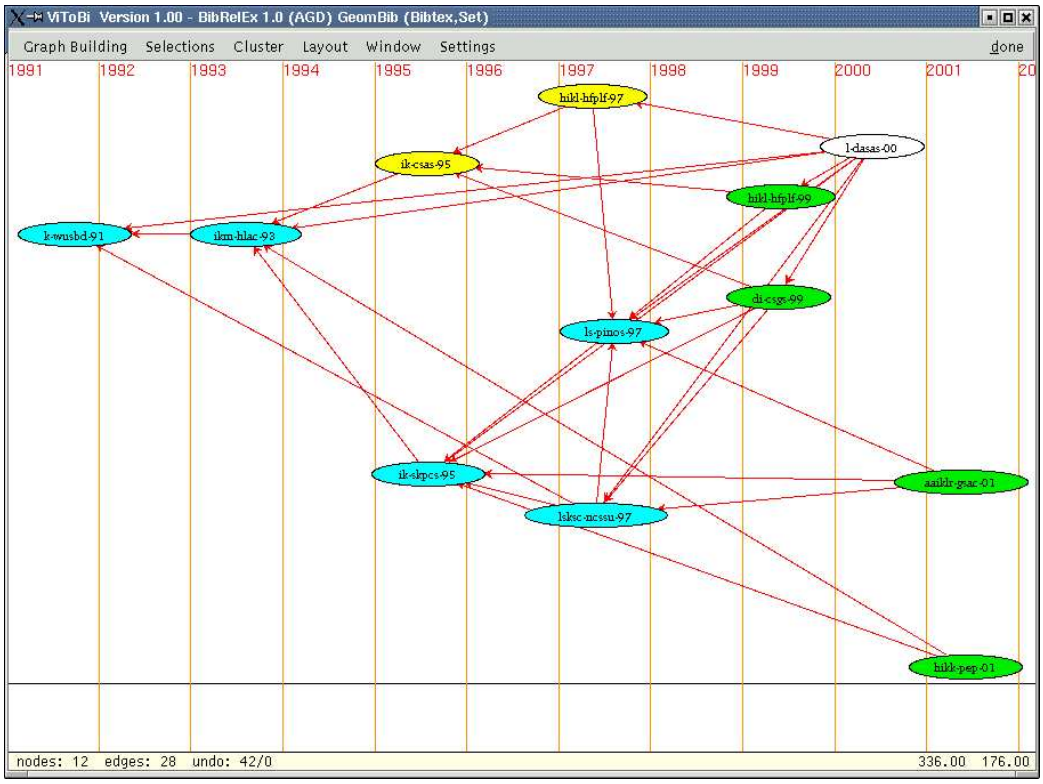


Abbildung 5.15: Folgearbeiten einer Publikation

In BibRelEx sind die kräfte- und energiegesteuerten Verfahren sowohl für zwei- als auch dreidimensionale Darstellungen realisiert worden. Bei den 2D Darstellungen bietet LEDA die Möglichkeit, diese schrittweise zu animieren und auch einen Teil der Knoten festzuhalten. Dies ermöglicht eine dynamische Darstellung unter Wahrung der Mental Map und gibt so dem Benutzer optimale Möglichkeiten, gezielt den Einfluss einzelner Arbeiten zu verfolgen. Auf die Realisierung gehen wir im nächsten Abschnitt ein.

5.6.3 Dynamisches Layout

Bei der Betrachtung der Darstellung eines Graphen machen sich Benutzer mit dieser Darstellung vertraut, indem sie eine kognitive Repräsentation der

Darstellung, die sogenannte „Mental Map“, aufbauen. Beim dynamischen Layout geht es darum, einen geeigneten Kompromiss zwischen der statischen Layoutqualität und der Wahrung der Mental Map des Benutzers zu finden.

Die in BibRelEx verwendeten kräfte- und energiegesteuerten Verfahren haben den Vorteil, dass sie sich leicht zu inkrementellen Verfahren erweitern lassen, um die Anforderungen an ein dynamisches Layout zu erfüllen. So können die einzelnen Iterationsschritte direkt angezeigt werden. Damit kann die Bildung des Layouts wie in einem Film mitverfolgt werden. Zusätzlich kann man die Kräfte nur auf bestimmte Knoten wirken lassen. Dies ermöglicht dem Benutzer den Einfluss der entsprechenden Knoten mitzuverfolgen. Der Ansatz animierter Knotenbewegung wird auch in [50] beschrieben.

In BibRelEx werden alle Knoten entsprechend der jeweiligen Layout-Methode neu positioniert, wenn keine Knoten selektiert wurden. Gibt es selektierte Knoten, so werden nur diese entsprechend der jeweiligen Layout-Methode neu positioniert. Die nicht selektierten Knoten bleiben an ihrer Position, werden aber bei der Berechnung der auf die selektierten Knoten wirkenden Kräfte berücksichtigt.

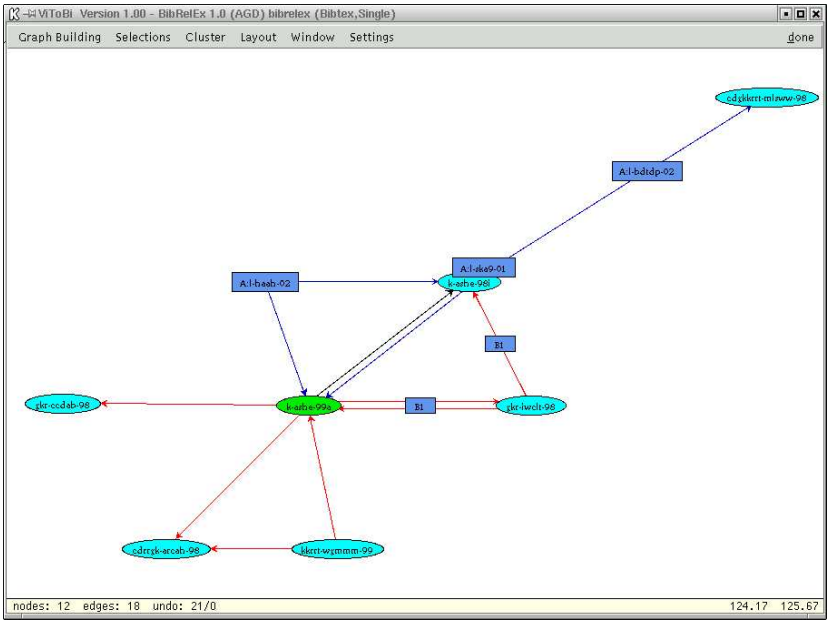
Bei der Ergänzung der Darstellungsmenge durch eine Anfrage an die Datenbasis werden die neu eingefügten Knoten automatisch selektiert. Wählt der Benutzer nun ein 2D Layout-Verfahren aus, werden nur die neu hinzugekommenen Knoten unter Anzeige der Zwischenschritte neu positioniert. Der Benutzer kann so die Bewegung der Knoten verfolgen, erkennt leichter den Einfluss der Knoten (Dokumente) und kann sich besser orientieren (Wahrung der Mental Map).

Die Abbildungen 5.16(a) bis 5.16(d) geben exemplarisch die Veränderungen des Layouts bei der Ergänzung der Darstellung um Anfrageergebnisse wieder. Der Übergang zwischen den einzelnen Abbildungen erfolgt dabei in Zwischenschritten, die Aufgrund der Beschränkung des Mediums Papier hier nicht wiedergegeben werden können. Dennoch zeigen die Bilder, dass sich die Änderung des Layouts gut nachvollziehen lässt.

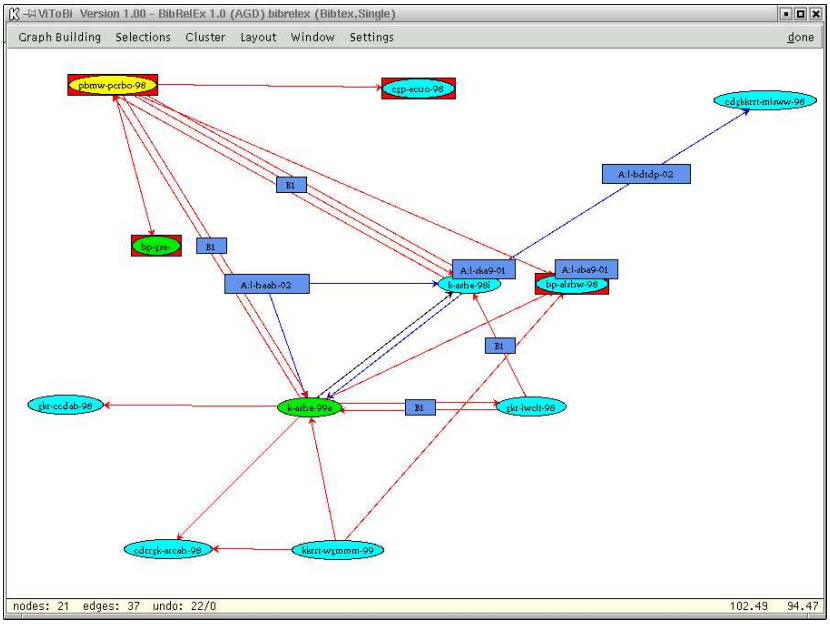
Zusätzlich hat der Benutzer selbst die Möglichkeit Knoten zu selektieren, beispielsweise mit Hilfe von Filterbedingungen. So kann er gezielt den Einfluss von Arbeiten bestimmter Autoren oder zu einem bestimmten Thema studieren.

Das Verfahren lässt sich noch weiter verfeinern, indem (bestimmen) Knoten ein Alter zugewiesen wird und diese Knoten dann abhängig vom Alter unterschiedlich gewichtet ins Layout eingehen und mitbewegt werden. Dieser Ansatz ist auch schon von Brandes und Wagner [17] vorgeschlagen worden.

Bei den 3D Darstellungen wird jeweils das Layout des gesamten Graph neu berechnet und abschließend animiert. Dabei wird der Graph in Rotation gezeigt, wobei der Benutzer mit Hilfe der Maus die Drehgeschwindigkeit und

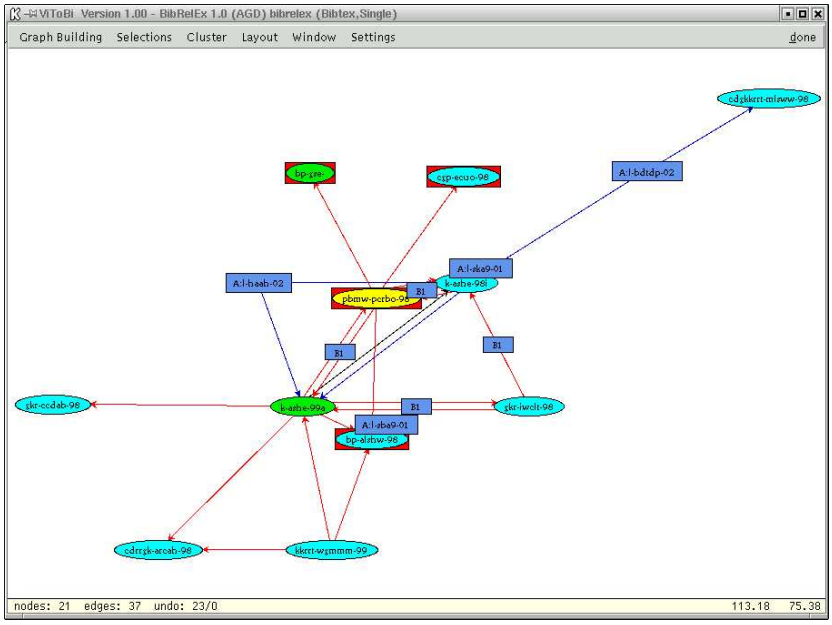


(a) Ursprüngliche Darstellung

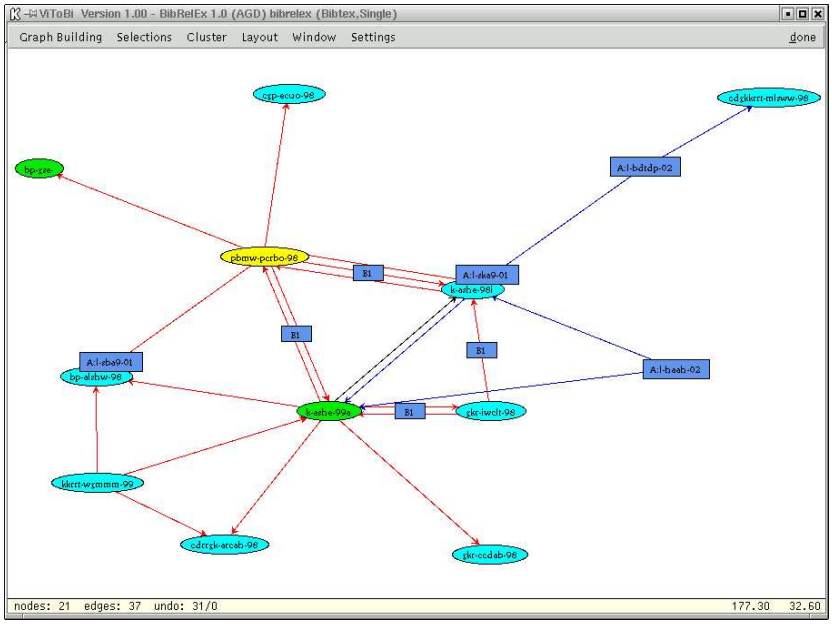


(b) Ergänzung um Anfrageergebnisse

Abbildung 5.16: Abfolge beim dynamischen Layout



(c) Neuplatzieren der Anfrageergebnisse



(d) Neuplatzieren der gesamten Darstellung

Abbildung 5.16: Abfolge beim dynamischen Layout

-richtung steuern kann. Über die Maus ist auch ein Ein- und Auszoomen der drehenden Darstellung möglich. Die 2D und 3D Darstellungen ergänzen sich somit: Mit Hilfe der 3D Darstellung gewinnt der Benutzer einen Eindruck der Gesamtstruktur, indem er das Geflecht in seiner Gesamtheit von allen Seiten betrachten kann. Die 2D Darstellung erlaubt ihm sich Einzelheiten genauer anzusehen.

Eine weitere wesentliche Fragestellung beim dynamischen Layout ist die Behandlung von Cluster. Damit die Mental Map des Benutzers bei der dynamischen Clusterung erhalten bleibt, sollten Knoten, die vorher in einem Cluster waren, nach Änderungen in diesen Clustern bleiben. Bleibt die Frage, ob neu in einer Darstellung hinzugekommende Knoten direkt in die entsprechenden Cluster eingeordnet werden sollen? Gegebenenfalls könnten neu hinzugekommende Knoten eine Reorganisation der globalen Clusterstruktur verursachen. Das gleiche gilt für das Entfernen oder Ausblenden von Knoten, beispielsweise durch Anwenden einer Filterbedingung.

Hier sind wir der Meinung, dass eine sofortige Einordnung neuer Knoten in die vorhandenen Cluster und eine eventuell daraus resultierende Umorganistation der Cluster für den Benutzer eher verwirrend ist. Nicht zu vernachlässigen ist außerdem, dass eine Reorganisation der Clusterstruktur für jede Änderung einen erheblichen Zeitaufwand erfordert.

Durch die in BibRelEx verwendeten Layout-Verfahren werden neue Knoten in der Nähe zu den Clustern angeordnet, zu denen sie einen inhaltlichen Bezug haben. Der Benutzer kann dann in einem folgenden Schritt eine Reclusterung anfordern.

Als weitere Unterstützung bei der Orientierung im Falle des Wechsels der Darstellungsart, beim Auflösen von Clustern oder beim Ein- und Ausblenden von Knoten durch Filterbedingungen, werden Knoten sobald sie wieder sichtbar werden, an ihrer alten Position in der entsprechenden Darstellungsart angeordnet. Damit kann der Benutzer auch testweise andere Darstellungsarten und Clusterverfahren ausprobieren und erhält beim rückgängigmachen der entsprechenden Operation wieder die alte Darstellung.

Zu einem dynamischen Layout gehört auch, dass auf Änderungen in der Datenbasis direkt reagiert wird. Das ist zur Zeit aufgrund von Problemen im Zusammenspiel zwischen den verwendeten Bibliotheken LEDA und QT in BibRelEx nicht automatisch ohne erheblichen Mehraufwand möglich. Der Benutzer kann aber die Aktualisierung der Darstellung bei Bedarf über das Menü selbst veranlassen. Eine etwas ausführlichere Darstellung dieses Problems findet sich bei der Beschreibung der Implementierung in Abschnitt 6.5.

Eine weitere Orientierungshilfe für Benutzer bietet das parallele Erforschen von verschiedenen Teilen des Graphen. In BibRelEx ist dies durch voneinander unabhängige Visualisierungsfenster für Cluster möglich. Der Benutzer kann Teile des Graphen markieren, zu einem Cluster zusammenfassen und in einem neuen Fenster separat visualisieren lassen. Daneben gibt es verschiedene Möglichkeiten die Darstellung nach inhalts- oder graphbasierten Kriterien zu clustern. Diese werden im folgenden Abschnitt beschrieben.

5.6.4 Clusterung

Eine weitere Orientierungshilfe für den Benutzer ist die Bildung von Clustern. Mit ihrer Hilfe lässt sich der Detaillierungsgrad der Darstellung des Graphen reduzieren und kann eine Voreinteilung der Dokumente nach bestimmten Kriterien getroffen werden. In BibRelEx kann sich der Benutzer für jeden Cluster ein eigenes Visualisierungsfenster anzeigen lassen und so gleichzeitig verschiedene Teile des Graphen erforschen. Durch Selektieren von Knoten können diese zu einem Cluster zusammengefasst werden und so gezielt bestimmte Bereiche eines Graphen als Clusterinhalt in einem separaten Fenster visualisiert werden. Die Visualisierungsfenster für Cluster bieten die gleichen Layout-Methoden wie das Visualisierungs-Hauptfenster, so dass der Benutzer sich die Cluster mit unterschiedlichen Verfahren darstellen lassen, Filter angewendet werden können und Untercluster gebildet werden können. Jedes der Visualisierungsfenster verfügt auch über einen eigenen Satz von Einstellungen, die beim öffnen des Fensters vom Elternfenster übernommen werden und individuell angepasst werden können. So können Untercluster in verschiedenen Fenstern unterschiedlich angeordnet werden oder die Anzeige von Knoten und Kanten individuell variiert werden.

Mit Hilfe der Voreinstellungen kann der Benutzer auch festlegen ob, ab welcher Knotenzahl und nach welchen Verfahren ein Graph bei seiner ersten Darstellung vorgeclustert werden soll. Damit kann von vornherein die Darstellung zu vieler Details vermieden werden. Da das Ziel der Visualisierung von BibRelEx die Erforschung der Beziehungen zwischen den Dokumenten ist, hat sich ein sehr simples Verfahren zur Vorclusterung bewährt: Zunächst werden alle Knoten mit einem Knotengrad von 0 zu einem Cluster zusammengefasst. Als nächstes werden alle Knoten, die genau nur einen eingehenden und keinen ausgehenden Verweis besitzen mit dem auf sie verweisenden Knoten zusammengefasst. Übertragen auf die Zitierrelation bedeutet dies, dass alle zitierten Arbeiten mit der sie zitierenden Arbeit zu einem Cluster zusammengefasst werden, wenn sie von keiner weiteren Arbeit zitiert werden

und selbst keine anderen Arbeiten zitieren. Durch dieses Vorgehen werden also alle Knoten zusammengefasst, die keinen wesentlichen Beitrag zur Gesamtstruktur des Beziehungsgeflechts leisten. Alternativ kann MajorClust zur Vorclusterung verwendet werden. MajorClust spiegelt die „natürliche“ Struktur des Graphen wieder.

In BibRelEx kann kann der Benutzer zwischen verschiedenen Cluster-Verfahren wählen. Eine Übersicht der verschiedenen Cluster-Verfahren gab der Abschnitt 3.2.5. Hier wird nun beschrieben, wie diese Verfahren in BibRelEx ggf. mit welcher Modifikation Anwendung gefunden haben und Verwendungsbeispiele gegeben.

Semantikbasierte Cluster-Verfahren

Im Bereich der semantischen Clusterung bietet BibRelEx die Möglichkeit, Dokumente nach textbasierter Gewichtung, Kozitation und bibliographischer Koppelung zusammenzufassen. Die Clusterung erfolgt mit einem agglomerativen hierarchischem Verfahren, wobei der Abstand zweier Cluster mittels Average Linkage, siehe Seite 3.2.5, ermittelt wird.

Single Linkage ist zwar schneller, erzeugt aber in der Regel eine geringe Anzahl großer Cluster und ist damit für die Kozitationsanalyse nicht geeignet [179]. Ein einziges Kozitat zwischen zwei verschiedenen Clustern kann bei Single Linkage dazu führen, dass die Cluster verschmolzen werden, obwohl kein engerer Zusammenhang zwischen den Dokumenten der beiden Cluster besteht. Ein Beispiel dafür zeigt Abbildung 5.17.

Textbasierte Retrieval-Methoden sind hinreichend bekannt und nicht Forschungsgegestand von BibRelEx. Unser Schwerpunkt liegt auf dem zusätzlichen Nutzen strukturbasierter Daten. Daher ist es zu vertreten, zunächst für den Prototypen von BibRelEx ein einfaches wohlbewährtes Verfahren zur Termindizierung basierend auf dem Vektorraummodell zu verwenden. Später kann das Verfahren leicht gegen effizientere Verfahren ausgetauscht werden und auch ein Preprocessing in Betracht gezogen werden.

Die Termindizierung operiert dabei auf benutzerwählbaren Feldern der Dokumentendaten. Sie weist jedem Dokument einen Wortvektor zu, der die Worthäufigkeiten enthält. Die Ähnlichkeit zweier Dokumente $d_i, d_j \in D$ aus einer Dokumentenmenge $D = \{d_1, \ldots, d_m\}$ wird mit Hilfe des Kosinuskoeffizienten

$$s_{\cos}(d_i, d_j) = \frac{\sum_{k=1}^{n} w_{i,k} w_{j,k}}{\sqrt{\sum_{k=1}^{n} w_{i,k}^2} \sqrt{\sum_{k=1}^{n} w_{j,k}^2}}$$

berechnet, wobei jedes Dokument $d_i \in D$ durch Gewichte $w_{i,k} \in \mathbb{R}$ zu jedem Term $t_k \in T$ einer endlichen Menge von Indextermen $T = \{t_1, \ldots, t_n\}$ beschrieben wird.

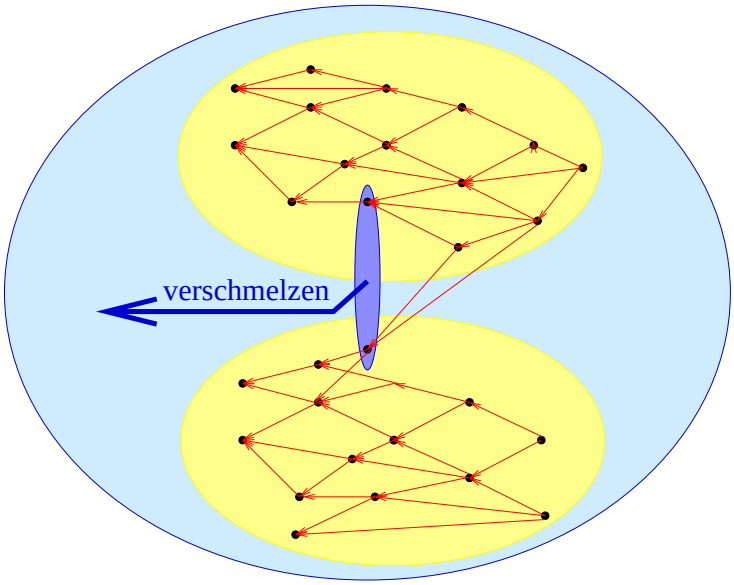


Abbildung 5.17: Beispiel für ungünstige Verschmelzung von Clustern bei Single Linkage [144]

Die Felder, die dabei berücksichtigt werden sollen, können über Optionen gesetzt werden. So ist auch eine Clusterung nach Autoren oder Schlüsselworten möglich.

Um zu zeigen, wie mit Hilfe beliebiger Beziehungen Wissensstrukturen aufgebaut und mit Hilfe der Verwendung semantischer Clusterverfahren weiter strukturiert werden können, wird an dieser Stelle nochmal auf das auf mehrere Themen erweiterte Nutzungsbeispiel 4.1 *Organisation eines Seminars* zurückgegriffen (Abschnitt 5.6.2). Das zugehörige Beziehungsgeflecht ist in Abbildung 5.5 dargestellt. Abbildung 5.18 zeigt den geclusterten Graphen und die entstehenden Cluster bei Clusterbildung nach Wortgewichtung auf dem *keyword*-Feld[8]. Die einzelnen Cluster zeigen die Arbeiten zu jeweils einem Seminarthema.

Als nächstes betrachten wir die Clusterung nach der Kozitation. Das in diesem Beispiel verwendete Beziehungsgeflecht basiert auf Annotationen. Daher wird der bei der Clusterung zu berücksichtigende Linktyp auf „annote“ gesetzt. Die Abbildungen 5.19(a) bis 5.19(h) zeigen den resultierenden Graphen und die zugehörigen Cluster. In diesem Fall werden durch die Clusterbildung diejenigen Arbeiten zusammengefasst, die dieselben Kriterien erfüllen.

[8]In BibRelEx werden die Cluster in eigenen Fenstern angezeigt. Diese sind in der Abbildung zur optimalen Platzausnutzung über das Visualisierungsfenster des gesamten Graphen geschoben worden.

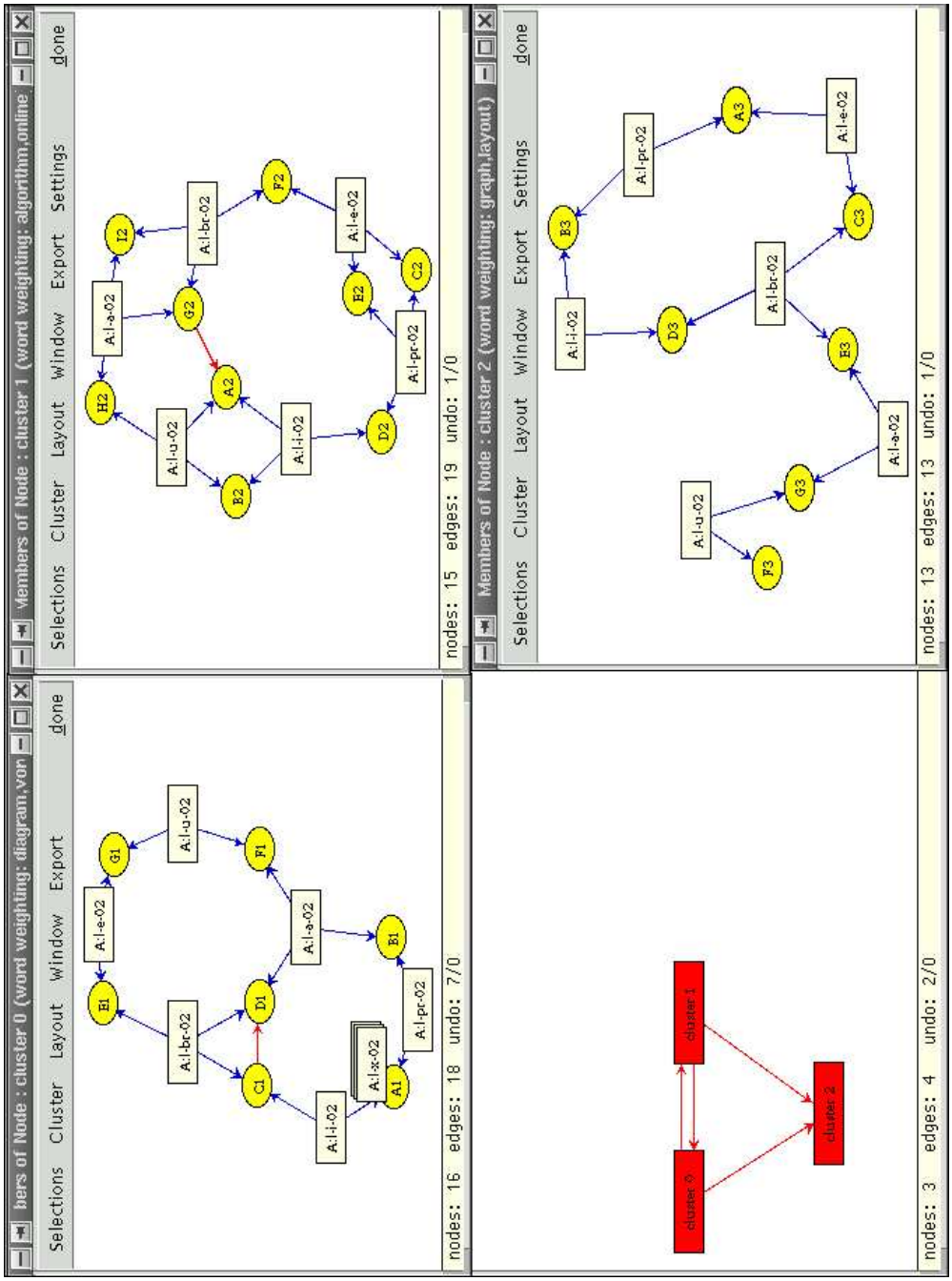


Abbildung 5.18: Seminarbeispiel, mehrere Themen, Clusterung mit Wortgewichtung auf dem Keywords-Feld[8]

Beispielsweise enthält Cluster 2 alle Arbeiten (unabhängig vom Thema), die als „Primary Reading“ und „Experienced“ gewertet wurden.

Ein schnelleres heuristisches Verfahren zur Clusterung basierend auf der Kozitation wurde von Popescul u. a. [155] vorgestellt. Sie gehen von der Annahme aus, dass wissenschaftliche Disziplinen sich um einflussreiche Arbeiten bilden und dass das Zitieren dieser Arbeiten ein wesentlicher Indikator dafür ist, wie eine Arbeit eingestuft werden sollte. Sie verwenden die am häufigsten zitierten Arbeiten als Clusterzentroide. Da neuere Arbeiten jedoch noch nicht so oft zitiert werden, langt es nicht, einfach die Anzahl der Zitate zu zählen. Popescul u. a. [155] normalisieren daher die Zitierrate einer Arbeit mit der Anzahl der Arbeiten, die nach dem Erscheinen dieser Arbeit erschienen sind. Als einflussreiche Arbeiten werden also diejenigen angesehen, deren normalisierte Zitierrate über einen Schwellwert liegt. Jede einflussreiche Arbeit wird mit allen Arbeiten, die gemeinsam mit ihr zitiert werden, zu einem sogenannten Softcluster zusammen gefasst. Abschließend werden die Softcluster mit einem klassischen Clusterverfahren zusammen gefasst. Dabei wird die Ähnlichkeit der Cluster nach folgendem Ähnlichkeitsmaß

$$\frac{|C_1 \cap C_2|}{|C_1| + |C_2| - |C_1 \cap C_2|} (= \frac{\text{Anzahl gemeinsamer Arbeiten}}{\text{Anzahl verschiedener Arbeiten}})$$

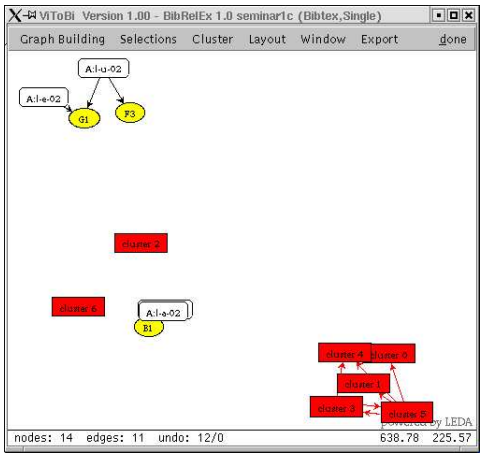
bestimmt. Dieses Verfahren ist ebenfalls in BibRelEx implementiert worden und empfiehlt sich bei großen Graphen als Alternative zu den vorher vorgestellten exakten Verfahren.

Graphbasierte Cluster-Verfahren

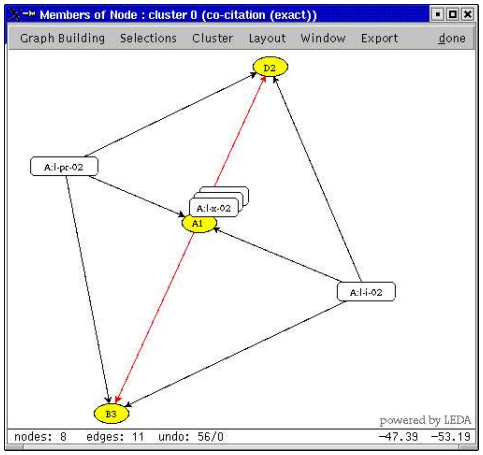
Für die Clusterung basierend auf der Struktur des Graphen wurden zunächst einige Algorithmen genutzt, die LEDA für Graphen bietet. So besteht die Möglichkeit, alle schwachen oder starken Zusammenhangskomponenten[9] des Graphen zu Clustern zusammen zu fassen. Diese beiden Methoden liefern jedoch nur sehr bedingt brauchbare Ergebnisse, da zu viele Knoten jeweils zu einem Cluster zusammen gefasst werden. Desweiteren wurden zwei auf Breitensuche basierende Algorithmen implementiert, die stark vernetzte Knoten mit ihren Kindknoten vorgegebener Tiefe zu einem Cluster zusammenfassen.

Die besten Ergebnisse bezüglich der Strukturwiedergabe des Wissensgeflechtes wurden bei der Clusterung mit dem MajorClust-Verfahren erzielt, dass zudem noch ein sehr gutes Laufzeitverhalten zeigt.

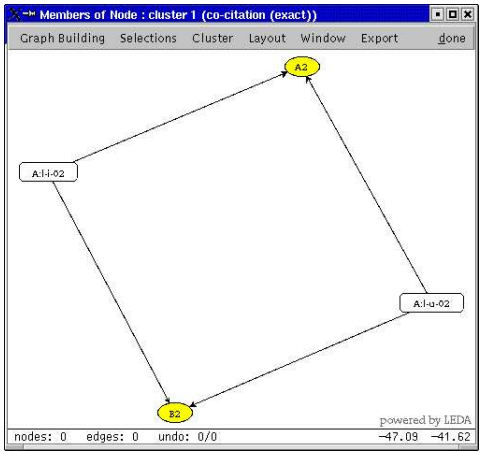
[9] Eine Zusammenhangskomponente ist der maximale Teilgraph bzgl. der betrachteten Zusammenhangseigenschaft. Ein Graph ist schwach bzw. stark zusammenhängend genau dann, wenn es zwischen je zwei Knoten des Graphen einen ungerichteten bzw. gerichteten Pfad zwischen ihnen gibt.



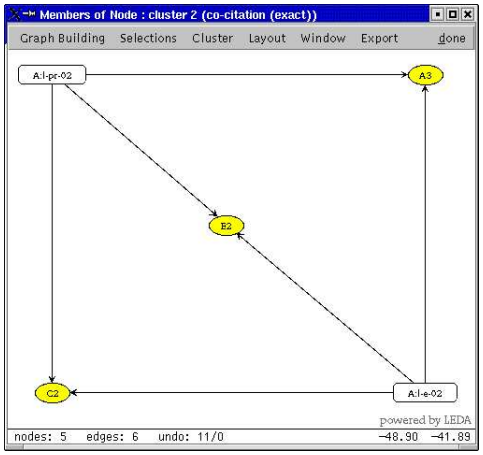
(a) geclusterter Gesamtgraph



(b) Cluster 0

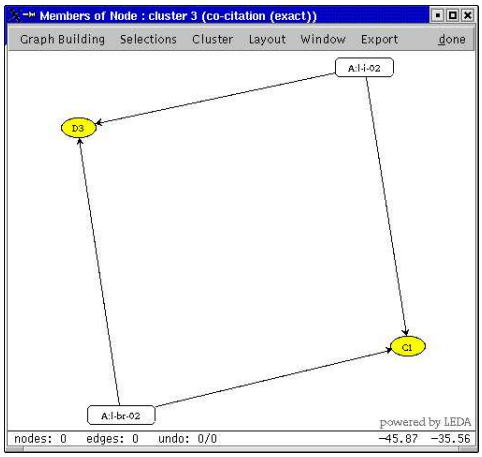


(c) Cluster 1

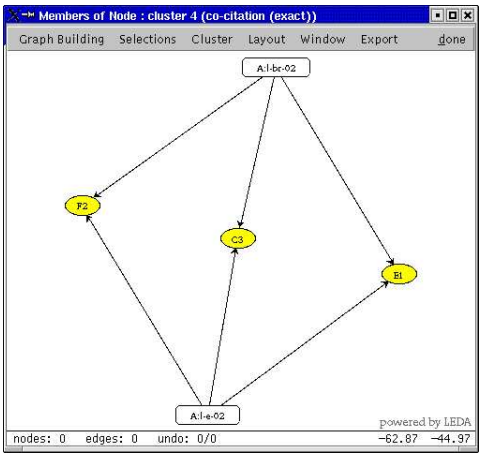


(d) Cluster 2

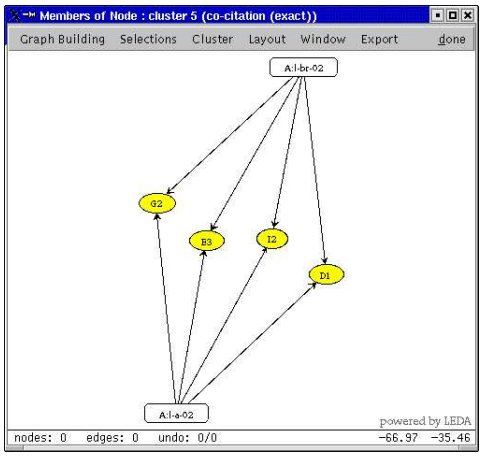
Abbildung 5.19: Seminarbeispiel, mehrere Themen, Clusterung nach Kozitation auf den Annotationsbeziehungen



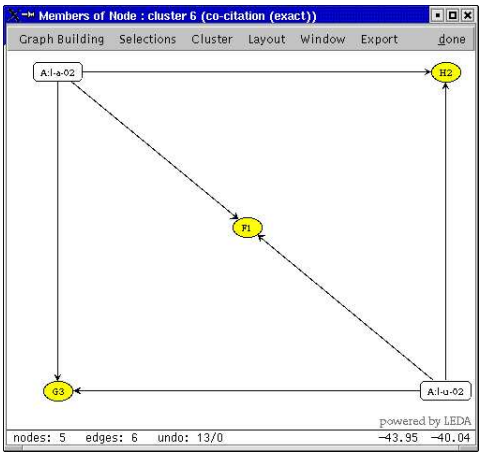
(e) Cluster 3



(f) Cluster 4



(g) Cluster 5

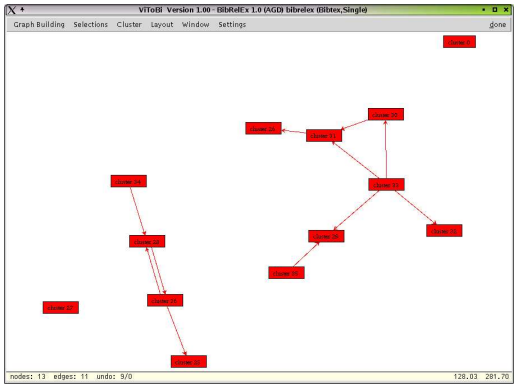


(h) Cluster 6

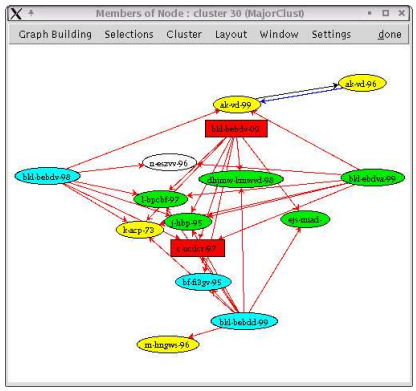
Abbildung 5.19: Seminarbeispiel, mehrere Themen, Clusterung nach Kozitation auf den Annotationsbeziehungen

Im MajorClust-Algorithmus (Algorithmus 7 in Abschnitt 3.2.5) wird derjenige Cluster für einen Knoten gewählt, der die meisten seiner Nachbarknoten enthält. Gibt es mehrere solcher Cluster, erfolgt die Auswahl zufällig aus diesen. Neben dieser zufälligen Auswahl wurde in BibRelEx zusätzlich eine semantikbasierte Auswahl des Clusters implementiert: Gibt es zu einem Knoten mehrere Cluster mit der maximalen Zahl von Nachbarknoten, so wird aus diesen der Cluster gewählt, der bzgl. der Wortgewichtung die größte Ähnlichkeit mit dem Knoten hat. Gibt es hier wiederum mehrere, so wird erst aus dieser (kleineren) Clustermenge einer zufällig ausgewählt.

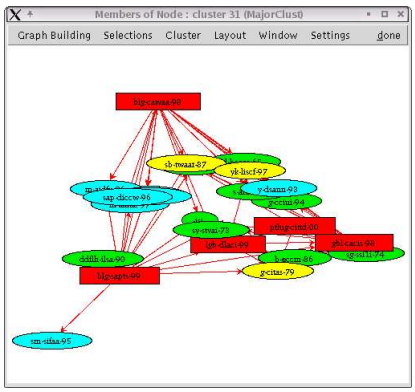
Betrachten wir nocheinmal das Zitiergeflecht dieser Dissertation, vergl. Abschnitte 4.4 und 5.6. Aufgrund der Größe der Datenbasis wurde wieder nach dem auf Seite 99 beschriebenen einfachen Clusterverfahren vorgeclustert. Abbildung 5.20(a) zeigt die resultierenden Darstellung nach anschließendem Anwenden des MajorClust-Verfahrens. Die Datenbasis wurde dadurch klar in nur noch wenige „Themengebiete“ unterteilt. Beispielsweise zeigt Cluster 30 (Abbildung 5.20(b)) unsere Veröffentlichungen zu BibRelEx und ihre häufigsten gemeinsamen Zitate [23, 24, 26, 27], Cluster 31 (Abbildung 5.20(c)) Veröffentlichungen aus dem Projekt Citeseer [13, 14, 70, 112, 155] und Cluster 33 (Abbildung 5.20(d)) Arbeiten, die sich mit der Analyse der Linkstruktur des WWW befassen [18, 43, 95, 96, 135, 150, 153, 203].



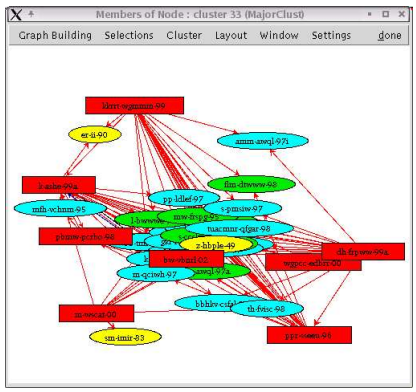
(a) Zitiergeflecht



(b) Cluster 30



(c) Cluster 31



(d) Cluster 33

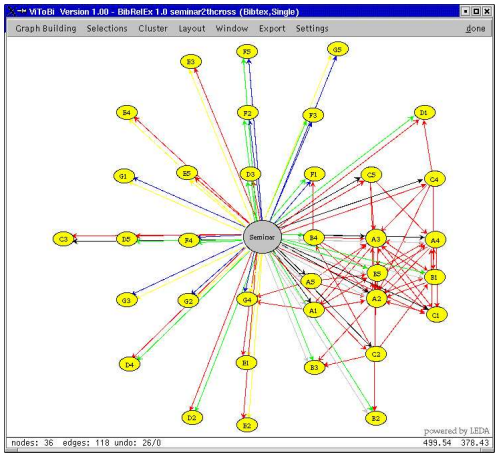
Abbildung 5.20: Beispiel für die Anwendung des MajorClust-Verfahrens

5.6.5 Link-Aggregation

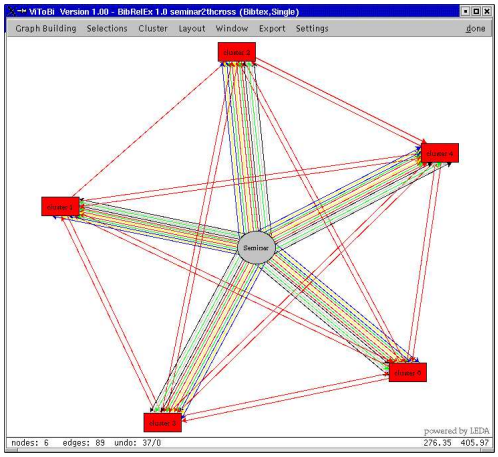
Eine weitere Möglichkeit die Skalierbarkeit und Übersichtlichkeit zu erhöhen ist die Aggregation von Links. Dabei werden die verschiedenen individuellen Links zwischen zwei Knoten zu einem Link zusammengefasst. Dies erhöht insbesondere die Übersichtlichkeit bei geclusterten Darstellungen, da gerade zwischen Clustern viele verschiedene Links verlaufen. Durch die Aggregation von Links geht ein geringer Anteil an Information in der Darstellung verloren. Dafür erhöht sich die Klarheit der Darstellung aber enorm. Um den Informationsverlust so gering wie möglich zu halten, kann der aggregierte Link gewisse Eigenschaften der ursprünglichen Links repräsentieren. Beispielsweise gibt die Dicke des aggregierten Links die Anzahl der ursprünglichen einzelnen Links zwischen den Knoten wieder. Darüber hinaus werden in BibRelEx eingehende und ausgehende Links getrennt aggregiert und unterschiedliche Typen von Links ebenfalls in 4 Kategorien (cites, precedes, succeeds, bibliographische Links) getrennt aggregiert. Noch größere Flexibilität wird dadurch erreicht, das die Link-Aggregation für jedes Visualisierungsfenster getrennt einfach an- und abgeschaltet werden kann, so dass zwischen den einzelnen Sichten hin- und hergeschaltet werden kann und verschiedene Cluster individuell mit und ohne Aggregation angezeigt werden können, siehe auch Abschnitt 5.6.4. Zusätzlich erhält der Benutzer beim Anwählen eines aggregierten Links eine Auswahlliste der einzelnen Links.

Die Methode der Link-Aggregation wurde auch von Brown [22] zur Reduktion der Details bei der Visualisierung von Hypertextdatenbasen verwendet. Die Möglichkeiten der Link-Aggregation sind in BibRelEx allerdings flexibler. Beispielsweise kann in BibrelEx nach Linkarten unterschieden werden und die Link-Aggregation für jedes Visualisierungsfenster unabhängig gesteuert werden. Außerdem kann der Benutzer in BibRelEx für jeden aggregierten Link auch Informationen über die einzelnen Links abfragen.

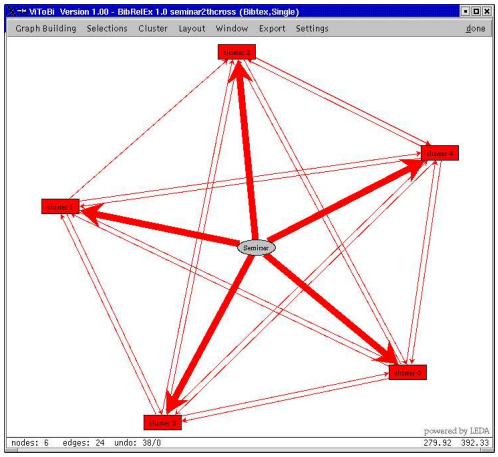
Abbildung 5.21 zeigt die Link-Aggregation anhand des auf 5 Themen erweiterten Nutzungsbeispiels 4.1 „Organisation eines Seminars“ für den Fall, dass das Wissen durch typisierte Links eingebracht wurde. Zusätzlich wurden noch einige Zitierbeziehungen ergänzt.



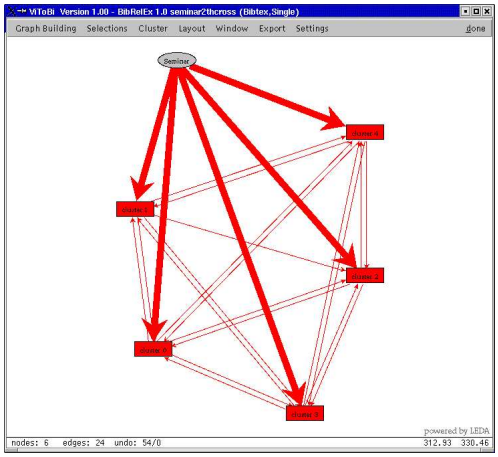
(a) Ausgangsgraph nach Spring Embedding



(b) nach textbasierter Clusterung



(c) mit Link-Aggregation



(d) und nochmaliger Anwendung des Spring Embedders

Abbildung 5.21: Link-Aggregation

Kapitel 6

Entwurf und Implementierung

Nachdem wir nun die verschiedenen Konzepte und Algorithmen von BibRelEx betrachtet haben, wird in diesem Kapitel der konkrete Entwurf und die Implementierung des Prototypen von BibRelEx vorgestellt. Dabei geht es nicht um eine ausführliche Beschreibung sämtlicher Implementierungsdetails, wozu auf die eigentliche Programmdokumentation, z.B. [106, 107, 108, 110], verwiesen sei. Vielmehr sollen hier grundlegende Konzepte und Ideen vorgestellt werden, die beim Gesamtentwurf und der Implementierung eine Rolle gespielt haben. Dazu werden zunächst in Abschnitt 6.1.1 in einem Exkurs zu Software Engineering verschiedene Entwurfsmuster vorgestellt. Auf diese Entwurfsmuster wird in der weiteren Beschreibung immer wieder zurückgegriffen. Anschließend stellen wir das Datenmodell von BibRelEx, das ein flexibles, benutzerdefinierbares Format der Literaturdaten ermöglicht, vor. Wesentlich für den flexiblen Einsatz von BibRelEx ist darüber hinaus die Anbindung verschiedener Datenhaltungssysteme. In BibRelEx wird dies mit Hilfe eines Hüll-Objektes für den Datenbestand, dem sogenannten Wrapper, realisiert. Exemplarisch stellen wir hier die Wrapper-Implementierung für BibTEX-Datenbasen vor. In BibRelEx wird mit einer Vielzahl von Objekten und Sichten (Editorfenster, Visualisierer, Eintragslisten u.a.) gearbeitet, die in ihrem Zustand oder Verhalten voneinander abhängig sind. Solche Abhängigkeiten können effizient mit Hilfe des Beobachter-Entwurfsmuster realisiert werden, dessen Umsetzung wir als letztes Entwurfskonzept ausführlicher vorstellen wollen. Im Anschluss wird ein Überblick über die in BibRelEx verwendeten Konzepte zur Unterstützung der Benutzungsfreundlichkeit gegeben. Nach einem kurzen Blick auf die Gesamtstruktur der Implementierung von BibRelEx folgen drei Abschnitte, die die Implementierung der Datenhaltung, der Visualisierung und der graphischen Benutzeroberfläche näher beschreiben.

6.1 Ausgewählte Entwurfskonzepte

6.1.1 Entwurfsmuster

Ein Entwurfsmuster beschreibt ein bestimmtes, in einem gegebenen Kontext immer wiederkehrendes Entwurfsproblem und stellt ein vorgegebenes Schema zu seiner Lösung zur Verfügung. Entwurfsmuster fassen Expertenwissen zusammen und unterstützen den Entwurf wiederverwendbarer objektorientierter Software. Bei der Entwicklung von BibRelEx wurden verschiedene Entwurfsmuster konsequent eingesetzt. Unter anderem wurden die folgenden Entwurfsmuster verwendet, siehe Gamma u. a. [65]:

- *Abstrakte Fabrik*: Schaffung einer Schnittstelle zum Erzeugen von verwandten oder voneinander abhängigen Objekten, ohne die konkrete Klasse zu benennen.

- *Fabrikmethode*: Definition einer Schnittstelle mit Operationen zum Erzeugen eines Objekts, wobei die Unterklasse entscheidet, von welcher Klasse das zu erzeugende Objekt sein soll.

- *Singleton*: Sicherstellung, dass von einer Klasse genau eine Instanz existiert und Bereitstellung eines globalen Zugriffspunkt auf diese Instanz.

- *Brücke*: Entkopplung einer Abstraktion von ihrer Implementierung, so dass beide unabhängig voneinander verändert werden können. Ermöglicht es in C++ die Repräsentation einer Klasse vor ihren Klienten zu verstecken.

- *Kompositum*: Zusammenfassung von Objekten in Baumstruktur bei einheitlicher Behandlung der Objekte.

- *Besucher*: Kapselung einer auf den Elementen einer Objektstruktur auszuführenden Operation als Objekt. Erlaubt die Definition neuer Operationen, ohne die Klassen der Objektstruktur zu erweitern.

- *Beobachter*: Definition von 1-zu-n-Abhängigkeiten zwischen Objekten, so dass die Änderung des Zustands eines Objekts dazu führt, dass alle abhängigen Objekte benachrichtigt werden und ihren Zustand anpassen.

- *Strategie*: Definiert eine Familie von Algorithmen, kapselt jeden einzelnen und macht sie so zur Laufzeit austauschbar.

- *Iterator*: Sequentielles Durchlaufen der Elemente eines zusammengesetzten Objekts, ohne die zugrundeliegende Repräsentation offenzulegen.

- *Wrapper*: Anpassung der Schnittstelle eines vorgegebenen Objektes auf eine gewünschte Schnittstelle.

Im Folgenden werden exemplarisch einige der Entwurfskonzepte von BibRelEx ausführlicher beschrieben. Dabei wird zur Notation der Klassendiagramme die in [56] beschriebene *Unified Modeling Language* (UML) verwendet. Diejenigen UML-Symbole, die in der vorliegenden Arbeit verwendet werden, haben wir in Anhang E zusammengestellt.

6.1.2 Modellierung der bibliographischen Daten

In einer Bibliographie-Datenbank werden verschiedene Eintragstypen verwaltet. Für jeden Typ wird festgelegt, welche Felder ein Eintrag beinhalten muss und welche Felder optional angegeben werden können. BibTEX kennt beispielsweise den Typ *Article* für Zeitschriftenartikel, *Inproceedings* für Konferenzbeiträge, *Techreport* für technische Berichte, etc.

Bei der Entwicklung von BibRelEx wurden die von BibTEX vorgegebenen Eintragstypen gemäß Abbildung 6.1 modelliert. Bibliographische Einträge entsprechen den einzelnen Datensätzen zu einem Dokument und werden durch Objekte der Klasse BibEntry repräsentiert. Für jeden BibTEX-Eingabetyp wird eine entsprechende Klasse von BibEntry abgeleitet. Die Attribute entsprechen den von BibTEX bzw. GeomBib vorgesehenen Feldern für den jeweiligen Eingabetyp.

Damit inhaltliche Beziehungen und Annotationen genau wie bibliographische Einträge behandelt werden können, werden sie analog zu den bibliographischen Einträgen definiert.

BibRelEx soll für verschiedene Datenbasen eingesetzt werden können. Daher müssen sowohl die Menge der bibliographischen Eintragstypen als auch die einzelnen bibliographischen Eintragstypen selbst erweiterbar sein. Dazu werden benutzerdefinierbare Typen und die Festlegung der Pflichtfelder und optionalen Felder über die Konfigurationsdatei vorgesehen und so eine flexible Erzeugung von Dokumenten ermöglicht.

Die Feldzuordnung zu den bibliographischen Eintragstypen aus der Konfiguration muss bei der Erzeugung der entsprechenden Objekte in BibRelEx berücksichtigt werden. Dazu wird in BibRelEx eine parametrisierbare Fabrikmethode genutzt. Diese ermöglicht bei der Erzeugung der Objekte zwischen Varianten zu wählen. Dazu wird die Parameterliste für die Konstruktoren

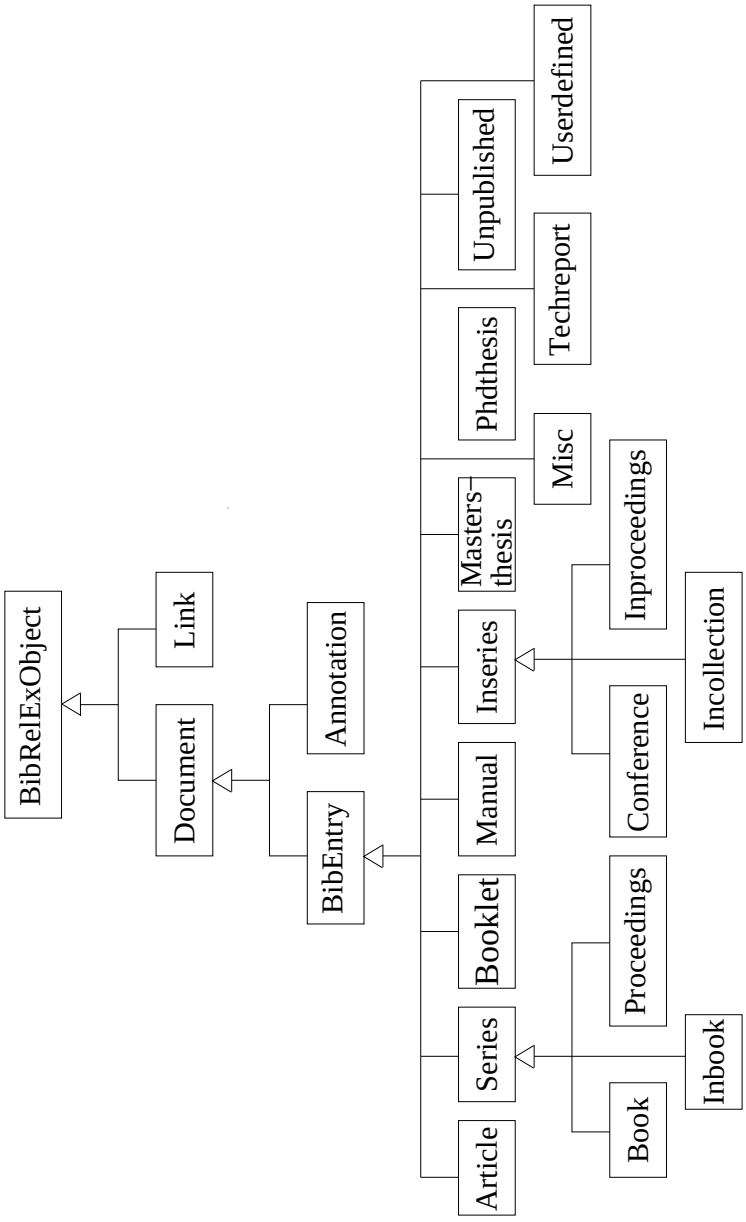


Abbildung 6.1: Datenmodell für bibliographische Daten

von BibRelEx-Objekten mit Hilfe einer Queue festgelegt. In der Queue steht was für ein Objekt zu erzeugen ist (*Annotation*, *Link*, *Article*, ...) und welche Felder mit welchem Inhalt zu belegen sind. Die Queue wird entlang der Klassenhierarchie der BibRelEx-Objekte weitergegeben, wobei die Konstruktoren der einzelnen Hierarchiestufen jeweils die für sie bestimmten Felder entfernen und zur Initialisierung verwenden. Nach Beendigung der Erzeugung eines BibRelEx-Objekts enthält somit die übergebene Queue nur noch solche Felder, die nicht zu dem erzeugten BibRelEx-Objekt-Typ gehören, z.B. ein *booktitle*-Feld nach der Erzeugung eines Artikels. So können falsch zugeordnete Felder einfach erkannt werden.

6.1.3 Inhaltliche Beziehungen und Annotationen

Eine wesentliche Entwurfsentscheidung war Annotationen und Links wie bibliographische Einträge zu definieren. Der Aufbau von Annotationen und Links für BibTEX-Datenbasen ist in Beispiel 6.1 angegeben. Damit stehen für Annotationen und Links dieselben Such- und Verwaltungsmöglichkeiten zur Verfügung wie für bibliographischen Einträge.

Beispiel 6.1 Aufbau von Annotationen und Links für BibTEX-Datenbasen

```
@annotation{a-gbubb-98
, author   =  "A. Author"
, year     =  1998
, month    =  jan
, title    =  "guter {B}eweis zu {B}ehauptung {B}"
, type     =  "Beweisskizze"
, contents =  "{I}n der vorliegenden {A}rbeit wird ein guter {B}eweis zu
               {B}ehauptung {B} gegeben, der im Folgenden skizziert
               werden soll: ..."
, keywords =  "..."
, update   =  "98.03 bibrelex"
}

@link{a-quelle#ziel-97
, author      =  "A. Author"
, year        =  1997
, month       =  dec
, type        =  annote
, update   =  "98.03 bibrelex"
}
```

Der Schlüssel von Annotationen wird analog zu den Schlüsseln bibliographischer Einträge aus den Angaben für Autoren, Titel und Jahr gebildet. Für

Links wird der Schlüssel aus den Angaben für Autoren und Titel und den Schlüsseln für Quelle und Ziel gebildet. Für Quelle und Ziel brauchen keine expliziten Felder vorgesehen werden, da sie einfach aus dem Link-Schlüssel zu extrahieren sind. Da es mehrere Links zwischen den selben BibRelEx-Objekten geben kann, muss der Schlüssel eines Links so ergänzt werden, dass er eindeutig ist. Linkschlüssel werden nur systemintern verwendet und sind nicht für den Benutzer relevant.

Da Quelle und Ziel von Links nicht beliebig gewählt werden können, führt die Bildung des Linkschlüssels aus Quelle und Ziel nicht zu unbeschränkt wachsenden Linkschlüsseln, wie wir im Folgenden zeigen werden. In BibRelEx können anhand der zulässigen Typen von Quelle und Ziel zwei Kategorien von Links unterschieden werden: Ein *BibLink* repräsentiert eine Beziehung zwischen zwei Dokumenten, wie etwa „cites“, „improves“, „generalizes“. Mit einem *AnnotationLink* wird eine Annotation an einem Dokument, einer Annotation oder einer Beziehung angebracht. Die Quelle eines Links kann damit immer durch einen Schlüssel, der ein Dokument bzw. eine Annotation eindeutig bezeichnet, spezifiziert werden. Ist das Ziel eines Links ein Dokument oder eine Annotation, so ist es ebenfalls eindeutig durch den zugehörigen Dokumentschlüssel[1] festgelegt. Damit enthält der Schlüssel eines BibLinks stets genau zwei Dokumentschlüssel. Ein AnnotationLink kann aber auch einen BibLink als Ziel haben. Da der Schlüssel eines BibLinks aber wie wir eben gesehen haben stets aus den Schlüsseln zweier Dokumente gebildet wird, enthält der Schlüssel des zugehörigen AnnotationLinks drei Dokumentschlüssel. Damit enthält jeder Linkschlüssel maximal drei Dokumentschlüssel und ist somit in seiner Länge beschränkt.

6.1.4 Anpassung an verschiedene Datenhaltungssysteme

Bei einem Datenvolumen von 3.2 MB, das zur Zeit mit GeomBib verwaltet wird, ist es nicht erforderlich, eine Datenbank zu verwenden. Man kann alle Daten im Hauptspeicher halten und so wirkungsvoll eine Echtzeit-Visualisierung unterstützen. Es wird aber eine Speicherung der Daten in einer relationalen Datenbank und ein Import von Daten aus Datenbanken vorgesehen, um die Übertragbarkeit des Ansatzes auf andere bibliographische Datenbasen zu untersuchen.

Die Anbindung der verschiedenen Datenhaltungssysteme erfolgt durch die Bildung eines Repräsentanten für den Datenbestand („Hüll-Objekt“, Wrap-

[1] Da Annotationsschlüssel wie Dokumentschlüssel gebildet werden, unterscheiden wir diese im Folgenden nicht mehr explizit.

per), der die auszuführenden Operationen an die für das jeweilig verwendete Datenhaltungsystem zuständige Komponente (DBHandler, Filesystem) delegiert, siehe Abbildung 6.2. Der *Wrapper* kapselt die Spezifika der verschiedenen Datenhaltungssysteme, damit diese über eine einheitliche Schnittstelle genutzt werden können.

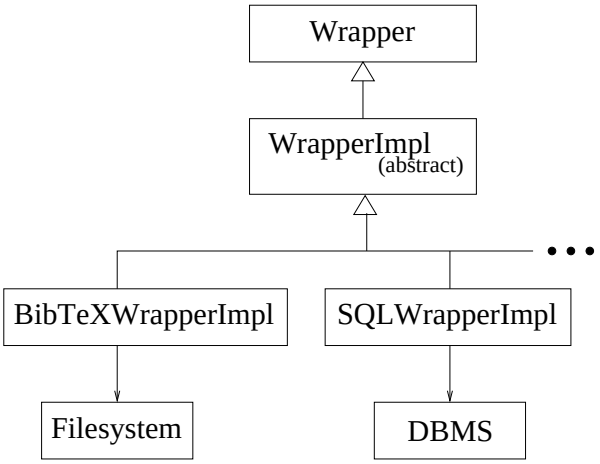


Abbildung 6.2: Anbindung unterschiedlicher Datenhaltungssysteme mit Hilfe eines Wrappers

Der Wrapper wird in BibRelEx auch zur Unterstützung der Aktualisierung von GeomBib verwendet, um die drei BibTeX-Dateien für den globalen, update und lokalen Datenbestand zu kapseln.

Die Auswahl eines konkreten Datenhaltungssystems zur Laufzeit des Programms erfolgt über die Angabe einer Kennzeichnung zur Festlegung des Datenbanktyps (*Vendor*), z.B. BibTEX oder SQL und eines Modells (*Model*), das angibt, ob es sich um einen einzelnen Datenbereich handelt (Modell Single) oder um ein Tripel aus globalen, update und lokalen Datenbereich (Modell Set) handelt. Abbildung 6.3 zeigt die Umsetzung des Wrapper-Konzepts für den Vendor BibTEX und die Modelle Set/Single.

Für das Modell Set werden zwei Index-Dateien verwendet. Die erste enthält den Index für den globalen Teil der Datenbasis und bleibt damit bis zu dem nächsten Aktualisieren der Datenbank unverändert. Die zweite enthält den Index für alle update- und lokalen Einträge. Dies hat den Vorteil, dass bei Änderungen durch mehrere Benutzer nur die zweite Indexdatei neu gebildet werden muss. Sie ist im Allgemeinen erheblich kleiner als die erste, da es meist deutlich weniger Änderungen und Ergänzungen gibt als Originaleinträge. Geombib enthält beispielsweise über 13000 Einträge, wobei die Update-Wünsche aller Benutzer i.A. nur 100-200 Einträge umfassen. Da-

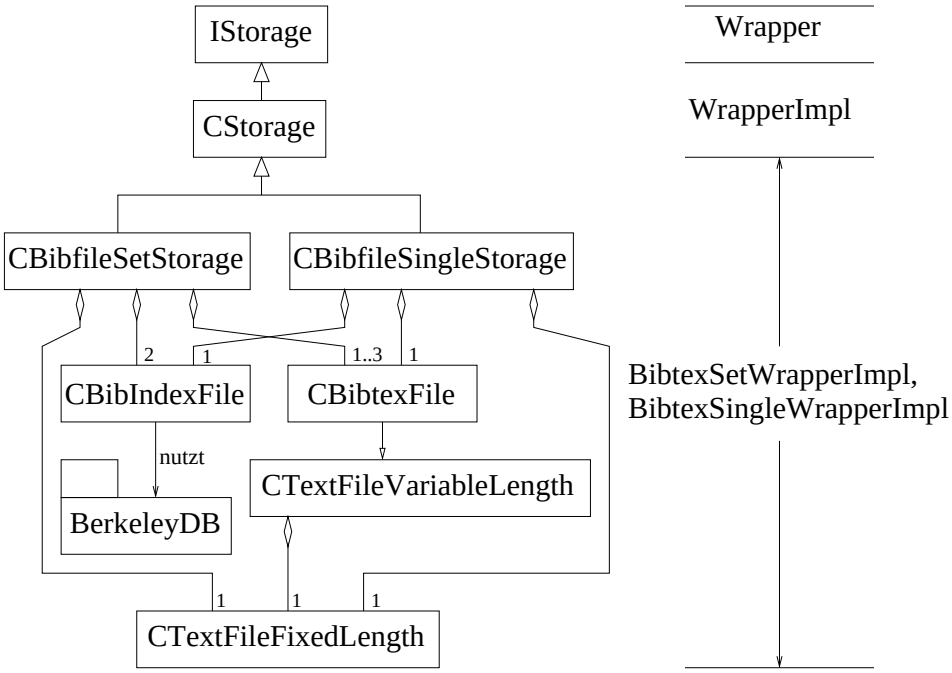


Abbildung 6.3: Klassenhierarchie BibTEX-Vendor

mit wird eine erhebliche Zeitersparnis beim Indizieren gewonnen, zumal sich gerade das Schreiben der Indexeinträge in die Datenbank als Flaschenhals erwiesen hat.

Die Verwendung eines Wrappers hat weiterhin den Vorteil, dass BibRelEx mit verschiedenen Datenbasen gleichzeitig arbeiten kann. So ist es z.B. möglich, Einträge aus einer BibTEX-Datenbasis in eine SQL-Datenbasis zu kopieren. Für den Benutzer bleibt dieser Vorgang völlig transparent. Er kann einfach mittels Drag&Drop den Eintrag aus dem Hauptfenster der einen Datenbasis in die andere ziehen und muss sich nicht um die Art der jeweils verwendeten Datenbasis kümmern.

Mit Hilfe des Besucher-Musters können datenbankabhängige Operationen für neue Datenhaltunssysteme ergänzt werden, ohne die Klassen für BibRelEx-Objekte ändern zu müssen. Auf diese Weise ist in BibRelEx beispielsweise die Zerlegung einzelner Felder in Token realisiert worden.

6.1.5 Effizienz durch Verwendung des Beobachter-Entwurfsmusters

Wie bereits in Abschnitt 6.1.3 beschrieben, werden in BibRelEx Annotationen und Beziehungen (Links) als eigenständige Objekte verwaltet, um bei

der Recherche und Verwaltung einen effizienten Zugriff auf sie zu haben. Die Zuordnung erfolgt über den Schlüssel eines Objekts (Dokument, Annotation, Beziehung). Ändert sich dieser oder wird ein Objekt gelöscht, müssen alle Verweise von und zu diesem Objekt aktualisiert werden. Die Aktualisierung kann wiederum dazu führen, dass weitere Objekte angepasst werden müssen.

Mit Hilfe des *Beobachter*-Entwurfsmuster können solche 1-zu-n-Abhängigkeiten zwischen Objekten definiert werden. Wenn ein Objekt seinen Zustand ändert, werden alle abhängigen Objekte darüber informiert und können sich entsprechend aktualisieren.

Die in BibRelEx verwendete Bibliothek Qt bietet das sehr mächtige Signal/Slot-Konzept. Mit Signalen kann man eine Nachricht aus einem Objekt heraus absenden, und Slots dienen dazu, solche Nachrichten zu empfangen. Über die Methode *connect* kann man ein Signal mit einem Slot zur Laufzeit verbinden. Ein Signal kann mit einer beliebigen Anzahl von Slots verbunden sein. Ebenso kann ein Slot Nachrichten von mehreren Signalen von verschiedenen Objekten empfangen. So können n-zu-m-Beziehungen definiert werden. Insbesondere können damit 1-zu-n-Beziehungen definiert werden, womit das Signal/Slot-Konzept das Beobachter-Entwurfsmuster beinhaltet.

In einer BibRelEx-Sitzung stehen dem Benutzer eine Vielzahl von Objekten zur Verfügung, aber nur wenige davon werden in einer Sitzung geändert. Daher werden in BibRelEx zur weiteren Effizienzsteigerung Verbindungen erst vor der Anforderung von Änderungen hergestellt.

Eine Besonderheit in BibRelEx ist das Konzept der dreistufigen Überdeckung bei Datenbasen nach dem Modell Set, vgl. Abschnitt 6.1.4. Zusätzlich zu den Abhängigkeiten der Objekte bei Schlüsseländerungen muss hier auch berücksichtigt werden, dass Änderungen an den Einträgen zu Änderungen in der Sichtbarkeit führen können. Beispielsweise muss sichergestellt werden, dass alle Verweise von und zu einem Objekt, das nur im lokalen Teil der Datenbasis existiert, ebenfalls im lokalen Teil der Datenbasis liegen müssen. Wird beispielsweise für ein Dokument, zu dem es einen lokalen und einen globalen Eintrag gibt, der globale Eintrag zur Löschung markiert, so müssen auch alle Verweise von und zu diesem Dokument zur Löschung markiert werden und ggf. in den lokalen Teil der Datenbasis kopiert werden, da sonst der öffentliche Datenbestand Verweise, die ins „Leere“ zeigen, enthielte und damit nicht mehr konsistent wäre.

Die Konsistenz bzgl. dieser Sichtbarkeitsregeln wird in BibRelEx ebenfalls mit Hilfe des Beobachter-Entwurfsmuster sichergestellt. Ein Objekt, dessen Sichtbarkeit sich ändert, sendet dazu an alle mit ihm in Verbindung stehenden Objekte eine entsprechende Nachricht.

Das Beobachter-Muster bzw. das Signal/Slot-Konzept kann darüber hinaus dazu genutzt werden, um die Darstellungsaspekte der Benutzungsschnitt-

stelle von den dahinterliegenden Anwendungsdaten zu trennen. So sendet in BibRelEx ein Objekt, dessen Zustand sich geändert hat, eine Nachricht an alle Fenster der Benutzungsschnittstelle, die eine Darstellung dieses Objekt enthalten, um eine Aktualisierung der Sichten zu veranlassen.

Neben den bisher betrachteten Entwurfskonzepten wurde eine Reihe weiterer Konzepte in BibRelEx verwendet, die hier aus Platzgründen nicht weiter ausgeführt werden können. Hierzu zählen unter anderem generische Container, intelligente Zeiger (Smart Pointer) und Referenzzählung zur Speichererspar-nis durch mehrfache Verwendung von Objektinstanzen.

6.2 Konzepte zur Unterstützung der Benutzungsfreundlichkeit

BibRelEx besitzt ein hypertextähnliches Hilfesystem mit kontextabhängiger Funktionalität. Zusätzlich bietet BibRelEx *Tooltips* und *What's-This-Hilfe* als Hilfestellung an. Tooltips sind kleine Hilfswörter, die in einem Fenster erscheinen, sobald man mit der Maus etwas länger über einem Bedienelement verweilt. Die What's-This-Hilfe bietet etwas umfangreichere Texte und wird über einen Button in der Hilfeleiste aktiviert.

Alle Einstellungen innerhalb von BibRelEx werden in Konfigurationsdateien gespeichert, so dass sie für den Benutzer beim nächsten Start des Programms wiederhergestellt werden und können über Bildschirmdialoge eingestellt werden. BibRelEx verwendet zwei Konfigurationsdateien. Die erste enthält die Standardkonfiguration und ist global für alle Benutzer. Die zweite Datei ist die lokale Konfigurationsdatei des jeweiligen Benutzers. Ein Benutzer kann verschiedene lokale Konfigurationsdateien besitzen, die er dem jeweiligen zu bearbeitenden Problem anpassen kann und die er während der Laufzeit austauschen kann. Beispielsweise erfordert das Bearbeiten einer Lehrveranstaltung, siehe Nutzungsbeispiel 4.1 *Organisation eines Seminars*, ganz andere Linktypen als die Bearbeitung einer Begutachtung oder die Literaturverwaltung der eigenen wissenschaftlichen Arbeit (Abschnitte 4.3 und 4.4). Mit Hilfe verschiedener Konfigurationsdateien lassen sich leicht die Linktypen dem jeweiligen Anwendungsbereich anpassen. Informationen über den Aufbau der Konfigurationsdateien und ihre Verwendung findet sich im Anhang C.

Alle Dialoge zwischen BibRelEx und den Nutzern finden derzeit in Englisch statt. Beim Entwurf von BibRelEx wurde aber auf weitgehende Sprach-

unabhängigkeit geachtet, so dass mit wenig Aufwand in der Benutzungsoberfläche optional weitere Sprachen unterstützt werden können. Dabei war von erheblichem Vorteil, dass die für die Implementierung der Benutzungsoberfläche verwendete Bibliothek Qt bereits Konzepte für mehrsprachige Anwendungen unterstützt. Für die Visualisierung ist die Anpassung aufwendiger, da LEDA keine entsprechenden Konzepte kennt. Für die Texte in der Visualisierungsoberfläche kann man eine Sprachunabhängigkeit dadurch erreichen, dass man die Texte für die verschiedenen Sprachen in der Standardkonfigurationsdatei definiert und diese dann zur Laufzeit entsprechend der gewählten Sprache einliest.

Für die Konsistenzprüfung wird der englischsprachige Soundex-Code verwendet (Abschnitt 5.1). Auch hier kann man Sprachunabhängigkeit erreichen, indem man durch Anwendung des Strategiemusters den Algorithmus zur Laufzeit austauschbar hält.

6.3 Struktur der Software

Das Design der Software ist in den Abbildungen 6.4 bis 6.9 in UML-Notation (Unified Modelling Language) [56] angegeben. Abbildung 6.4 zeigt die Teilsysteme der Software. Die graphische Benutzungsoberfläche GUI von BibRelEx wurde mit Hilfe der C++ Klassenbibliothek Qt der norwegischen Firma Trolltech [113] implementiert. Die Klasse QObjekt dieser Bibliothek wird im Teilsystem BibManage, das die gesamte Datenhaltung und -verwaltung umfasst, verwendet, um das in Abschnitt 6.1.5 beschriebene Signal/Slot-Konzept zur Realisierung des Beobachtermusters zu nutzen. Damit kann innerhalb von BibManage die Konsistenz zwischen miteinander in Beziehung stehenden Objekten sichergestellt werden. Zusätzlich ermöglicht das Signal/Slot-Konzept, die Darstellungsaspekte der Benutzungsoberfläche von den dahinterliegenden Daten der BibManage-Objekte zu trennen. Bei der Implementierung von BibManage wurden darüber hinaus die Bibliothek BerkeleyDB [177] zur effizienten Indizierung der Datenbasis verwendet. BibManageC stellt ein Kommandozeileninterface zu BibManage zur Verfügung. Die Visualisierungskomponente Visual benutzt die Bibliotheken LEDA (Library of Efficient Data types and Algorithms) [3, 130] und AGD (Algorithms for Graph Drawing) [128] zur Realisierung der Visualisierung. Zusätzlich wurde der Layout-Algorithmus von GEM3D [28] weitestgehend übernommen.

Die Software ist in der Programmiersprache C++ [185] auf einer UNIX/X-Windows Plattform implementiert und getestet worden. Die derzeitige Implementierung des Prototypen umfasst ca. 52000 Zeilen Code verteilt auf 145 Klassen. Zum Test wurden verschiedene BibTEX Bibliographien ver-

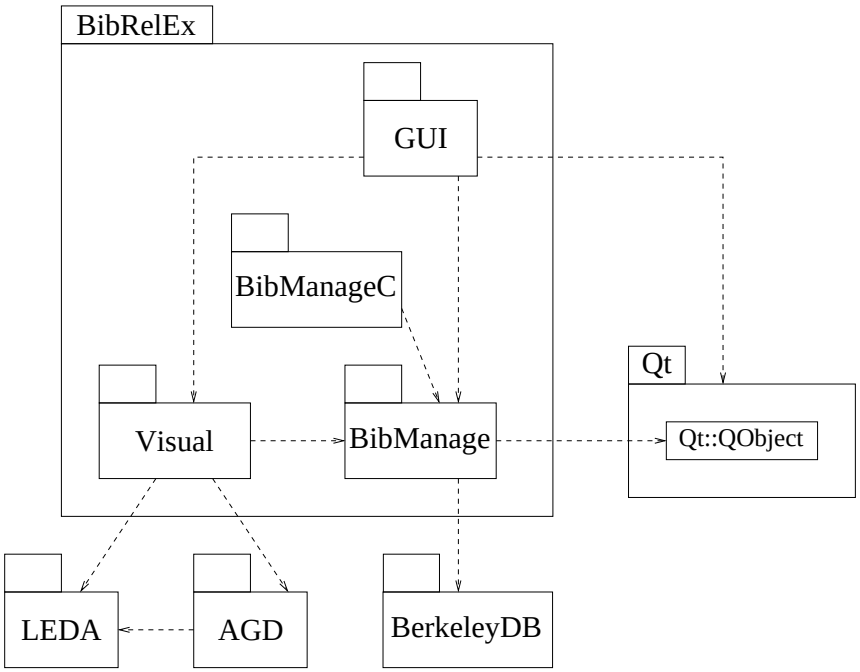


Abbildung 6.4: Die Teilsysteme von BibRelEx

wendet, wobei GeomBib mit ca. 14000 Bibliographie-Referenzen die größte Sammlung darstellt.

Da die Systemumgebung (Hardware, Betriebssystem, Netzwerk) auf der Nutzerseite sehr heterogen ist, wurde der Programmcode für weitere Plattformen möglichst portabel gehalten. Wir haben uns dabei nach eingehenden Tests für die Verwendung von C++ entschieden. Alle verwendeten Bibliotheken stehen auch für Windows zur Verfügung.

Java wurde trotz seiner Portabilität nicht gewählt, weil die Ablaufgeschwindigkeit von Java-Programmen nicht mit der von Qt-Programmen mithalten kann. Außerdem hatte Java3D zumindest noch zu Beginn der Implementierung der Visualisierungskomponente von BibRelEx im grafischen Bereich Geschwindigkeitsprobleme und es fehlten noch viele für unser Projekt nützliche Funktionen, so dass eher mit einer längeren Entwicklungszeit zu rechnen war.

6.4 Die Datenhaltungskomponente

Die gesamte Datenhaltung und -verwaltung erfolgt im Teilsystem BibManage, siehe Abbildung 6.5. Der Zugriff auf dieses Teilsystem erfolgt über die Klasse BibManageObj. Diese Klasse ist als Singleton realisiert, d.h. es ist

sicher gestellt, dass von dieser Klasse genau ein Objekt existiert und es wird ein globaler Zugriffspunkt darauf bereitgestellt.

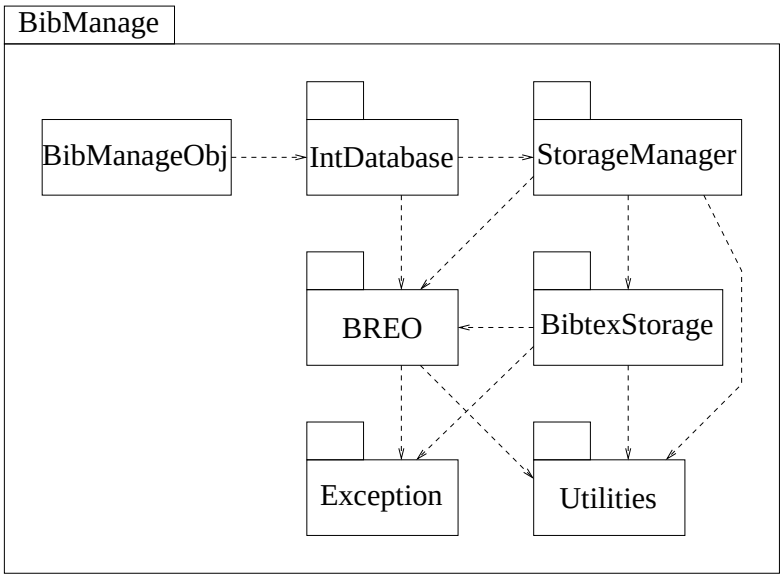


Abbildung 6.5: Die Teilsysteme von BibManage

Das Teilsystem Utilities enthält einige für das gesamte System nützliche Komponenten, wie beispielsweise die Implementierung der Konfigurationsdateien oder Templates für Singletons, Referenzzählung und Smart Pointer. Die Ausnahmebehandlung ist im Teilsystem Exception angesiedelt.

StorageManager enthält die Implementierung des Wrapperinterfaces (Abschnitt 6.1.4) und BibtexStorage die Implementierung des Wrappers für BibTEX-Datenbasen. Der Wrapper ist so implementiert worden, dass er zunächst nur mit Metadaten arbeiten kann, d.h. zu einem Eintrag wird zunächst nur die Schlüsselinformation und seine Position in der Datenbasis eingelesen, nicht aber der Eintrag selbst geparst. Diese Informationen wird zu jedem Teil der Datenbasis in einer eigenen Datei abgelegt und kann so effizient verarbeitet werden. Auch zum textbasierten Suchen müssen die Einträge selbst nicht eingelesen werden, da alle notwendigen Informationen in der Indexdatei stehen und damit ebenfalls effizient auf sie zugegriffen werden kann. Erst wenn die Daten eines Eintrags wirklich gebraucht werden, z.B. zur Anzeige in einem Editorfenster, werden diese explizit geladen.

Das Teilsystem BREO [107] stellt Objekte für die bibliographischen Daten (Abschnitt 6.1.2), Annotationen und Links (Abschnitt 6.1.3) bereit. Da dieses Teilsystem der Kern der Datenhaltung ist, wurde um eine Verringerung von Übersetzungs-Abhängigkeiten zu erreichen, das Pimpl-Idiom [188]

verwendet. Abbildung 6.1 zeigte die Klassenhierarchie für die Implementierung von BREO. An der Schnittstelle ist diese Feinunterteilung der Hierarchie nicht sichtbar. Dort wird nur zwischen bibliographischen Einträgen, Annotationen und Links unterschieden, siehe Abbildung 6.6.

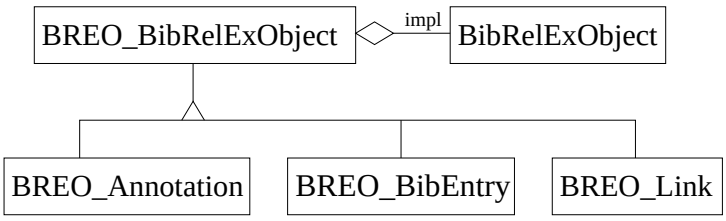


Abbildung 6.6: Die Exportschnittstelle des Teilsystems BREO

IntDatabase stellt die Schnittstelle zu den einzelnen Datenbasen her und ist für die Bearbeitung von Anfragen zuständig. In Abschnitt 5.5 haben wir bereits darauf hingewiesen, dass für den Benutzer das Ergebnis einer Suche wie die ursprüngliche Datenbasis als eine Menge von bibliographischen Einträgen, Annotationen und Links erscheint und er so auf beide dieselben Operationen anwenden kann. Dies wurde dadurch realisiert, dass sowohl Datenbasen als auch Ergebnismengen als Behälter vom selben Typ definiert wurden. Der Zugriff auf beide kann dadurch transparent über einen Iterator auf diesem Behälter erfolgen.

Abbildung 6.7 zeigt den zugehörigen Ausschnitt aus der Klassenhierarchie der internen Datenbasis. Um eine effektive Trennung zwischen der Schnittstelle und der Implementierung der einzelnen Klassen der internen Datenbasis zu erreichen, werden abstrakte Klassen (Interfaces) verwendet. Ein Storage dient dazu, die formatspezifischen Datenbankoperationen zu kapseln, siehe Abschnitt 6.1.4. Datenbasen (Database) bzw. Ergebnismengen (Resultset) enthalten die Einträge im internen Format (als BibEntryTriple).

Zur Realisierung des in Abschnitt 5.2 beschriebenen Überdeckungskonzept zur Unterstützung des periodischen Updates, besteht jeder Eintrag der Datenbasis aus bis zu drei BibRelEx-Objekten für die globale, update und lokale Version eines bibliographischen Eintrags.

BibManage vergibt eindeutige Nummern (Handle), um die einzelnen Datenbasen bzw. Ergebnismengen von Anfragen ansprechen zu können. Beispiel 6.2 zeigt die einfache Verwendung der Datenhaltungskomponente. In ihm wird eine Datenbasis geöffnet und nach einem Begriff im Titel gesucht. Anschließend wird die Ergebnismenge mit Hilfe eines Iterators durchlaufen und jeweils Autor, Titel und Jahr ausgegeben. Erst beim Zugriff auf die einzelnen Felder für die Ausgabe müssen die betreffenden Datensätze einmal

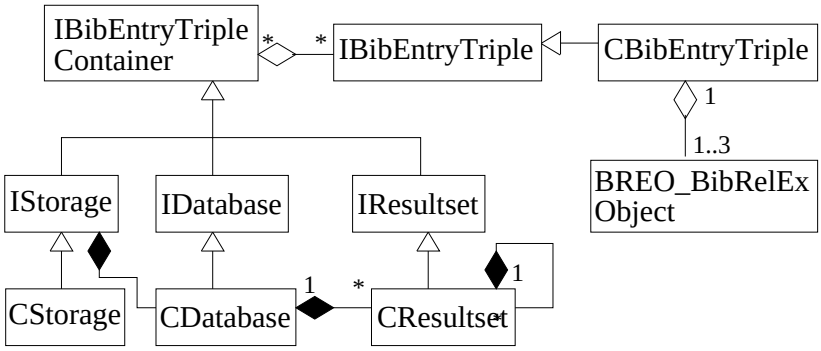


Abbildung 6.7: Klassenhierarchie der internen Datenbasis

geladen werden. Das erfolgt wiederum transparent für den Benutzer in der zugehörigen Wrapperimplementierung.

Beispiel 6.2 Verwendung des Teilsystems BibManage

```
...
#include "BibManage.h"
...
using namespace BibManageStuff;
...
eModel model = ModelFileSet;
string vendor = "BibTeX";
string storageName = "geom";

pIDatabase db =
   BibManageObj().wsp()->OpenDatabase( model,vendor,storageName );
string query = "ti voronoi";
unsigned handle = BibManageObj().query( m_acthandle, query );
pIBibEntryTripleIterator it = BibManageObj().getIter( handle );
for ( it->MoveFirst(); !it->IsLast(); ++(*it) )
{
   pIBibEntryTriple entry = **it;
   cout << "\n" << entry->GetField( "author" )
        << " "  << entry->GetField( "title" )
        << " "  << entry->GetField( "year" );
}
...
```

6.5 Die Visualisierungskomponente

Abbildung 6.8 zeigt einen Ausschnitt der Klassenhierarchie der Visualisierungskomponente von BibRelEx. Die Kommunikationsschnittstelle (Klasse VIS_ViToBi) zu der LEDA-Visualisierungsumgebung ist wie BibManageObj als Singleton realisiert. Sie ist mit Hilfe des Brückenmusters implementiert. Das hat zum einen den Vorteil, dass die Implementierung von der abstrakten Schnittstellenklasse entkoppelt ist. Zum anderen lassen sich so bei der Übersetzung Konflikte zwischen den unterschiedlichen eingesetzten Bibliotheken vermeiden.

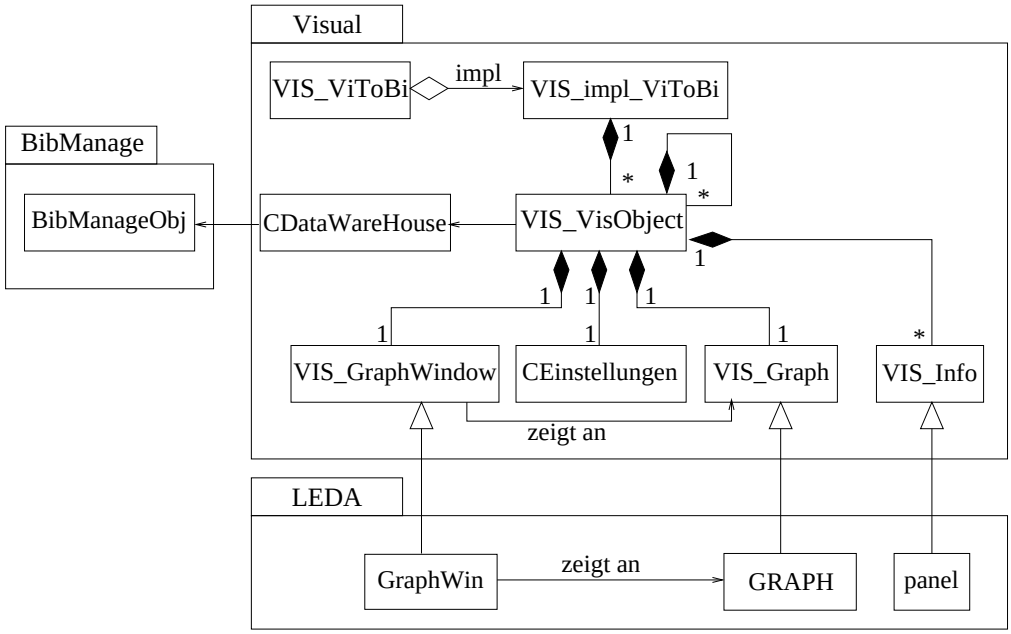


Abbildung 6.8: Ausschnitt aus der Klassenhierarchie der Visualisierungskomponente

Beim Öffnen eines Visualisierungsfensters wird der von BibManage eindeutig vergebene Handle der zugrundeliegenden Datenbasis bzw. Ergebnismenge übergeben. Über diesen kann auf alle Daten innerhalb dieser Datenbasis bzw. Ergebnismenge zugegriffen werden und zwar nur auf diese. Sämtliche Zugriffe auf die Datenbasis sind in der Klasse CDataWareHouse gekapselt. Diese enthält auch einige Hilfsstrukturen für einen effizienten Zugriff auf das Beziehungsgeflecht.

Die Visualisierungsumgebung kann beliebig viele Visualisierungsobjekte (Klasse VIS_VisObject) verwalten. Ein Visualisierungsobjekt ist eine Komposition aus einem Visualisierungsfenster (VIS_GraphWindow), einem Gra-

phen (Klasse VIS_Graph) und den für dieses Objekt relevanten Einstellungen (Klasse CEinstellungen). Darüber hinaus kann ein Visualisierungsobjekt beliebig viele Informationsfenster verwalten und weitere Visualisierungsobjekte erzeugen. Beispielsweise können zu einem Visualisierungsobjekt mehrere Cluster-Fenster gehören. Alle Kindfenster eines Visualisierungsobjekt werden bei seiner Beendigung ebenfalls zerstört. Ebenso werden alle Visualisierungsobjekte zerstört, wenn die Visualisierungsumgebung selbst beendet wird.

Die Visualisierungfenster (Klasse VIS_GraphWindow) sind von der LEDA-Fensterklasse GraphWin (siehe Kapitel 12 in [130]) abgeleitet und fügen noch speziell für BibRelEx entworfene Navigationsfunktionen wie das Zoomen ausschließlich in eine Richtung, wie es in der zeitlichen Darstellung benötigt wird, oder die automatische Anzeige von Hinweisfenstern für Dokumentschlüssel, wenn die Maus längere Zeit auf einem Graphknoten verweilt, hinzu.

Die Klasse VIS_Graph kapselt alle Zugriffe auf den zugrundeliegenden parametrisierten LEDA-Graphen und stellt alle Funktionen zur Bearbeitung des zum Beziehungsgeflecht gehörenden Graphen zur Verfügung. Hierzu gehören unter anderem die verschiedenen Layout-Algorithmen und Clusterfunktionen.

LEDA verlangt, dass Menü- und Ereignisfunktionen global sind. Diese bekommen beim Aufruf eine Referenz auf das zugehörige LEDA-GraphWin übergeben. Um in den Menü- und Ereignisfunktionen auf die Methoden des betroffenen Visualisierungsobjektes zugreifen zu können, verwaltet die Visualisierungsumgebung eine Liste, die zu jedem GraphWin einen Zeiger auf das Visualisierungsobjekt bereithält, zu dem dieses GraphWin gehört.

Eine wichtige Aufgabe der Visualisierungsumgebung ist die Ereignisverarbeitung (Mausklicks etc.). Die Ereignisse werden in einer Schleife abgearbeitet, in der das jeweils eingehende Ereignis an dasjenige Fenster zur Verarbeitung weitergeleitet wird, in dem es aufgetreten ist. Diese Schleife wird erst verlassen, wenn irgendein eventuell auch schon früher geöffnetes Visualisierungsfenster geschlossen oder dessen Done-Button gedrückt wurde. Erst dann ist die Qt-Benutzungsoberfläche wieder bedienbar. Die LEDA-Umgebung ist allerdings dann so lange blockiert bis die Kontrolle wieder explizit durch Benutzeranforderung an die Visualisierungsumgebung übergeben wird. Bedingt durch dieses Vorgehen bei der Ereignisverarbeitung in der Visualisierungsumgebung blockieren sich die Qt-Benutzungsoberfläche und die LEDA-Visualisierungsumgebung gegenseitig.

Die gegenseitige Blockierung von Qt-Benutzungsoberfläche und der LEDA-Visualisierungsumgebung hat Auswirkungen auf die Dynamik des Systems. So kann das Signal/Slot-Konzept von Qt, mit dem das Beobachter-Muster leicht umgesetzt werden konnte, nicht genutzt werden, um die jewei-

ligen Visualisierungfenster darüber zu informieren, dass sich die dargestellten Daten geändert haben. Die einzige Möglichkeit auf Änderungen am Datenbestand zu reagieren, besteht darin, dass die Visualisierungsumgebung nachdem sie die Kontrolle zurück erhalten hat, überprüft, ob alle Daten noch gültig sind und ggf. die Darstellung neu aufbaut.

Eine Möglichkeit, die gegenseitige Blockierung aufzuheben, besteht darin, die Bearbeitung der Benutzeroberfläche und der Visualisierung in zwei Threads aufzuspalten. Die für die Benutzungsoberfläche verwendete Qt-Bibliothek (s. nächster Abschnitt) stellt Threads zur Verfügung. Allerdings sind alle GUI Klassen der Qt-Bibliothek nicht threadsicher[2], so dass diese nicht in Threads verwendet werden können.

Wir haben zusätzlich die Realisierung der Visualisierung in einem Thread mit Hilfe des JThread-Pakets [117] implementiert. Dieses Paket liefert Klassen mit deren Hilfe Threads auf verschiedenen Plattformen einfach genutzt werden können. Die Klassen sind zur Zeit einfache Wrapper um vorhandene Thread-Implementierungen. Damit war ein blockierungsfreies Arbeiten der Qt- und LEDA-Oberflächen möglich. Leider kam es (unter Linux) immer wieder zu Programmabstürzen durch unerwartete asynchrone Xlib-Antworten. Ursachenforschung hat ergeben, dass auch die Xlib-Bibliothek aufgrund von Bugs nicht threadsicher ist. Das Gleiche gilt auch für einige andere auf dem X Window-System aufbauende Software, wie beispielsweise OSF/Motif. Unter Windows sieht die Sache besser aus, aber auch hier sind beispielsweise die Treiber der Hardware-OpenGL-Beschleunigung nicht immer threadsicher programmiert. Damit würde eine Implementierung mit mehreren Threads auch die Portabilität von BibRelEx stark einschränken. Insgesamt haben wir uns daher für die „blockierende“ Lösung entschieden, da ein einfaches Umschalten zwischen den beiden Umgebungen mit Hilfe des „Visualization“-Buttons dem Benutzer eher zuzumuten ist, als die Abstürze durch Probleme mit der Xlib-Bibliothek und der Verlust der Portabilität.

6.6 Die graphische Benutzungsoberfläche

Für die Implementierung der graphischen Benutzungsoberfläche von BibRelEx wurde die von der norwegischen Firma Trolltech entwickelte C++ Klassenbibliothek Qt verwendet. Sie ermöglicht eine einfache und portable GUI-Programmierung. Mit Qt entwickelte Programme sind sofort ohne

[2]Eine Funktion oder Programmbibliothek ist threadsicher programmiert, wenn mehrere Threads damit arbeiten können ohne dass sie sich gegenseitig oder die Arbeit der Funktion negativ beeinflussen. Diese Beeinflussung findet meist über interne Datenbereiche statt, auf die immer nur ein Thread gleichzeitig Zugriff haben sollte/darf.

zusätzlichen Portierungsaufwand sowohl unter allen Linux/Unix- wie auch unter allen Windows-Systemen lauffähig. Sie müssen lediglich mit den entsprechenden Compilern (Visual C++ oder Borland C++ unter Windows-Systemen oder dem auf dem jeweiligen Linux/Unix-System angebotenen cc-Compiler) kompiliert und gelinkt werden.

Die Implementierung der Benutzungsoberfläche umfasst im wesentlichen die Anzeigefenster für Datenbasen, Suchergebnissen, bibliographischen Einträgen und Inkonsistenzen, sowie verschiedene Dialoge zur Einstellung von Optionen, Eingabe von Suchanfragen, Auswahl von Dateien, Fortschrittsanzeige, etc. Abbildung 6.9 zeigt einen Auschnitt der Klassenhierarchie der zentralen Anzeigefenster. Ähnlich wie in der internen Datenbasis werden Datenbasen und Ergebnismengen wieder gleich behandelt. Ein Fenster vom Typ GUI_Database kann entweder eine Datenbasis oder eine Ergebnismenge anzeigen. Die Unterschiede liegen in den angebotenen Menüfunktionen. Diese Fenster können beliebig viele Unterfenster besitzen, z.B. Fenster zur Darstellung verschiedener Suchergebnisse und Editierfenster für Einträge der Datenbasis (GUI_ViewTriple). Für Datenbasen vom Modell Set, siehe Seite 6.1.4 werden der globale, update und lokale Eintrag als Karteikarten dargestellt, wobei die sichtbare Überdeckung zuoberst angezeigt wird. Zur Aktualisierung von abhängigen Sichten wird das Signal/Slot-Konzept von Qt eingesetzt.

Auf weitere Implementierungsdetails der Benutzungsoberfläche soll hier nicht näher eingegangen werden. Das ist Aufgabe der Dokumentation. Vielmehr wollen wir diesen Abschnitt mit einer kurzen Aufzählung weiterer implementierter Funktionalität der Benutzungsoberfläche beenden:

- Hypertext-Hilfesystem,
- Drag&Drop-Mechanismus,
- Laden von Dokumenten direkt via FTP und WWW,
- einfaches Navigieren innerhalb der Datenbasis durch Verfolgen von Links,
- Drucken von Hilfeseiten und verschiedener Darstellungen der Bibliographien und Suchergebnisse.

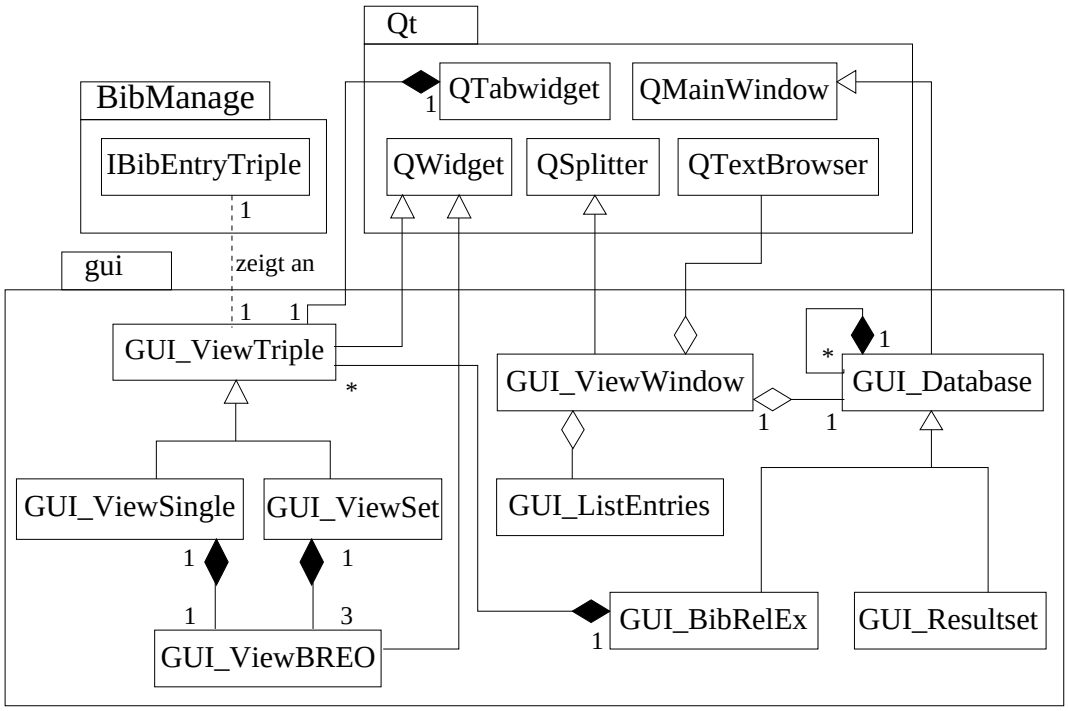


Abbildung 6.9: Ausschnitt aus der Klassenhierarchie der Benutzungsoberfläche (GUI)

Einen Einblick in die Gestaltung der graphischen Benutzungsoberfläche gibt Anhang A.

Kapitel 7

Evaluierung

Nachdem wir im letzten Kapitel Entwurf und Implementierung des Prototypen von BibRelEx beschrieben haben, wenden wir uns nun der Erprobung der entwickelten Konzepte zu. Obwohl eine Vielzahl an prototypischen Systemen und zahlreiche Forschungsprojekte im Bereich der Informationsvisualisierung existieren, finden sich nur wenige in kommerziellen Systemen oder realen Anwendungen wieder. Eine Ursache dafür ist, dass bei vielen Systemen der Schwerpunkt auf die Anwendung neuer Techniken gelegt wird, die Entwicklung aber an den Bedürfnissen der Benutzer vorbei geht. Die wenigsten Systeme werden einer Evaluation durch potentielle Benutzergruppen unterzogen. Daher ist uns eine informelle Evaluierung in realen Anwendungen wichtig, um die Tragfähigkeit des Ansatzes, die Einsetzbarkeit der entwickelten Konzepte und die Akzeptanz beim Benutzer zu überprüfen.

Wir haben BibRelEx dazu von verschiedenen Anwendern in verschiedenen Anwendungen testen lassen. So wurde BibRelEx zur Vorbereitung eines Seminars, zur Verwaltung der Literatur für Veröffentlichungen in einer Arbeitsgruppe, zur Recherche in GeomBib und zur Prüfungsvorbereitung eingesetzt. Um den Benutzer dabei nicht in den vielfältigen Einsatzmöglichkeiten zu beschränken, haben wir keinen Testplan vorgegeben. Im Anschluss daran haben alle Testbenutzer das System in einem freien Feedback beurteilt. Auf eine formellere Evaluierung und Auswertung haben wir erstmal verzichtet, da uns dies sowohl aufgrund der Gruppengröße und der verschiedenen Anwendungen nicht sinnvoll erschien.

Die allgemeine Resonanz war sehr positiv. Das Arbeiten mit der graphischen Benutzungsoberfläche wurde als anschaulich und intuitiv empfunden. Als komfortabel wurde die Drag&Drop Funktionalität bezeichnet, die vor allem die Eingabe von Beziehungen, insbesondere der Zitierrelation vereinfacht. Ebenso wurde die Hypertextfunktionalität in den Eingabemasken, die ein einfaches Navigieren zwischen den Einträgen ermöglicht, positiv hervor-

gehoben. Durch den zusätzlichen Inkonsistenztest half BibRelEx korrekte Einträge zu erstellen und Doppeleinträge zu vermeiden. BibRelEx wurde als hilfreich bei der Organistation des eigenen Literaturbestandes beurteilt.

In allen erprobten Anwendungsszenarien waren die angebotenen Möglichkeiten der Wissensaggregation geeignet, die Datenbasis mit zusätzlichem Wissen anzureichern und so zu strukturieren, dass die verschiedenen Informationsbedürfnisse der Benutzer gedeckt wurden.

Die Visualisierungen wurden sowohl für das Verständnis des Informationsraumes als auch bei der Recherche als hilfreich angesehen. Zusammenhänge wurden schnell deutlich und auch die Verlaufsdarstellung über die Jahre half zusätzlich beim Verständnis des Informationsraumes.

Bei der Benutzung von BibRelEx gab es aufgrund des Funktionsumfang einige Anfangsschwierigkeiten bei den Benutzern. Nach einer relativ kurzer Einarbeitungszeit fanden sich die Benutzer aber gut in der komplexen Struktur von BibRelEx zurecht. Es hat sich dann gezeigt, dass gerade die Vielzahl der Einstellungs- und Darstellungsmöglichkeiten dazu geeignet ist, den unterschiedlichen Informationsbedürfnissen der Benutzer in verschiedenen Anwendungen gerecht zu werden. Beim Einarbeiten in neuen Arbeitsgebieten bietet beispielsweise die Zeitdarstellung einen guten Überblick über die Entwicklung eines Gebietes, wie wir in der Arbeit anhand der Geschichte der Voronoi-Diagramm, siehe Seite 5.6, exemplarisch gezeigt haben. Die kräftebasierten Darstellungen helfen Zusammenhänge innerhalb eines Wissensgebietes zu erkennen und zentrale oder fundamentale Arbeiten zu lokalisieren. Mit Hilfe der verschiedenen Cluster-Verfahren können Arbeiten nach bestimmten Kriterien eingeteilt werden, vgl. Seminarbeispiel mit Clusterung nach Kozitation auf den Annotationsbeziehungen in den Abbildungen 5.19, oder Themen in Unterthemen zerlegt werden, siehe Beispiel für die Anwendung des MajorClust-Verfahrens in Abbildung 5.20.

In den unterschiedlichen Anwendungen sind auch verschiedene Kritikpunkte bzw. Anregungen für Erweiterungsmöglichkeiten aufgetreten, die wir mit der Beschreibung zweier Anwendungsszenarien ausführlicher erläutern möchten. Die Seminarvorbereitung und auch das Recherchieren in Datenbasen wie beispielsweise GeomBib ist in der Arbeit schon an verschiedenen Stellen vorgekommen, so dass wir hier auf eine separate Beschreibung dieser beiden Anwendungsszenarien verzichten. Das Arbeiten in Verbindung mit GeomBib kommt außerdem noch im ersten der im folgenden beschriebenen Anwendungen vor.

7.1 Anwendungen und Ergebnisse

Anwendung 1: Überarbeiten von Publikationen

Publikationen sind Speicher wissenschaftlicher Informationen und unterliegen somit dem sich ständig ändernden Wissensfluss. Nach dem Erscheinen einer Arbeit kommen neue Erkenntnisse dazu, werden offene Probleme gelöst oder treten Verbesserungsvorschläge auf. Dieses neue Wissen wird vom Wissenschaftler gesammelt und führt zu neuen Publikationen, Folgepublikationen oder Neuauflagen bei Büchern. BibRelEx kann verwendet werden, um Überarbeitungsideen oder Arbeitsnotizen zu Veröffentlichungen zu sammeln und zu verwalten. In der Praxis hat sich gezeigt, dass es wertvoll ist, solche Informationen vor der Veröffentlichung so abzulegen.

In der Evaluierungsphase wurde BibRelEx am Lehrgebiet verwendet, um Überarbeitungsideen für das Kapitel *Voronoi Diagrams* von Aurenhammer und Klein [6] im Handbuch *Handbook of Computational Geometry* für eine Neuaulage zu sammeln. Im Folgenden wird das Kapitel kurz mit ak-vd-00, seinem Zitierschlüssel aus GeomBib, bezeichnet[1]. Das Kapitel steht im Kontext der Algorithmischen Geometrie und zitiert viele Arbeiten aus GeomBib. Daher konnte auch gleich das Arbeiten mit GeomBib getestet werden und die Teilung der Datenbasis nutzbringend eingesetzt werden.

Zu diesem Zweck wurde eine Datenbasis angelegt, die die Originaldatenbank GeomBib als globalen Teil verwendet, die Arbeitsnotizen in einen lokalen Teil schreibt und neue Arbeiten, die in dem Kapitel berücksichtigt werden sollen und auch für die Computational Geometry Community interessant sind in den update Teil der Datenbasis ablegt. So können die Arbeitsnotizen privat verwaltet werden und neue Arbeiten der Community mit der nächsten Aktualisierung zur Verfügung gestellt werden.

Zur Veranschaulichung, wie solche Arbeitsnotizen aussehen können, betrachten wir hier einen Ausschnitt aus dem in der Evaluierung entstandenem Wissensgeflecht, siehe Abbildung 7.1. Zum besseren Verständnis wurde in der Abbildung der Inhalt der eingegebenen Notizen (in Form von Dokumenten oder Annotationen) in Textboxen zugefügt.

Beim Aufbau des Geflechts wurden als erstes neue mit dem zu überarbeitenden Kapitel in Zusammenhang stehende Arbeiten eingegeben und diese über Links mit ak-vd-00 verbunden. Die Links wurden mit entsprechenden Typen versehen und anschließend mit Hilfe von Annotationen näher erläutert. Ein Beispiel hierfür sind die in der Abbildung durch violette Kan-

[1] Auch alle anderen in diesem Abschnitt verwendeten Arbeiten werden mit ihrem Zitierschlüssel aus GeomBib bezeichnet. Die zugehörigen ausführlichen bibliographischen Daten können dem Literaturverzeichnis entnommen werden.

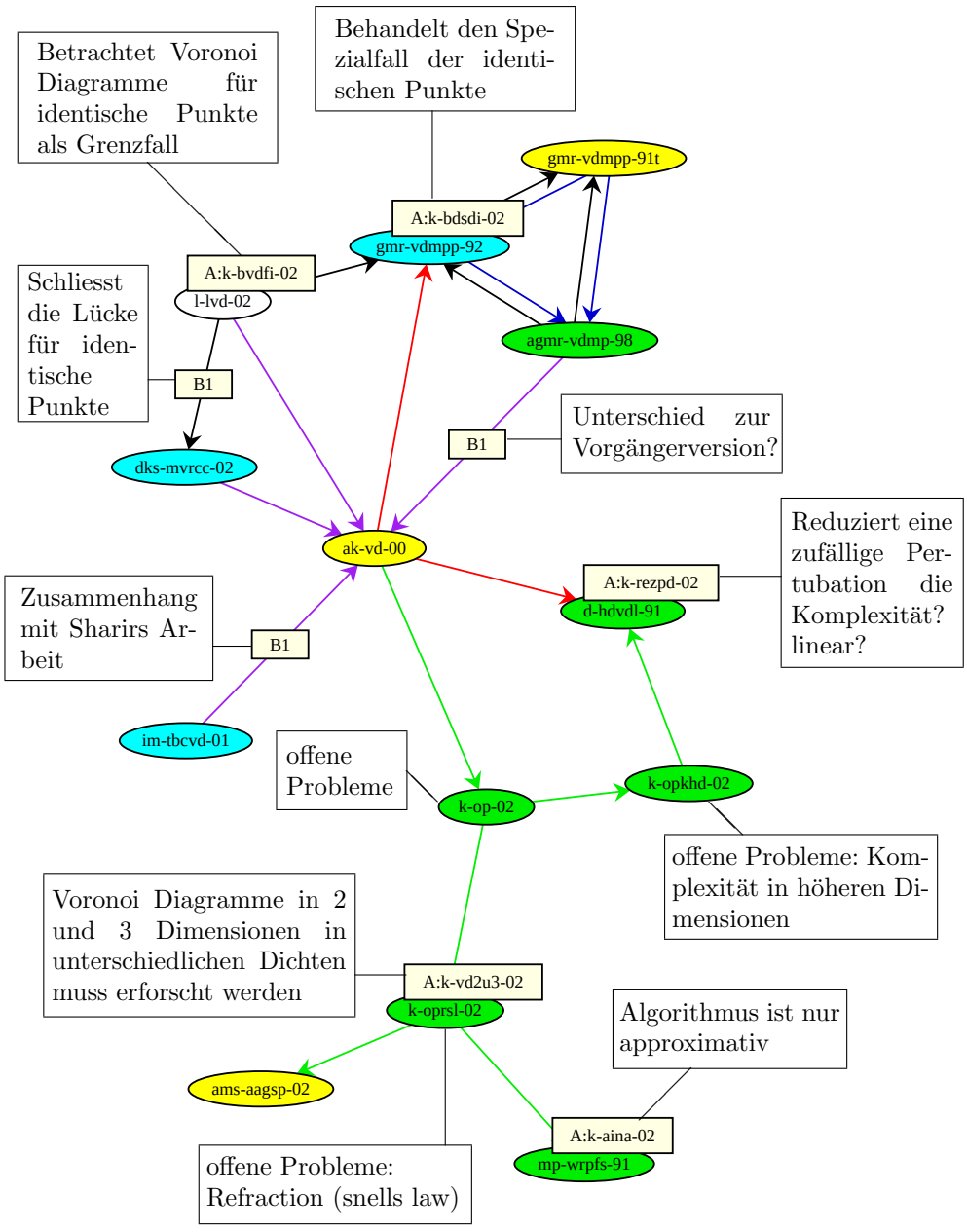


Abbildung 7.1: Arbeitsnotizen zu einer Veröffentlichung

ten dargestellten *muss zitiert werden* Beziehungen. Die Richtung dieser Kanten ist dabei wie folgt zu lesen: <Quelle> *muss von* <Ziel> *zitiert werden.* Zum besseren Verständnis erklären wir hier eine der annotierten *muss zitiert werden* Beziehungen: Die Arbeit gmr-vdmpp-92 [76] wurde bisher schon in ak-vd-00 zitiert. Zu dieser Arbeit ist inzwischen eine Nachfolgearbeit (agmr-vdmp-98 [2]) erschienen, wie wir übrigens erst bei der Visualisierung des entsprechenden Ausschnitts von GeomBib festgestellt haben. Dies zeigt einen weiteren zusätzlichen Nutzen der Literaturverwaltung mit BibRelEx und einer gemeinschaftlich gepflegten Datenbasis. Man erhält jederzeit einen aktuellen Überblick über das Umfeld der eigenen Veröffentlichungen und wird so auf neuere Entwicklungen aufmerksam. Bei einer Überarbeitung von ak-vd-00 sollte die aktuellere Version zitiert werden. Außerdem sollten dazu noch die Unterschiede zur Vorgängerversion festgestellt werden. Dieser Zusatz wurde als Annotation an die Beziehung angebracht.

Anschließend wurden noch offene Probleme in die Datenbasis eingebracht und mit Arbeiten, die für ihre Lösung nützlich sein könnten, verknüpft. Hier zeigte sich, dass es verschiedene Möglichkeiten gibt, diese Information in die Datenbasis einzubringen und dass es für den Benutzer nicht einfach zu entscheiden ist, welche Lösung die praktikabelste ist. Man kann beispielsweise die zur Lösung offener Probleme relevanten Arbeiten direkt über einen Link vom Typ *Offenes Problem* mit ak-vd-00 verknüpfen und dann mit einer Annotation an diesen Link näher beschreiben, welcher Art das offene Problem ist. Wir haben einen beim Einbringen in die Datenbasis etwas aufwendigeren Weg gewählt, der aber in der Visualisierung übersichtlicher ist. Zunächst wurde eine Dokumentbeschreibung (k-op-02 in der Abbildung 7.1) angelegt, die als Sammelpunkt für alle offenen Probleme bzgl. des Kapitels ak-vd-00 dient. Für jedes offene Problem wurde wieder eine Dokumentbeschreibung angelegt, die dieses Problem näher beschreibt und mit k-op-02 verknüpft (z.B. k-opkhd-02 und k-oprsl-02 in der Abbildung 7.1). Die für offene Probleme relevanten Dokumente wurden dann mit den entsprechenden *offenes Problem* Dokumentenbeschreibungen verknüpft. Bei Verwendung eines kräftebasierten Layoutverfahrens werden die offenen Probleme so getrennt von den anderen Notizen gut erkennbar angeordnet.

Alle Kanten, die zur Verknüpfung mit offenen Problemen verwendet wurden, erhielten den Typ *Offenes Problem.* Durch die Verwendung verschiedener Farben für Beziehungen unterschiedlichen Typs ist auf einen Blick zu erkennen, welche Teile des Geflechts welche Überarbeitungsideen betreffen: Die linke obere Hälfte mit violetten Kanten betrifft alle Arbeiten, die in ak-vd-00 noch zitiert werden müssen. Die untere rechte Hälfte des Geflechts mit grünen Kanten stellt die offenen Probleme dar.

Insgesamt hat sich gezeigt, dass die verschiedenen Möglichkeiten der Wis-

sensaggregation eine hohe Flexibilität erlauben, es dem Benutzer aber auch erschweren, Wissen einheitlich in gut wiederverwendbarer Form einzugeben. Je nach verwendeten Aggregationsmechanismus (Annotationen, typisierte Links, annotierte Links) sind bei der Recherche unterschiedliche Anfragemethoden (textbasiert oder strukturbasiert) zu nutzen. Außerdem sieht das Beziehungsgeflecht unterschiedlich aus, da Aggregation mit Hilfe von Links zu stärkerer Vernetzung führt, wohingegen Annotationen die Struktur des Geflechts nicht beeinflussen, vergleich auch Abbildung 4.2.

Daher sollten die verschiedenen Aggregationsmechanismen zunächst in kleineren Gruppen weiter erprobt werden, um dann geeignete Regeln für das Arbeiten in einer Community festlegen zu können. Denn das eingebrachte Wissen kann nur dann gezielt genutzt werden, wenn dem Benutzer klar ist, in welcher Form es vorliegt und wie er danach suchen kann.

Bisher kann man in der Visualisierungsoberfläche von BibRelEx interaktiv Knoten und Kanten annotieren und Dokumente bzw. Annotationen über Links miteinander verknüpfen. Beim Erzeugen der Dokumentbeschreibungen für offene Probleme wurde der Wunsch nach mehr Interaktionsmöglichkeiten geäußert. So sollte es möglich sein auch neue Dokumente in der Visualisierung erzeugen und ändern zu können.

Benötigt wird also eine Kombination der LEDA Visualisierungsoberfläche mit der QT-Benutzungsoberfläche zu der Datenhaltungskomponente BibManage. Ereignisse in der Visualisierungsoberfläche sollen zum Aufruf von Eingabemasken aus der BibManage Oberfläche führen und umgekehrt sollen Ereignisse in BibManage (wie z.B. Sichern von neuen oder geänderten Einträgen) eine Aktualisierung der Visualisierung veranlassen.

Hierfür bietet sich die Verwendung des Beobachtermusters an, dass sich leicht mit Hilfe des Signal/Slot-Konzepts von Qt umsetzen lässt, vgl. Abschnitt 6.1.5. Aufgrund der gegenseitigen Blockierung der beiden Oberflächen durch die Ereignisverarbeitung in der LEDA Visualisierung, siehe Abschnitt 6.5, kann aber stets nur eine der beiden Oberflächen aktiv sein, d.h. Ereignisse entgegen nehmen und verarbeiten. Da wegen Problemen mit der Xlib-Bibliothek keine portable Realisierung mit Threads in Frage kommt, kann derzeit keine volle Interaktion zwischen beiden Oberflächen erreicht werden. Um dennoch möglichst viel Interaktion in der Visualisierung zu erreichen, müssten die Eingabemasken in der Visualisierungsoberfläche nochmal implementiert werden. Die gesamte Oberfläche von BibRelEx in LEDA zu implementieren ist nicht sinnvoll, da LEDA den Schwerpunkt auf Datentypen, Algorithmen und die Visualisierung von Graphen gesetzt hat. Viele für eine komfortable Oberfläche notwendigen Bedienelemente wie Listviews sind in LEDA nur in einfacher Form vorhanden oder fehlen, so dass auf einige nützliche Funktionalität, die bereits in der Qt-Oberfläche implementiert ist,

wie z.B. das Hypertext-Hilfesystem, verzichtet werden müsste.

Uns erscheint es sinnvoller auf die angekündigte threadsichere Version der Qt Bibliothek und die Beseitigung des Bugs in der Xlib-Bibliothek zu warten. Zumal die derzeitigen Möglichkeiten der Interaktion – Annotationen und Links können interaktiv in der Visualisierung erzeugt werden, Dokumente nicht; Änderungen interaktiv nicht möglich – einen geeigneten Kompromiss darstellen. Für Änderungen oder die Neuanlage von Dokumentbeschreibungen kann einfach mit einem Button zwischen der BibManage-Oberfläche und der Visualisierung gewechselt werden. Bei größeren Änderungen an der Datenbasis muss die Visualisierung neu aufgebaut werden. Dieses zweistufige Vorgehen, d.h. im ersten Schritt die Daten erstellen bzw. ändern und im zweiten Schritt das Layout berechnen, entspricht dem Vorgehen bei Markup-Systemen wie beispielsweise dem Textverarbeitungssystem LaTeX[105].

Die Benutzer von BibRelEx haben weiterhin den Wunsch nach mehr textueller Information in der Visualisierung geäußert. Bisher kann der Benutzer zu jedem Knoten ein separates Textfenster anzeigen lassen, das die zugehörigen Beschreibungsdaten wie Autor, Titel, etc. enthält. Über eine Lupenfunktion erhält er außerdem eine Anzeige des vollständigen Schlüssels eines Dokuments oder einer Annotation, wenn die Maus sich länger über einem Knoten befindet. Analog erhält er Informationen über Quelle, Ziel und Typ eines Links wenn die Maus auf einem Link verweilt. So hilfreich diese Möglichkeiten sind, sieht der Benutzer dabei jeweils nur für einen Teil der Darstellung ausführlichere Informationen.

Die von BibRelEx erzeugte Darstellung des Beziehungsgeflechts in Abbildung 7.1 wurde zum besseren Verständnis um Textboxen erweitert, die den Inhalt der Annotationen bzw. den Titel von Dokumenten enthalten. Solche Zusatzinformationen sollten auch in BibRelEx wahlweise möglich sein. Beispielsweise könnte zu jedem Dokument der Titel oder eine Kurzform des Titels und zu jeder Annotation der Inhalt (in Kurzform) als Label in der Nähe des zugehörigen Knotens angeordnet werden. Zur Lösung dieses Problems kann man eines der zahlreichen bekannten Map Labeling Verfahren in BibRelEx integrieren. Eine Übersicht über Map Labeling Verfahren für Graph Drawing gibt [142]. Eine detaillierte und aktuelle Bibliography für Map Labeling findet sich unter `http://www.math-inf.uni-greifswald.de/map-labeling/bibliography/`.

Anwendung 2: Prüfungsvorbereitung

Ein Student hat BibRelEx in der Prüfungsvorbeitung eingesetzt. Er hat das System verwendet, um Zusammenhänge wie *Der Beweis von Satz C stützt sich auf Definition A und Satz B ab* oder *Korollar A wird in Satz B benutzt* zu erfassen. Das Fazit des Studenten war, dass BibRelEx für ihn bei der Prüfungsvorbereitung nützlich war und er „die fertige Bibliothek auch beim Lernen intensiv benutzen konnte“. Die Darstellung der Beziehungen zwischen den Lerninhalten haben ihm besonders beim Rekapitulieren des Stoffes geholfen.

Bei der Eingabe der Zusammenhänge zeigte sich aber, dass das Typkonzept von BibRelEx flexibler gestaltet werden sollte. Statt verschiedene Dokumenttypen sind hier Objekte vom Typ Satz, Definition, Lemma, usw. nötig. Zwar können mit Hilfe der Konfigurationsdateien für verschiedenen Dokumenttypen Pflicht- und optionale Felder festgelegt und benutzerdefinierte Typen erzeugt werden, aber die interne Abbildung auf Objekte des bestimmten Typs ist zu starr. So werden alle benutzerdefinierten Typen intern auf einen einzigen Typ (*userdefined*) abgebildet und nicht weiter unterschieden.

Die notwendige Flexibilisierung ist durch die Verwendung dynamischer Typen möglich, die mit Hilfe des *Type Object Pattern* [88] bzw. seiner Erweiterung *Class Object Pattern* [119] zur Definition dynamischer Typen und dem *Product Traders Pattern* [8] zur Erzeugung von Objekten eines dynamischen Typs realisiert werden können. Die hierfür notwendigen Änderungen an BibRelEx beschränken sich auf die Datenhaltungskomponente.

Dem Student gelang aber dennoch mit den zur Zeit zur Verfügung stehenden Möglichkeiten (benutzerdefinierte Typen), die Zusammenhänge geeignet einzugeben.

Bei dieser Anwendung von BibRelEx ist ein weiteres Problem aufgetreten. Zu den einzelnen Definitionen, Sätzen, usw. ist meistens keine explizite Jahresangabe möglich, um die „zeitliche“ Entwicklung des Lerngebietes nachzuvollziehen. Vielmehr bestimmen die Beziehungen zwischen den Definitionen, Sätzen, usw. diese Ordnung implizit. Hier zeigt sich, dass eine hierarchische Anordnung nicht nur in der Zeitdarstellung sinnvoll ist. Daraufhin haben wir BibRelEx um eine allgemeine hierarchische Darstellung ergänzt. Diese basiert wie die Zeit-Darstellung auf dem Algorithmus von Sugiyama aus der AGD-Bibliothek. Die einzelnen Schichten werden dabei nicht durch ein Erscheinungsjahr bestimmt, sondern die Schichtzuweisung in der 1. Phase des Algorithmus (siehe Algorithmus 4, Seite 46) basierend auf *Längster-Pfad-Schichtung* bestimmt, d.h. die Höhe der resultierenden Darstellung ist minimal (= Länge eines längsten gerichteten Pfades im Graphen). Das Ergebnis für das Geflecht zur Prüfungsvorbereitung zeigt Abbildung 7.2.

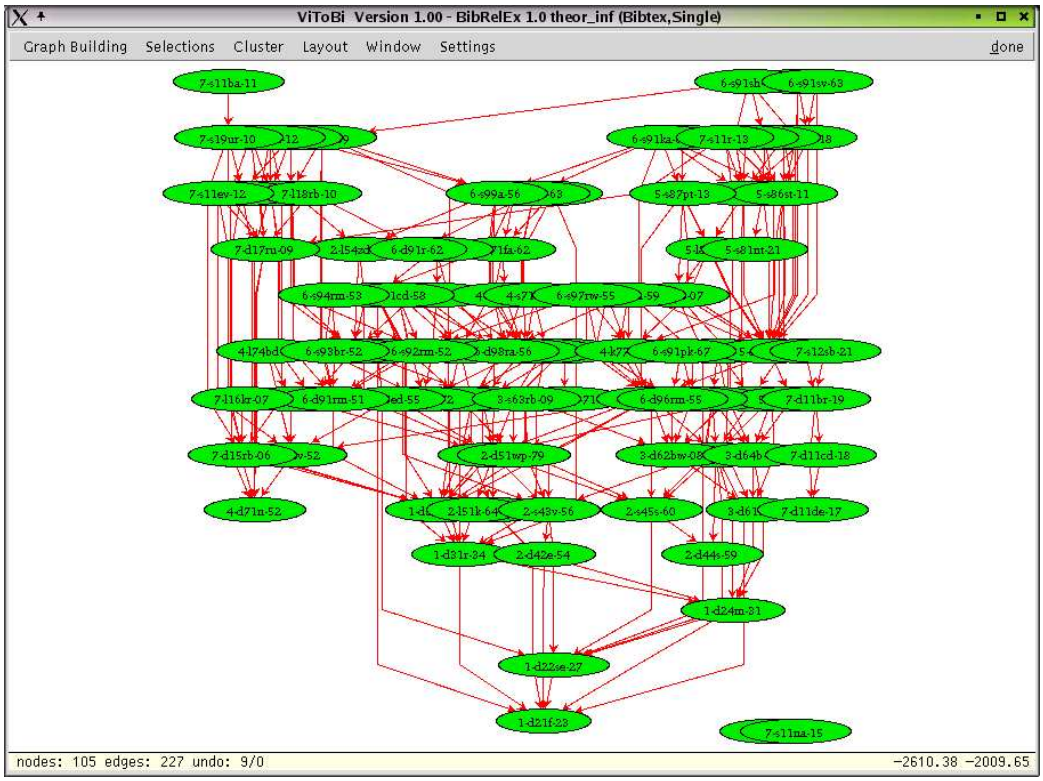


Abbildung 7.2: Hierarchische Darstellung des Geflechts zur Prüfungsvorbereitung

Zusammenfassend lässt sich feststellen, dass der Student zufrieden mit dem Einsatz von BibRelEx bei der Prüfungsvorbereitung war. „Es hat sich für ihn gelohnt, dass Programm zu verwenden.“ Die von ihm vermisste hierarchische Darstellung wurde ergänzt und die Eingabe der Zusammenhänge des Lernstoffes war trotz des etwas starren Typkonzepts von BibRelEx möglich. BibRelEx ist prinzipiell ein geeignetes Hilfsmittel, um den Lernstoff so zu organisieren und übersichtlich wiederzugeben, dass der Lernende Zusammenhänge versteht und die entstandenen Wissensgeflechte zur Rekapitulation des Stoffes einsetzen kann.

7.2 Fazit

Die verschiedenen Anwendungen von BibRelEx in der Evaluierungsphase haben gezeigt, dass die Aggregation von Wissen und die Visualisierung von Beziehungsgeflechten wesentlich dazu beitragen, schnell und gezielt Informationen zu finden und zu verwalten. Die Anwendungsmöglichkeiten von

BibRelEx gehen dabei über eine reine Literaturverwaltung weit hinaus, wie die verschiedenen Anwendungsszenarien in Kapitel 4 und im vorherigen Abschnitt gezeigt haben. Der Prototyp von BibRelEx ist gut für die Verwaltung privater Literaturbestände und für das Arbeiten in kleinen Gruppen geeignet.

Änderungsvorschläge aus der Evaluierungsphase betreffen vor allem den Bedienkomfort. So ist BibRelEx aufgrund seiner vielen optionalen Möglichkeiten nicht ganz einfach zu bedienen. Auf der anderen Seite wird dadurch aber gerade ein Nachteil vieler anderer Systeme vermieden, die Aufgrund zu vieler Vorgaben oft nicht in der Lage sind das Informationsbedürfnis des Benutzers wirklich zu decken. Für den „ungeübten“ Benutzer von BibRelEx sind aber Standardeinstellungen vorgegeben, die nach unseren bisherigen Erfahrungen für die meisten Anwendungen gute Ergebnisse liefern.

In Bezug auf die Funktionalität waren die Benutzer sehr zufrieden. In allen erprobten Anwendungsszenarien waren sie mit den angebotenen Möglichkeiten der Wissensaggregation in der Lage, die Datenbasis mit zusätzlichem Wissen anzureichern und so zu strukturieren, dass ihre verschiedenen Informationsbedürfnisse gedeckt wurden. Die Nützlichkeit der verschiedenen Darstellungsarten und Cluster-Verfahren in der Visualisierung hat sich bestätigt.

Die Evaluierung hat aber auch gezeigt, dass nicht immer klar ist, welche Methode zur Aggregation von Wissen die günstigste ist, um bestimmte Fakten in die Datenbasis einzubringen und nutzbringend wieder verwenden zu können. Die verschiedenen Aggregationsmechanismen müssen daher zunächst in kleineren Gruppen weiter erprobt werden, um dann geeignete Regeln für das Arbeiten in einer Community festlegen zu können.

Da BibRelEx bisher erst in kleineren Gruppen eingesetzt wurde, steht nur eine überschaubare Menge an Zusatzwissen bereit. Zu GeomBib wurden von uns zwar bisher über 3000 neue Einträge und 13000 Verweise beigesteuert. Die Aggregation von Expertenwissen in GeomBib erfolgte aber bislang nur in zwei Arbeitsgruppen[2], da - wie bereits weiter oben beschrieben - die Aggregationsmechanismen vor der Verteilung von BibRelEx weiter erprobt werden sollten. Von besonderem Interesse bei der Beurteilung von BibRelEx dürfte es sein, in Folge eines längeren Betriebes eine größere Wissenssammlung zu bilden. Insofern ist die Evaluierungsphase von BibRelEx als noch nicht abgeschlossen zu betrachten.

Zusammenfassend ziehen wir das Fazit, dass die entwickelten Konzepte grundsätzlich anwendbar sind und zu dem gewünschten Ergebnis führen.

[2]Lehrgebiet Praktische Informatik VI an der Fernuniversität Hagen und Abteilung I im Institut für Informatik der Rheinischen Friedrich-Wilhelms-Universität Bonn

Kapitel 8

Zusammenfassung und Ausblick

8.1 Ergebnisse

In der Arbeit wurde ein Konzept zur Wissensaggregation basierend auf Annotationen und Links, das die kollaborative Nutzung des Wissens durch periodische Aktualisierung unterstützt, entwickelt. Damit die Qualität der gemeinschaftlich erstellten Datenbasis sichergestellt werden kann, wurde ein Verfahren zur Konsistenzprüfung basierend auf den Literaturdaten entwickelt und Richtlinien zur gemeinschaftlichen Arbeit an einem dynamischen Datenbestand aufgestellt. Um eine effiziente Recherche zu ermöglichen, wurden verschiedene strukturbasierte Suchverfahren zur Nutzung inhaltlicher Beziehungen erweitert. Zusätzlich wurde die inhaltliche Navigation durch eine graphischen Visualisierung der Beziehungsnetzwerke nach benutzerdefinierten Kriterien unterstützt. Um unterschiedlichen Informationsbedürfnissen gerecht zu werden, wurden für die Visualisierung der Einsatz verschiedener Darstellungsarten und Cluster-Verfahren vorgeschlagen.

Die verschiedenen für die Konsistenzprüfung, Wissensaggregation und Visualisierung entwickelten Konzepte wurden in einem Prototypen integriert und in verschiedenen Anwendungen erprobt. Aufgabe des Prototypen war der Nachweis, dass die Aggregation von Wissen und die Visualisierung des Wissensgeflechts wesentlich dazu beitragen, schnell und gezielt Informationen zu finden und zu verwalten. Dies ist sicherlich gelungen.

Der Prototyp selbst ermöglicht bereits sehr viele Arten der Nutzung. Änderungsvorschläge aus der Evaluierungsphase betreffen vor allem den Bedienkomfort. In Bezug auf die Funktionalität waren die Benutzer sehr zufrieden. In allen erprobten Anwendungsszenarien waren die angebotenen Möglichkeiten der Wissensaggregation geeignet, die Datenbasis mit zusätzlichem Wissen anzureichern und so zu strukturieren, dass die verschiedenen Infor-

mationsbedürfnisse der Benutzer gedeckt wurden.

Als besonders wertvoll hat sich der Einsatz von BibRelEx beim Überarbeiten von Veröffentlichungen in Verbindung mit einer gemeinschaftlich gepflegten Datenbasis (hier GeomBib) erwiesen. Die eigenen Arbeiten werden dabei nicht isoliert sondern im Kontext der Veröffentlichungen in dem Forschungsgebiet betrachtet. Durch die übersichtliche Darstellung in der Visualisierung erkennt man schnell (neue) Arbeiten, die für die eigene Arbeit relevant sein können.

Das elektronische Abspeichern von Zusammenhängen, Arbeitsnotizen, etc. bei der Literaturauswertung und die anschliessende Visualisierung des entstandenen Geflechts geben ein klares Bild von dem betreffenden Themengebiet. Die Literatur wird so inhaltlich erschlossen und man kann jederzeit auf das Erfahrungswissen zurückgreifen.

Dabei wird von BibRelEx sowohl die persönliche als auch die communitybasierte Wissensverwaltung unterstützt. Durch die Verwendung von Konfigurationsdateien für die Vorgabe von Linktypen, Konferenznamen, Zeitschriften usw. ist eine gewisse Normierung innerhalb der wissenschaftlichen Community möglich. Durch die Verwendung privater Notizen kann ein individueller Wissensraum aufgebaut werden.

Eine weitere Zielsetzung des Prototypen war ein flexibles Werkzeug zu bieten, um weitere Algorithmen, z.B. für das Layout von Beziehungsgeflechten, zu testen. Das konsequente Einsetzen von Entwurfsmustern hat zu einer leicht erweiterbaren Software geführt. Beispielsweise ermöglicht die Verwendung des Besuchsmusters die Funktionalität zu erweitern, ohne dass Klassen selbst geändert werden müssen.

Die systematische Untersuchung der Anwendungstauglichkeit in einer wissenschaftlichen Community steht derzeit noch aus, da die verschiedenen Aggregationsmechanismen erst in kleinen Gruppen weiter getestet werden sollten.

Der Nutzen von BibRelEx basiert vor allem auf dem eingebrachten Expertenwissen. Hier ist sicherlich festzustellen, dass durch die derzeit noch nicht ausreichende Nutzung, ein gewisser Mangel besteht. Eine hinreichende Anreicherung mit Expertenwissen im Bereich der Algorithmischen Geometrie soll im Rahmen der anstehenden Evaluierung in der Community erfolgen. Für die Beitragenden wird das System dann interessant, wenn sie eine ausreichend große Zahl von Benutzern ansprechen. Hier ist also mit einer gewissen Vorlaufzeit zu rechnen, in der der Bekanntheitsgrad des Systems erstmal wachsen muss.

Wesentlich zur Verbereiterung der Anwenderschaft kann eine Anbindung an das WWW beitragen. Diesen und andere Erweiterungsvorschläge diskutieren wir im nächsten Abschnitt.

8.2 Weitere Arbeiten

Die Evaluierung des Prototypen von BibRelEx hat gezeigt, dass es sich bereits um ein vom Endbenutzer gut einsetzbares hilfreiches Werkzeug handelt. Dennoch sind noch einige Erweiterungen sinnvoll, um den Prototypen, der als Testplattform für neue Konzepte und Algorithmen entwickelt wurde, in ein vollwertiges wissensbasiertes Informationssystem zu überführen. Keine der Erweiterungsvorschläge hat Auswirkungen auf die in Kapitel 5 entwickelten Techniken, sondern lediglich auf die in Kapitel 6 vorgestellte spezifische Implementierung. Dass solche Erweiterungen wünschenswert sind um den Prototypen von BibRelEx in ein professionelles Werkzeug zu verwandeln, ist keineswegs negativ zu sehen, sondern liegt in der Natur eines Prototypen als Testimplementierung zu dienen. Es hat sich aber gezeigt, dass bereits der Prototyp ein umfangreiches und nützliches Werkzeug ist, das mit seiner derzeitigen Funktionalität zur persönlichen und gemeinschaftlichen Wissens- und Literaturverwaltung in kleineren Gruppen eingesetzt werden kann.

Bei der Evaluierung, Abschnitt 7.1, wurden bereits einige der Erweiterungsmöglichkeiten vorgestellt. Diese wollen wir hier zusammenfassend aufzählen und nicht nochmal beschreiben:

- Höherer Bedienkomfort durch Schaffung weiterer Interaktionsmöglichkeiten zum Bearbeiten der Datenbasis in der Visualisierung,
- Bessere Lesbarkeit der graphischen Darstellung durch Integration von mehr textueller Information mit Hilfe von Map Labeling Verfahren,
- Größere Flexibilität für den Einsatz in anderen Anwendungen wie beispielsweise Lernumgebungen durch dynamische Typen.

Daneben hat es sich gezeigt, dass es sinnvoll ist, die in dieser Arbeit in verschiedenen Anwendungsszenarien gezeigten verschiedenen Möglichkeiten der Wissensaggregation weiter zu erproben. Ziel der Erprobung ist geeignete Regeln zur Wissensaggregation an gemeinschaftlich genutzten Datenbasen aufzustellen. Nur so ist ein verteilter Einsatz von BibRelEx für alle nutzbringend möglich, denn für eine effiziente Recherche sollte das Wissen in einheitlicher Form vorliegen, damit dem Anwender klar ist, wie er es finden kann.

Der nächste Schritt muss dann sein, BibRelEx zusammen mit geeigneten Regeln zur Wissensaggregation in der Computational Geometry Community zu verteilen und eine größere Menge von Experten zum Informationsbeitrag in BibRelEx zu motivieren. Für die Beitragenden wird das System dann interessant, wenn sie eine ausreichend große Zahl von Benutzern ansprechen.

Auf der anderen Seite wird BibRelEx für die Nutzer erst dann interessant, wenn die Datenbasis hinreichend mit Expertenwissen angereichert ist. Neben dem Problem eine hinreichend große Wissensmenge zu erreichen, die notwendig ist um BibRelEx für alle Nutzer - Informationsbeitragende und Informationssuchende - attraktiv zu machen, gibt es psychologische Hürden bei der Wissensverteilung. Das Preisgeben des eigenen Wissens wird häufig als subjektiver Machtverlust („Wissen ist Macht“) empfunden. In dem DFG-Projekt *Wissensaustausch mittels einer geteilten Datenbank* [39] wird untersucht, inwieweit die Funktionalität von Datenbanken Nutzer dazu motivieren kann, eigenes Wissen in eine Datenbank einzubringen. Ergebnisse aus diesem Projekt können eventuell nutzbringend in BibRelEx einfließen. Das lebhafte Interesse, das BibRelEx auf Fachtagungen, z.B. einem Dagstuhl-Seminar über Algorithmische Geometrie auf der wir das System vorgestellt haben, entgegengebracht wurde, macht uns zuversichtlich, dass die Community unser System annehmen wird.

Aufgrund des relativ langen Aktualisierungszyklus von GeomBib (4 Monate) muss der Betrieb von BibRelEx innerhalb der Community über einen verhältnismäßig langen Zeitraum hinweg beobachtet werden, um zuverlässige Aussagen treffen zu können. Im Rahmen dieser Arbeit war dies aus Zeitgründen leider nicht mehr möglich.

Neben den aus der Evaluierung resultierenden Erweiterungswünschen, möchten wir hier noch einige Erweiterungen vorschlagen, die zur Erhöhung der Benutzerakzeptanz, Skalierbarkeit und zu einer breiteren Nutzung beitragen können:

- Preprocessing zur Erhöhung der Skalierbarkeit
- Anpassung an weitere Datenhaltungssysteme
- Erweiterungen der Recherchemöglichkeiten
- Anbindung an das WWW
- Profildienst

In den folgenden Abschnitten betrachten wir die Erweiterungsmöglichkeiten der Reihe nach, und beschreiben ihren Nutzen und die Änderungen, die sie erfordern.

Preprocessing zur Erhöhung der Skalierbarkeit

Ein Engpaß in BibRelEx ist zur Zeit die Skalierbarkeit einiger Clusterverfahren. Beim hierarchischen Clustern wird beispielsweise auf einer Ähnlichkeits-

matrix gearbeitet, die bei großer Knotenzahl schnell zu Speicherplatzproblemen führt. Hier muss das Verfahren besser skalierbar implementiert werden, z.B. indem für textbasierte Clusterung die Worthäufigkeiten im Index gespeichert werden. Analog ist für die Clusterung basierend auf der Zitierrelation eine Speicherung der CCIDF-Werte sinnvoll. Die Worthäufigkeiten und CCIDF-Werte werden auch bei der Recherche genutzt. Da davon auszugehen ist, dass in dem System wesentlich mehr recherchiert wird als Daten geändert, ist so auch eine höhere Laufzeitperformance zu erreichen.

Lösungen für dieses Problem sind hinreichend bekannt, beispielsweise die Implementierung mittels einer relationaler Datenbank (nach [75]): Man benötigt dazu drei Tabellen zur Speicherung der indexierten Daten:

- Dokumenttabelle (eventuell mit zusätzlichen Metadaten)
- Termtabelle, die alle Terme und die Dokumentenhäufigkeit enthält
- Invertierte Liste, die Dokumenten-Term Paare mit der Termhäufigkeit enthält

Die Auswertung erfolgt mittels geeigneter Select-Statements und einer Hilfstabelle. Dieses Vorgehen hat eine Reihe von Vorteilen. Durch die Verwendung eines Datenbankmanagementsystems stehen Transaktionen, Recovery, Caching und einiges mehr zur Verfügung und man erreicht eine gute Performance bei einfacher Implementierung.

Für die Implementierung der Tabellen kann die in BibRelEx bereits zur Speicherung der Indexdateien genutzte BerkeleyDB verwendet werden.

Anpassung an weitere Datenhaltungssysteme

Bisher wurde der Prototyp von BibrelEx nur für BibTEX-Dateien verwendet. Literaturdaten werden häufig auch in anderen Formaten wie beispielsweise Refer [111] verwaltet. Die Anbindung verschiedener Datenhaltungssystem wurde bereits beim Entwurf des Prototypen berücksichtigt, siehe Abschnit 6.1.4. Damit sind lediglich noch die Schnittstellenanpassungen für weitere Formate zu implementieren.

Erweiterung der Recherchemöglichkeiten

BibRelEx bietet bereits zahlreiche Recherchemöglichkeiten. Es wird sowohl textbasierte als auch strukturbasierte Recherche unterstützt und die Suche nach ähnlichen Dokumenten, Hubs und Authorities ist möglich. Darüberhinaus kann man die Behandlung der Beziehungen verfeinern. Beispielsweise

haben Zitate innerhalb einer Arbeit unterschiedliche Wichtigkeit bzw. Bedeutung. Für den einzelnen Benutzer können Links verschiedener Typen unterschiedlich wichtig sein. Sowohl die unterschiedliche Bedeutung von Zitaten als auch die Berücksichtigung der Nutzerbedürfnisse kann man leicht mit Hilfe einer Gewichtung der entsprechenden Beziehungstypen in der internen Datenbasis realisieren.

Zur Unterstützung der Benutzer kann man eine weitere Formalisierung der Anfragemöglichkeiten, die die Wissensaggregation berücksichtigt, vorsehen. Sinnvoll wären beispielsweise Regeln für Anfragen wie *Wer ist Experte im Gebiet X?*

Die Frage nach den Folgearbeiten einer Publikation kann bisher in BibRelEx nur mit Hilfe der Visualisierung beantwortet werden. Mit relativ wenig Aufwand kann hier die Abfragesprache um folgende Anfrage erweitert werden:

(Query:) follow <citetag>

Ergebnis dieser Anfrage ist eine Liste der Einträge, die direkte oder indirekte Zitiernachfolger der durch <citetag> gegebenen Arbeit sind. Als Rank eines Ergebniseintrags kann seine Entfernung (kürzester Zitierpfad) von der vorgegebenen Arbeit geliefert werden. Zur weiteren Einschränkung der Suchergebnisse ist zu überlegen, ob es sinnvoll ist, eine maximal zu berücksichtigende Enfernung vorzugeben.

Anbindung an das WWW

In der Arbeit sind bereits Vorschläge gemacht worden, wie man das System einer breiten Nutzerschaft zugänglich machen kann. Vorrangig ist hier die Anbindung an das WWW zu nennen. Der Zugriff auf die Datenbasis mit allen Recherchemöglichkeiten ist leicht durch die Schaffung einer CGI-Schnittstelle mit Hilfe des Kommandozeileninterfaces zu BibManage realisierbar. Außerdem können mit Hilfe des Netscape Plugins aus dem *Qt Extension* Paket die Eingabemasken der Benutzungsoberfläche sowohl im Netscape als auch im Internet Explorer genutzt werden.

In Abschnitt 5.3 wurde die Vergabe von unterschiedlichen Zugriffsrechten diskutiert, um die Zugriffe über das WWW zu steuern. Die Vergabe von Zugriffsrechten wird von der in BibManage verwendeten Datenbank BerkeleyDB unterstützt, so dass die Erweiterung um Zugriffsrechte ohne großen Änderungsaufwand möglich ist.

Weitgehende Änderungen sind dagegen bzgl. der Visualisierung notwendig, da LEDA nicht über das WWW genutzt werden kann. Hier ist eine Umstellung auf die *Virtual Reality Modeling Language* (VRML) [190], eine

standardisierte Sprache für die Beschreibung von interaktiven 3D-Objekten und -Welten für das WWW, notwendig. VRML ist eine reine Beschreibungssprache. Daher fehlen jegliche prozedurale Elemente einer Programmiersprache. Somit lassen sich zunächst nur statische Szenarien erzeugen, die ungeeignet zur Visualisierung variabler Ergebnismengen, wie sie in BibRelEx auftreten, sind. Um eine dynamische Visualisierung mit VRML zu realisieren, muss daher eine Bibliothek implementiert werden, die es ermöglicht, komplexe VRML-Skripte zu generieren. Einen auf Perl basierten Ansatz hierzu findet man in [41].

Profildienst

Damit es nicht zu einer Wissensüberflutung des Einzelnen in der Community kommt, ist es hilfreich, wenn das Wissen nur an diejenigen verteilt wird, die es benötigen bzw. nutzen können. Innerhalb einer Community kann man dies dadurch erreichen, dass man jedem Benutzer ein Profil zuordnet, in dem seine Interessenschwerpunkte definiert sind und mit dessen Hilfe ihm Informationen selektiv zugeteilt werden können. So kann das System einen Nutzer jederzeit über Änderungen der für ihn relevanten Informationsobjekte unterrichten lassen. Beispielsweise nutzen das Recommender-System Knowledge Pump [71], die Informationsdrehscheibe der Fakultät für Informatik der TU München [98] und das Informatik-Fachinformationssystem Ariadne [63] Benutzerprofile zur gezielten Informationsverteilung. Für BibRelEx bietet sich die Integration eines Profildienstes an, wenn das System auf eine WWW-Anbindung verbunden mit einer Authentifizierung erweitert wird.

Standards

Es ist zu prüfen, inwieweit es – im Hinblick auf die Erweiterung von BibRelEx als verteiltes System – sinnvoll ist, Standards, wie beispielsweise MAB[1] zu berücksichtigen. Sinnvoll für ein verteiltes System ist sicher, das auf GeomBib basierende Schlüsselkonzept in BibRelEx durch den Einsatz standardisierter Identifier zur Identifikation von Objekten, z.B. *Digital Object Identifier* (DOI) [86], zu ersetzen.

[1] Maschinelles Austauschformat der Bibliotheken in Deutschland. Es wird innerhalb des Z39.50 Standards verwendet

8.3 Abschließende Bemerkungen

In dieser Arbeit wurden Konzepte für eine wissensbasierte Recherche und Literaturverwaltung entwickelt und prototypisch implementiert. In den mit Hilfe des Prototypen evaluierten Anwendungszenarien hat sich gezeigt, dass die Aggregation von Expertenwissen und die Visualisierung von Beziehungsgeflechten wesentlich dazu beitragen, schnell und gezielt Informationen zu finden und zu verwalten. Die Anwendbarkeit der entwickelten Konzepte ist dabei nicht nur auf Literaturdatenbanken beschränkt.

Anhang A

Interaktionsbeispiel

In diesem Anhang werden die wichtigsten Bildschirmmasken der Benutzungsoberfläche zu BibManage anhand eines typischen Arbeitsablaufs vorgestellt.

Zuerst wird der Benutzer eine Datenbasis anlegen oder eine bestehende auswählen, siehe Abbildung A.1.

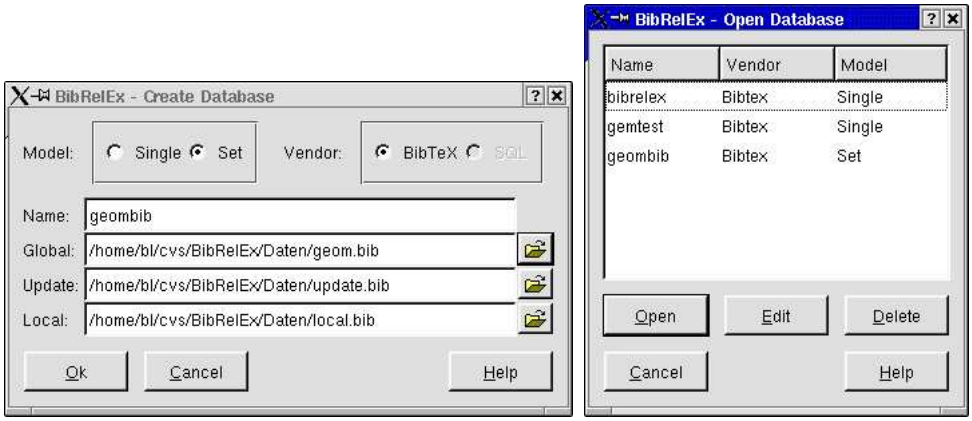


Abbildung A.1: Anlegen/Auswahl eines Datenhaltungssystems

Sobald eine Datenbasis angelegt oder geöffnet wurde, wird im oberen Teil des Hauptfensters (Abbildung A.2) eine Liste aller Einträge angezeigt. Im unteren Teil des Hauptfensters wird der jeweilig aktuelle Eintrag (Auswahl durch Mausclick) vollständig angezeigt. Durch Menüauswahl, Tastenkürzel oder Doppelclick kann ein Eintrag zum Editieren geöffnet werden.

Abbildung A.3 zeigt die Eingabemaske für bibliographische Einträge am Beispiel eines Konferenzbeitrags. Pflichtfelder sind fett umrandet. Für neue Einträge eines Typs werden zunächst nur die Pflichtfelder angezeigt, optionale Felder können aus einer Auswahlliste ergänzt werden. Verweise wie im

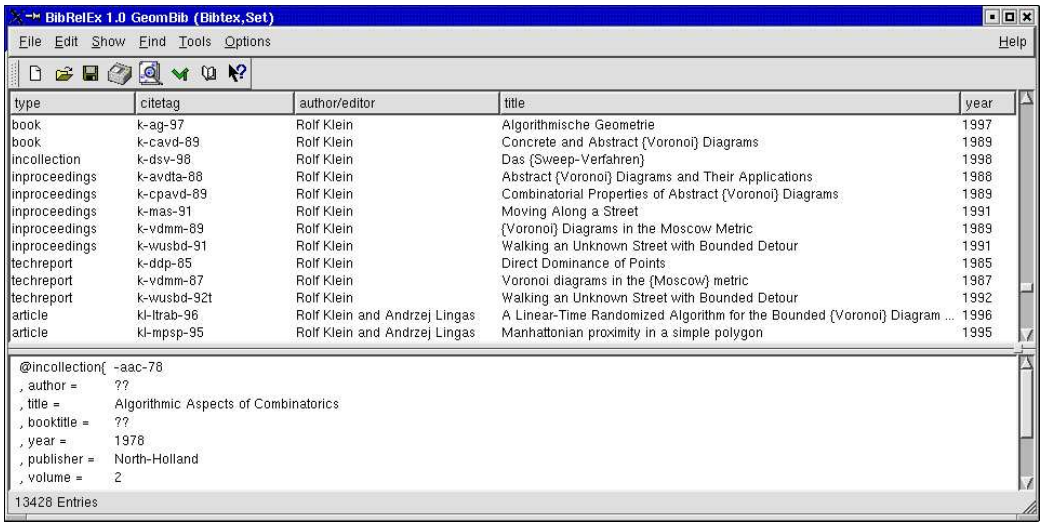


Abbildung A.2: Das Datenbankfenster

cites-Feld können durch Anklicken des Pfeils direkt verfolgt werden. Es wird dann ein Fenster für den entsprechenden Eintrag geöffnet.

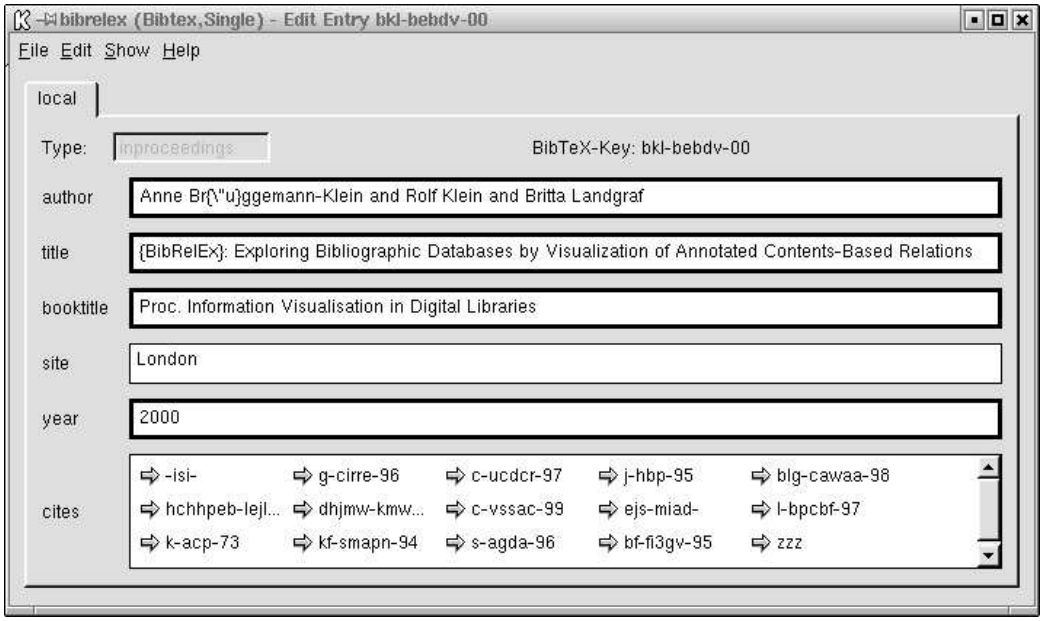


Abbildung A.3: Eingabemaske für bibliographische Einträge

Für Einträge mit einem *URL*-Feld können die zugehörigen Dokumente direkt (Kontextmenü, rechte Maustaste) geladen werden.

Für Textfenster sind Kontextmenüs definiert, über die Abkürzungen, Konferenznamen, Schlüsselworte, Zeitschriften, usw. ausgewählt werden können. Abbkürzungen können durch den Benutzer bearbeitet werden, siehe Abbildung A.4.

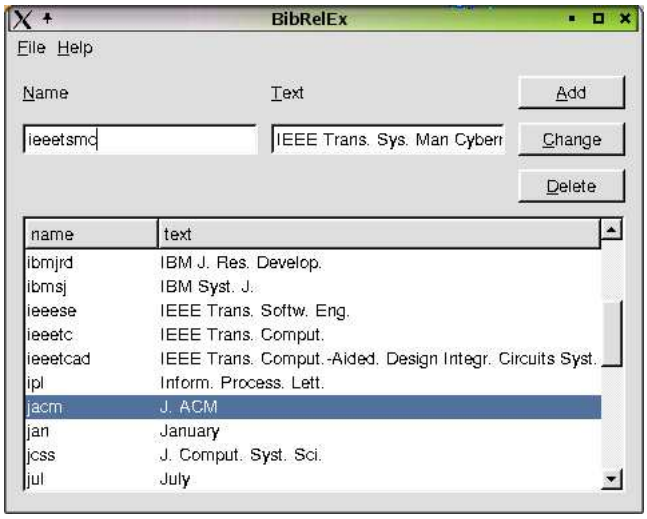


Abbildung A.4: Bearbeiten von Abkürzungen

Der Suchdialog ist in Abbildung A.5 dargestellt. Die Schaltflächen im oberen Bereich sind für die Bediengung des Dialogs, wie z.B. das Starten der Abfrage oder das Schließen des Dialogs, zuständig. Mit Hilfe des „Fields“-Button können die unteren Schaltflächen ein-/ausgeblendet werden. Sie dienen als Eingabehilfe für das Erzeugen einer Suchanfrage. Beim Betätigen einer dieser Schaltflächen wird die Suchanfrage um den zugehörigen Text ergänzt.

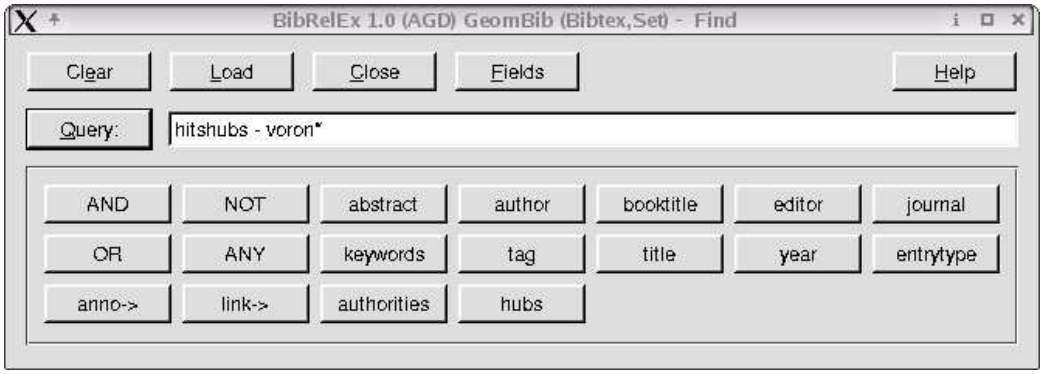
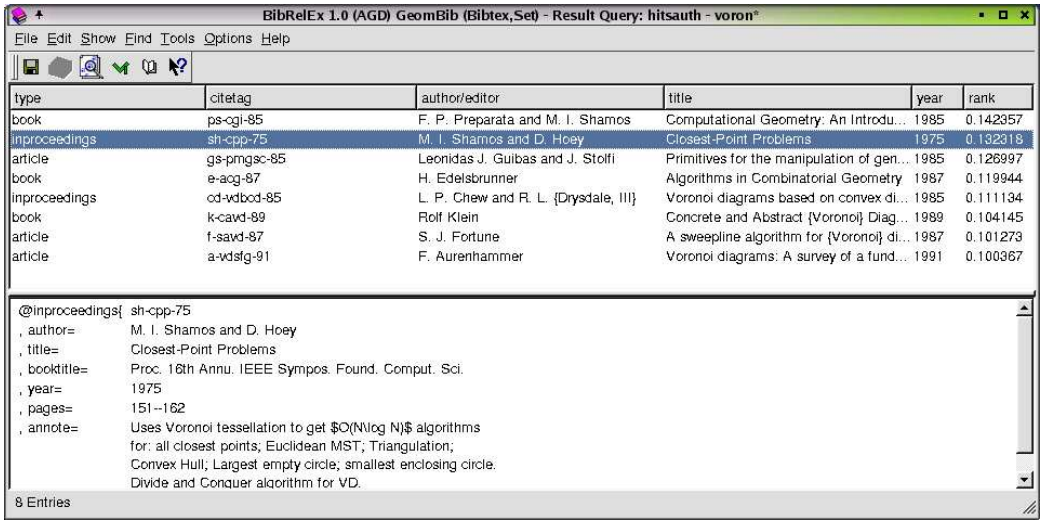


Abbildung A.5: Der Suchdialog

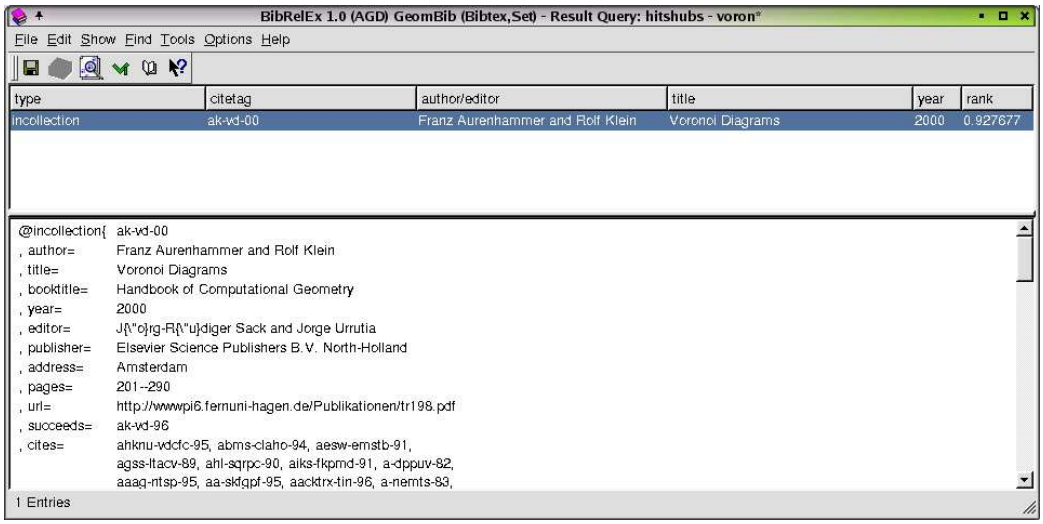
Das Ergebnis einer erfolgreichen Suche wird in der in Abbildung A.6 gezeigten Listendarstellung analog zu der Listendarstellung im Haupfenster angezeigt, wobei in einer zusätzlichen Spalte bei einigen Suchmethoden – wie hier z.B. nach dem HITS-Algorithmus (Abschnitt 2.4, Seite 21) – der Rank des jeweiligen Suchergebnisses angezeigt wird. Die Suche ist hierarchisch aufgebaut, d.h. wird in einem Ergebnisfenster erneut eine Anfrage gestellt, so wird nur in dieser Ergebnismenge gesucht. Auf diese Art kann man Anfragen verfeinern.

Neu eingegebene oder geänderte Einträge werden beim Speichern auf Konsistenz zu allen Einträgen in der Datenbasis untersucht (vgl. Abschnitt 5.1). Bei neu importierten Datenbasen empfiehlt es sich, die gesamte Datenbasis eines Konsistenztests zu unterziehen. Das Ergebnis dieses Tests wird in einer Listendarstellung (Abbildung A.7) angezeigt. Der obere Teil des Fensters enthält die Liste der Inkonsistenzen, im unteren Teil wird die jeweils aktuell ausgewählte Inkonsistenz angezeigt, indem die inkonsistenten Einträge und das zugehörige Vergleichsergebnis in drei Teilfenstern nebeneinander dargestellt werden.

Der Benutzer kann jederzeit das kontextsensitive Hypertext-Hilfesystem von BibRelEx (Abbildung A.8) aufrufen.



(a) Authorities



(b) Hubs

Abbildung A.6: Das Ergebnisfenster

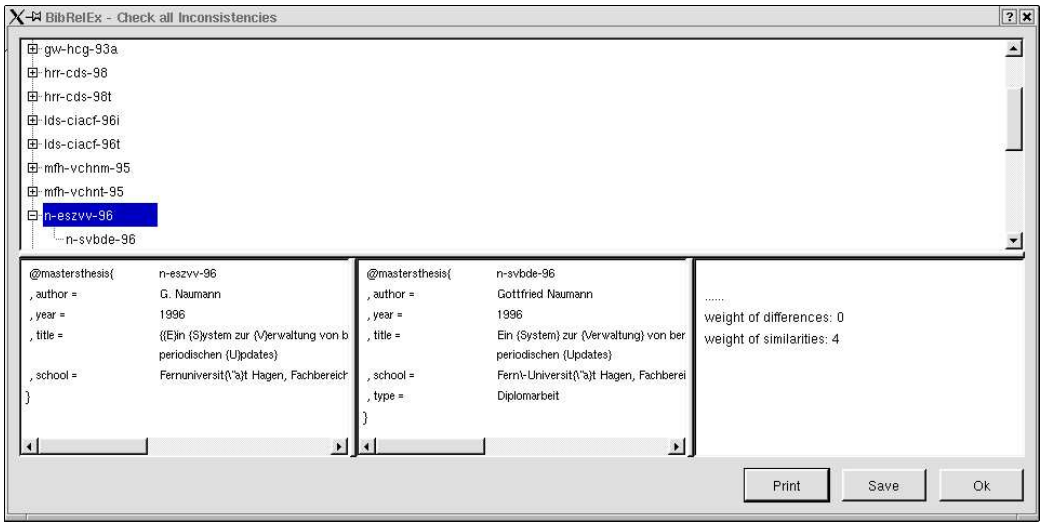


Abbildung A.7: Bearbeitung von Inkonsistenzen

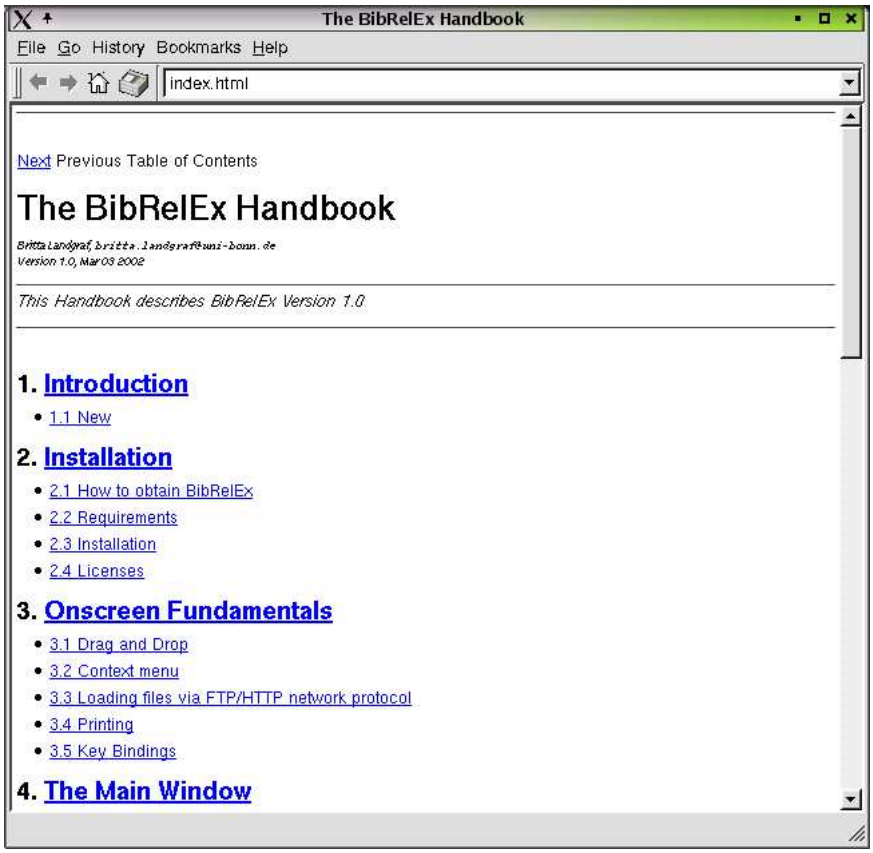


Abbildung A.8: Hypertext-Hilfesystem

Anhang B

Anfragesprache

In diesem Abschnitt wird die Syntax der Anfragen, die man im Suchdialog, siehe Abbildung A.5, eingeben kann, und ihre Wirkung beschrieben. Die Syntax lehnt sich an die des den Mitgliedern der Computational Geometry Community bekannten Tools biblook an.

(Query:) [not] <Feld> <Worte> [and|or [not] <Feld> <Worte>]*

durchsucht die Einträge der Datenbasis nach allen Worten aus <Worte> in den Feldern, deren Präfix mit <Feld> übereinstimmen. So wird in den Feldern *author* und *address* gesucht, wenn 'a' als Feldbezeichner eingegeben wurde. 'au' reduziert die Suche auf das *author* Feld. Soll in allen Feldern gesucht werden, so ist anstatt eines Feldbezeichners '-' zu benutzen. 'not' vor <Feld> negiert die Abfrage, d.h. sucht nach Einträgen, die keins der entsprechenden Worte in den angegebenen Feldern enthält.

Bei der Suche werden Groß- und Kleinschreibung nicht unterschieden. Endet ein Wort mit *, so wird es als Präfix angesehen.

Teilabfragen können mit 'and' oder 'or' verknüpft werden. Es wird dann die Schnittmenge bzw. Vereinigungsmenge der Ergebnisse der einzelnen Abfragen gebildet. Wird innerhalb einer Ergebnismenge erneut eine Anfrage abgesetzt, so entspricht das einer and-Verknüpfung.

Beispiel B.1 zeigt die Verwendung textbasierter Anfragen.

Neben den textbasierten Anfragen, die in dieser Form bereits in biblook möglich waren, unterstützt BibRelEx auch strukturbasierte Anfragen:

(Query:) panno [not] <Feld> <Worte>

findet alle Einträge, zu denen es Annotationen gibt, die die hinter panno angegebene textbasierte Anfrage erfüllen. Analog findet

(Query:) plink <Feld> <Worte>

alle Einträge, die von mindestens einem Link, der die angegebene Anfrage erfüllt, referenziert wird.

Eine weitere strukturbasierte Anfragemöglichkeit ist die Suche nach Authorities und Hubs, siehe Abschnitt 2.4, Seite 21 und Beispiel B.3.

Die letzte Möglichkeit der Suche, die BibRelEx zur Zeit anbietet, ist die Suche nach ähnlichen Dokumenten. Diese kann entweder durch den entsprechenden Menüpunkt ausgelöst werden oder durch Angabe der Anfrage:

(Query:) sim citetag

wobei citetag der Schlüssel der zugrundeliegenden Arbeit ist. Berechnet wird die Ähnlichkeit anhand des CCIDF.

Alle strukturbasierten Anfragen können auch wieder untereinander mit and oder or verknüpft werden.

Beispiel B.1 Textbasierte Anfragen

(Query:) au landgraf and y 1996
findet alle Arbeiten, die 1996 veröffentlicht wurden und einen (Mit)Autor Namens Landgraf haben.

(Query:) t geom al**
sucht alle Einträge mit den Präfixen 'geom' und 'al' im Titel, d.h. entspricht der Anfrage (query:) t geom* and t al*

*(Query:) ab **
liefert genau diejenigen Einträge, für die ein Abstrakt angegeben wurde.

Beispiel B.2 Strukturbasierte Anfragen

(Query:) panno ti advanced and au klein
findet alle Arbeiten des Autors Klein, zu denen es eine Annotation gibt, die das Schlüsselwort 'advanced' im Titel enthält.

*(Query:) plink au klein and t geom**
sucht alle Arbeiten mit dem Präfix geom im Titel, die von einem Link des Autors Klein referenziert werden.

(Query:) panno - experienced
liefert alle Arbeiten, die eine Annotation mit dem Wort 'experienced' in einem beliebigen Feld haben.

Beispiel B.3 Suche nach Hubs und Authorities

(Query:) hitsauth ti visualization
liefert fundamentale Arbeiten zum Thema 'visualization', wobei die Basismenge aufbauend auf der Menge der Arbeiten, die im Titel das Wort 'visualization' enthalten, konstruiert wird. Zur Erinnerung: Die gefundenen Authorities selbst müssen nicht die Anfrage erfüllen, d.h. müssen nicht das Wort 'visualization' im Titel enthalten. Zusätzlich wird der Rank des Suchergebnisses angezeigt. Dokumente mit höherem Rank sind „bessere“ Authorities. Analog findet

(Query:) hitshubs ti visualization
Übersichtsarbeiten zum Thema 'visualization'. Die Basismenge wird wie im vorherigen Beispiel aufgebaut. Dokumente mit höherem Rank sind „bessere“ Hubs.

(Query:) hits ti visualization
liefert alle Arbeiten, die direkt oder indirekt (je nach gewählter Iterationstiefe im HITS Algorithmus) Arbeiten zitieren bzw. von Arbeiten zitiert werden, deren Titel das Wort 'visualization' enthält.

Anhang C

Konfigurationsdateien

Alle Einstellungen innerhalb von BibRelEx können in Konfigurationsdateien gespeichert werden. Diese Konfigurationsdateien sind ASCII-Dateien, die lesbar strukturiert sind. Das Format lehnt sich an das für KDE übliche Format an (Die Behandlung von Kommentarzeichen in einer Konfigurationsdatei ist derzeit noch nicht implementiert.): Der Inhalt der Konfigurationsdatei ist in Gruppen unterteilt. Der Beginn einer Gruppe wird durch einen Gruppennamen in eckigen Klammern bezeichnet. Anschließend werden Paare von Schlüsseln und zugehörigen Werten gebildet oder eine Liste von Zeichenketten angegeben. Das Beispiel C.1 zeigt einen Ausschnitt aus der globalen Konfigurationsdatei zu BibRelEx.

Beispiel C.1 Aufbau der Konfigurationsdateien

```
...
[Types]
article
book
booklet
conference
...
[StdAbbrevs]
jan = "January"
feb = "February"
mar = "March"
...
```

BibRelEx verwendet zwei Konfigurationsdateien: Die eine Konfigurationsdatei (bibmanage.conf) steht im selben Verzeichnis wie BibRelEx, die

andere (BibRelEx.rc) im Arbeitsverzeichnis des Benutzers. Die erste ist die globale Konfigurationsdatei mit den Standardeinstellungen und wird von allen Anwendern benutzt. Die zweite Datei ist die lokale Konfigurationsdatei, die die Änderungen des jeweiligen Benutzers enthält. Beide Dateien dürfen gleiche Gruppen und gleiche Schlüssel enthalten, wobei die lokale Datei Vorrang vor der globalen hat.

Die Implementierung der Konfigurationsdateien befindet sich im Teilsystem Utilities (UTIL_Config, UTIL_ConfigGroup und UTIL_ConfigFile), wobei der Zugriff auf die Konfigurationsdateien in UTIL_Config gekapselt wird. Dabei wird wieder nach dem Singleton-Prinzip sichergestellt, dass nur ein solches Konfigurationsobjekt existiert.

Das Beispiel C.2 zeigt die Verwendung des Konfigurationsobjektes.

Beispiel C.2 Verwendung des Konfigurationsobjektes

```
...
#include "Utilities/UTIL_Config.h"
...
UTIL_getConfig().setGroup( "NodeColor" );
for ( unsigned i=0; i<node_types.size(); i++ )
{
   string type = node_types[i];
   string color = UTIL_getConfig().getEntry( type, (string) "" );
   set_Node_color( type, color );
}

UTIL_getConfig().setGroup( "Visual" );
int size_node = UTIL_getConfig().getEntry( "size_node", 40 );
bool show_anno = UTIL_getConfig().getEntry( "show_anno", false );
...
UTIL_getConfig().setGroup( "Visual" );
UTIL_getConfig().setEntry( "size_node", size_node );
UTIL_getConfig().setEntry( "show_anno", show_anno );
...
```

Anhang D

Legende zu den BibRelEx Visualisierungen

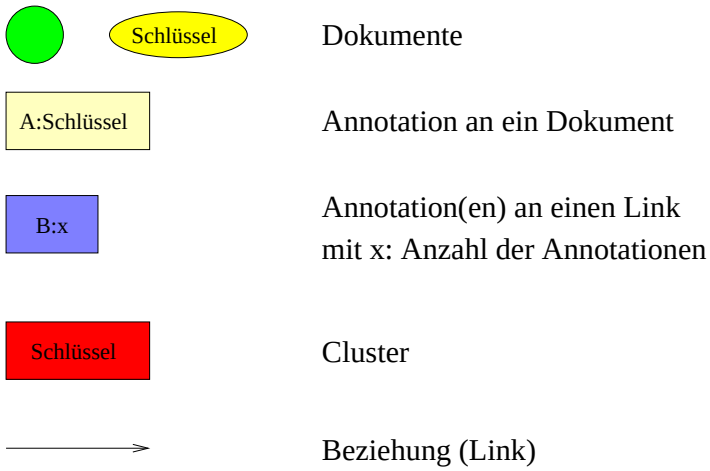


Abbildung D.1: Legende zu den BibRelEx-Visualisierungen

Die Knotenfarben geben den Typ eines Dokuments wieder, die Kantenfarben den Typ einer Beziehung. Bei den Bildern in dieser Arbeit wurde folgende Zuordnung gewählt[1]:

Kanten:

- cites: rot

[1]Da für verschiedene Anwendungen unterschiedliche Beziehungstypen verwendet wurden, ist die Zuodnung der Kantenfarbe bei diesen unterschiedlich und nicht in der Tabelle wiedergegeben.

- annote: blau
- precedes: blau
- succeeds: schwarz
- cooccur[2]: green
- ohne Typ: schwarz

Knoten:

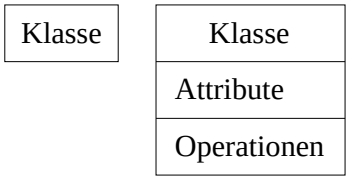
- cluster: rot
- annotation: hellblau/beige
- article: grün
- book: gelb
- booklet: gelb
- conference: weiß
- inbook: violett
- incollection: gelb
- inproceedings: zyan
- manual: weiß
- mastersthesis: weiß
- misc: grün
- phdthesis: weiß
- proceedings: weiß
- techreport: gelb
- unpublished: weiß
- userdefined: weiß

[2]Kanten im minimalen Spannbaum basierend auf Co-Occurence-Ähnlichkeiten

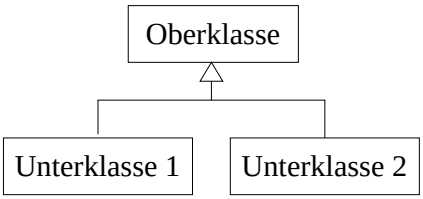
Anhang E

Grundlegende UML Notation für Klassendiagramme

In diesem Anhang sind die in der vorliegenden Arbeit verwendeten Symbole für Klassendiagramme zusammengestellt. Die Notation basiert auf der in [56] beschriebenen *Unified Modeling Language* (UML). Die wichtigsten Elemente eines Klassendiagramms sind in Abbildung E.1 dargestellt. Die einzelnen Konzepte für die sie stehen werden hier nur kurz vorgestellt. Für eine ausführlichere Beschreibung sei auf [56] verwiesen.

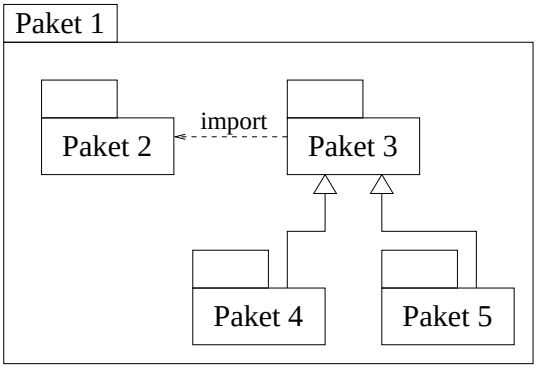


(a) Klasse

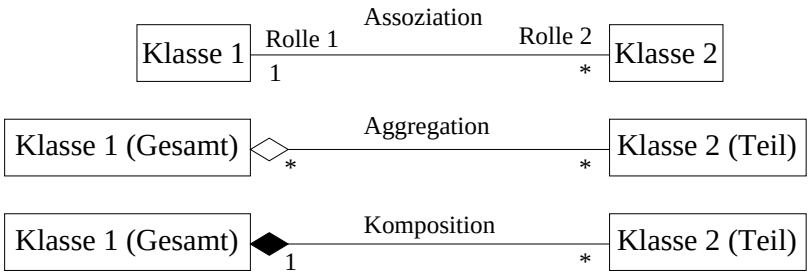


(b) Vererbung

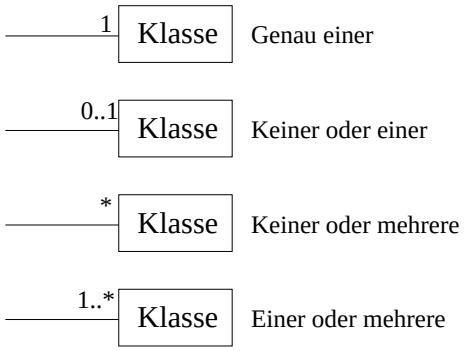
Abbildung E.1: UML Notation für Klassendiagramme



(c) Paket



(d) Assoziationen



(e) Kardinalitäten

Abbildung E.1: UML Notation für Klassendiagramme

Eine Klasse ist eine Beschreibung einer Gruppe von Objekten mit gemeinsamen Eigenschaften (Attributen), gemeinsamen Verhalten (Operationen), gemeinsamen Beziehungen zu anderen Objekten und gemeinsamer Seman-

tik. Mit Hilfe der Vererbung kann eine Klasse durch Erweiterung oder Einschränkung einer anderen Klasse definiert werden. Pakete sind Behälter für beliebige Modellelemente, mit denen das Gesamtmodell in kleinere überschaubare Einheiten gegliedert werden kann. Eine Assoziation beschreibt eine Beziehung zwischen Klassen bzw. deren Instanzen. Eine Aggregation ist eine besondere Form der Assoziation. Sie bezeichnet den Vorgang der Bildung eines Objektganzen durch Zusammensetzen anderer Objekte. Die Komposition ist eine strenge Form der Aggregation. Bei ihr sind die Teile abhängig von ihrem Aggregat, d.h. ohne das Ganze können die Teile nicht existieren. Dagegen kann das Aggregat auch ohne seine Teile existieren. Die Kardinalität einer Assoziation definiert, wie viele Objekte der einen Klasse ein bestimmtes Objekt einer anderen Klasse kennen.

Literaturverzeichnis

[1] Ahlberg, C. ; Wistrand, E.: IVEE: An Environment for Automatic Creation of Dynamic Queries Applications. In: *CHI '95 Proceedings* Chalmers University of Technology, Göteborg (Veranst.), 1995

[2] Albers, G. ; Guibas, Leonidas J. ; Mitchell, Joseph S. B. ; Roos, T.: Voronoi Diagrams of Moving Points. In: *Internat. J. Comput. Geom. Appl.* 8 (1998), S. 365–380

[3] Algorithmic Solutions Software GmbH: *LEDA: Library of Efficient Data Types and Algorithms*. – URL `http://www.algorithmic-solutions.de/leda.htm`

[4] Allen, R. B.: Retrieval from Facet Spaces. In: Brown, A. (Hrsg.) ; Brüggemann-Klein, A. (Hrsg.) ; Feng, A. (Hrsg.): *Proceedings of the 6th International Conference on Electronic Publishing, Document Manipulation and Typography*, September 1996

[5] Amsler, R.: Applications of citation-based automatic classification / Linguistics Research Center, Univ. Texas at Austin. Dezember 1972 (72-14). – Forschungsbericht

[6] Aurenhammer, F. ; Klein, R.: Voronoi Diagrams. In: Sack, J.R. (Hrsg.) ; Urrutia, J. (Hrsg.): *Handbook of Computational Geometry*. Amsterdam : Elsevier Science Publishers B.V. North-Holland, 2000, S. 201–290. – URL `http://wwwpi6.fernuni-hagen.de/Publikationen/tr198.pdf`

[7] Austrian Research Centers Seibersdorf: *BibTechMon*. – URL `http://www.arcs.ac.at/S/ST/BibTechMon`

[8] Bäumer, D. ; Riehle, D.: *Product Trader*. Kap. 3, S. 29–46. In: Martin, R. (Hrsg.) ; Riehle, D. (Hrsg.) ; Buschmann, F. (Hrsg.): *Pattern Languages of Program Design 3*, Addison-Wesley, 1998

[9] Baur, M. ; Benkert, M. ; Brandes, U. ; Cornelsen, S. ; Gaertler, M. ; Köpf, B. ; Lerner, J. ; Wagner, D.: Visone - Software for Visual Social Network Analysis. In: *Proc. 9th Intl. Symp. Graph Drawing (GD '01)*, Springer-Verlag, 2001 (Lecture Notes in Computer Science, LNCS 2265), S. 463 ff.

[10] Berners-Lee, T.: *Design Issues - Architectural and philosophical points*. 2001. – URL `http://www.w3.org/DesignIssues/Overview.html`. – W3C

[11] Bharat, K. ; Henzinger, M. R.: Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, 1998 (Distributed Retrieval), S. 104–111. – URL `http://www.acm.org/pubs/articles/proceedings/ir/290941/p104-bharat/p104-bharat.pdf`

[12] Bichteler, J. ; Eaton, E.: The combined use of bibliographic coupling and cocitation for document retrieval. In: *Journal of the American Society for Information Science* 31 (1980), Nr. 4, S. 278–282

[13] Bollacker, K. ; Lawrence, S. ; Giles, C. L.: CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications. In: Sycara, Katia P. (Hrsg.) ; Wooldridge, M. (Hrsg.): *Proceedings of the Second International Conference on Autonomous Agents*. New York : ACM Press, 1998, S. 116–123

[14] Bollacker, K. ; Lawrence, S. ; Giles, C. L.: A System For Automatic Personalized Tracking of Scientific Literature on the Web. In: *Proceedings of the Fourth Conference on Digital Libraries*. New York : ACM Press, 1999, S. 105–113

[15] Börner, K.: Extracting and Visualizing Semantic Structures in Retrieval Results for Browsing. In: *Proceedings of the 5th ACM Conference on Digital Libraries*, 2000

[16] Boyack, K. W. ; Wylie, B. N. ; Davidson, G. S.: Information Visualization, Human-Computer Interaction, and Cognitive Psychology: Domain Visualizations. In: *Proceedings of the 6th ACM Conference on Digital Libraries*, 2001

[17] Brandes, U. ; Wagner, D.: A Bayesian Paradigm for Dynamic Graph Layout. In: *Lecture Notes in Computer Science* 1353 (1997), S. 236–247

[18] Brandes, U. ; Willhalm, T.: Visualization of Bibliographic Networks with a Reshaped Landscape Metaphor. In: *Proc. 4th Joint Eurographics - IEEE TVCG Symp. Visualization (VisSym '02)*, ACM Press, 2002

[19] Branke, J.: Dynamic Graph Drawing. In: Kaufmann, M. (Hrsg.) ; Wagner, D. (Hrsg.): *Drawing Graphs, Methods and Models*, Springer, 2001, S. 228–246

[20] Branke, J. ; Bucher, F. ; Schmeck, H.: A Genetic Algorithm for Drawing Undirected Graphs. In: *Proc. 3rd Nordic Work. Genetic Algorithms and Their Applications, 3NWGA*, Finnish Artificial Intelligence Society, August 1997, S. 193–206

[21] Breu, M. ; Brüggemann-Klein, A. ; Endres, A.: Elektronische Informations- und Publikationsdienste für die Informatik: Ergebnisse des Projekts MeDoc. In: *Informatik '97*, Springer-Verlag, 1997, S. 235–245

[22] Brown, C.: *Visualising the Structure and Use of Large Scale Hypermedia Databases*, University of Nottingham, Dissertation, 1998

[23] Brüggemann-Klein, A. ; Klein, R. ; Landgraf, B.: BibRelEx: Exploring Bibliographic Databases by Visualization of Annotated Contents-Based Relations. In: *Proc. International Forum on Multimedia and Image Processing*, 1998

[24] Brüggemann-Klein, A. ; Klein, R. ; Landgraf, B.: BibRelEx: Erschließung bibliographischer Datenbasen durch Visualisierung von annotierten inhaltsbasierten Beziehungen. In: Lausen, G. (Hrsg.) ; Oberweis, A. (Hrsg.) ; Schlageter, G. (Hrsg.): *Angewandte Informatik und Formale Beschreibungsverfahren: Festschrift zum 60. Geburtstag von Wolffried Stucky*, B. G. Teubner, 1999, S. 33–44

[25] Brüggemann-Klein, A. ; Klein, R. ; Landgraf, B.: BibRelEx: Erschließung bibliographischer Datenbasen durch Visualisierung von annotierten inhaltsbasierten Beziehungen / FernUniversität Hagen. Juni 1999 (252). – Forschungsbericht

[26] Brüggemann-Klein, A. ; Klein, R. ; Landgraf, B.: BibRelEx: Exploring Bibliographic Databases by Visualization of Annotated Contents-Based Relations. In: *D-Lib Magazine* 5 (1999), Nr. 11

[27] Brüggemann-Klein, A. ; Klein, R. ; Landgraf, B.: BibRelEx: Exploring Bibliographic Databases by Visualization of Annotated

Contents-Based Relations. In: *Proc. Information Visualisation in Digital Libraries*, 2000

[28] Bruss, I. ; Frick, A.: Fast interactive 3-D graph visualization. In: *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95)* Bd. 1027, Springer-Verlag, 1995, S. 99–110

[29] Callan, J. P. ; Croft, W. B. ; Harding, S. M.: The INQUERY Retrieval System. In: *Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications*, 1992, S. 78–83

[30] Cameron, R. D.: A Universal Citation Database as a Catalyst for Reform in Scholarly Communication. In: *First Monday* 2 (1997), Nr. 4. – URL `http://www.firstmonday.dk/issues/issue2_4/cameron/index.html`

[31] Chakrabarti, S. ; Dom, B. ; Gibson, D. ; Kleinberg, J. ; Kumar, R. ; Raghavan, P. ; Rajagopalan, S. ; Tomkins, A.: Mining the Link Structure of the World Wide Web. In: *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, 1998

[32] Chakrabarti, S. ; Dom, B. ; Gibson, D. ; Kumar, S. R. ; Raghavan, P. ; Rajagopalan, S. ; Tomkins, A.: Experiments in Topic Distillation. In: *ACM SIGIR Workshop on Hypertext Information Retrieval on the Web*, 1998

[33] Chakrabarti, S. ; Dom, B. ; Raghavan, P. ; Rajagopalan, S. ; Gibson, D. ; Kleinberg, J.: Automatic resource compilation by analyzing hyperlink structure and associated text. In: *Proceedings of the 7th International World-Wide Web Conference*, 1998

[34] Chen, C.: *The StarWalker Virtual Environment.* Kap. 6.3, S. 189–198. In: *Information Visualisation and Virtual Environments*, Springer-Verlag, 1999

[35] Chen, C.: The StarWalker virtual environment - an integrative design for social navigation. In: *Proceedings of the Eighth International Conference on Human-Computer Interaction* Bd. 2, 1999, S. 207–211

[36] Chen, C.: Visualising Semantic Spaces and Author Co-Citation Networks in Digital Libraries. In: *Information Processing and Management* 35 (1999), S. 401–420

[37] Chen, C. ; Carr, L.: Trailblazing the Literature of Hypertext: Author Co-Citation Analysis (1989-1998). In: Tochtermann, K. (Hrsg.) ; Westbomke, J. (Hrsg.) ; Wiil, U.K. (Hrsg.) ; Leggett, J.J. (Hrsg.): *Proceedings of the Conference on Returning to Our Diverse Roots (Hypertext-99)*. New York, N.Y. : ACM Press, Februar 21–25 1999, S. 51–60. – ISBN 1-58113-064-3

[38] Chen, C. ; Paul, R. J.: Visualizing a Knowledge Domain's Intellectual Structure. In: *IEEE Computer* (2001)

[39] Cress, U. ; Barquero, B. ; Hesse, F. W.: Arbeitsbericht zum DFG-Projekt: Wissensaustausch mittels einer geteilten Datenbank / Institut für Wissensmedien, Tübingen. 2002. – Forschungsbericht

[40] Dalitz, W. ; Heyer, G.: *Hyper-G, Das Internet-Informationssystem der 2. Generation*. dpunkt Verlag, 1995

[41] Dässler, R. ; Palm, H.: *Virtuelle Informationsräume mit VRML. Informationen recherchieren und präsentieren in 3D*. dpunkt Verlag, 1998

[42] Davidson, R. ; Harel, D.: Drawing Graphs Nicely Using Simulated Annealing. In: *ACM Transactions on Graphics* 15 (1996), Nr. 4, S. 301–331

[43] Dean, J. ; Henzinger, M. R.: Finding related pages in the World Wide Web. In: *Computer Networks (Amsterdam, Netherlands: 1999)* 31 (1999), Mai, Nr. 11–16, S. 1467–1479. – URL `http://www.elsevier.com/cas/tree/store/comnet/sub/1999/31/11-16/2148.pdf`

[44] Defays, D.: An efficient algorithm for a complete link method. In: *The Computer Journal* 20 (1977), S. 364–366

[45] Delaunay, B.: Sur la sphère vide. A la memoire de Georges Voronoi. In: *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskih i Estestvennyh Nauk* 7 (1934), S. 793–800

[46] DeRose, S.J. ; Durand, D.G.: *Making Hypermedia Work: A User's Guide to HyTime*. Kluwer Academic Publishers, 1994

[47] Di Battista, G. ; Eades, P. ; Tamassia, R. ; Tollis, I. G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999

[48] Dreger, M. ; Lohrum, S. ; Schweppe, H. ; Ziegler, C. D.: Ariadne - Ein aktives Informationssystem für die Informatik. In: *Informatik '97, Workshop Multimediale Digitale Bibliotheken.* Aachen, September 1997

[49] Eades, P.: A Heuristic for Graph Drawing. In: *Congressus Numerantium* 42 (1984), S. 149–160

[50] Eades, P. ; Cohen, R.F. ; Huang, M. L.: Online animated graph drawing for web navigation. In: *Proceedings of the 5th International Symposium on Graph Drawing (GD'97).* Springer Lecture Notes in Computer Science 1353, 1997, S. 330–335

[51] Eibl, M. ; Mandl, T.: Die Qualität von Visualisierungen: Eine Methode zum Vergleich von zweidimensionalen Karten / Universität Hildesheim, FB Informations- und Kommunikationswissenschaften. 2001. – Forschungsbericht

[52] El-Hamdouchi, A. ; Willet, P.: Hierarchic document clustering using ward's method. In: *Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1986

[53] Englberger, H.: *Computergestützte Informationsvisualisierung*, Technische Universität München, Diplomarbeit, 1995

[54] Erickson, J.: *Biblook/bibindex 2.9.* – URL `http://compgeom.cs.uiuc.edu/~jeffe/biblook.html`

[55] Erickson, J. ; Jones, B. ; Schwarzkopf, O.: *More information about the database.* – URL `http://www.cs.duke.edu/~jeffe/compgeom/geombib/geombib_1.html`

[56] Erler, T.: *Das Einsteigerseminar UML.* bhv Verlag, 2000

[57] Fasulo, D.: An analysis of recent work on clustering algorithms / Department of Computer Science&Engineering, University of Washington. 1999. – Forschungsbericht

[58] Fellner, D. ; Kusserow, A. ; Schäfer, S.: Realisierung eines Nutzeragenten für den MeDoc-Dienst auf Basis von Hyper-G / Institut für Informatik III, Universität Bonn. 1996. – Forschungsbericht

[59] FIZ Karlsruhe: *WWW-Version of CompuScience.* – URL `http://www.zblmath.fiz-karlsruhe.de:80/cs`

[60] Fox, K. ; Frieder, O. ; Knepper, M. ; Snowberg, E.: SENTINEL: A Multiple Engine Information Retrieval and Visualization System. In: *Journal of the American Society for Information Science* 50 (1999), Nr. 7, S. 616–625

[61] Frick, A. ; Ludwig, A. ; Mehldau, H.: A fast adaptive layout algorithm for undirected graphs. In: *Proceedings of the DIMACS International Workshop on Graph Drawing (GD'94)*. Springer Lecture Notes in Computer Science 894, 1994, S. 388–403

[62] Fruchterman, T. M. J. ; Reingold, E. M.: Graph-Drawing by Force-directed Placement. In: *Software — Practice and Experience* 21 (1991), Nr. 11, S. 1129–1164

[63] FU Berlin: *Ariadne – Ein Navigations- und Suchsystem zu Informatik Informationen im Internet und ein Vermittlungsdienst zu Informatik-Fachinformationsdiensten*. 1996. – URL `http://ariadne.inf.fu-berlin.de:8000`

[64] Furnas, G. W.: The FISHEYE View: A New Look at Structured Files / Bell Laboratories. Murray Hill, New Jersey 07974, U.S.A., 12 Oktober 1981 (#81-11221-9). – Technical Memorandum. – URL `http://www.si.umich.edu/~furnas/POSTSCRIPTS/FisheyeOriginalTM.ps`

[65] Gamma, E. ; Helm, R. ; Johnson, R. ; Vlissides, J.: *Design Patterns*. Addison Weseley, 1995

[66] Garey, M. R. ; Johnson, D. S.: Crossing Number is NP-Complete. In: *SIAM J. Algebraic Discrete Methods* 4 (1983), Nr. 3, S. 312–316

[67] Garfield, E.: Citation indexes for science: a new dimension in documentation through association of ideas. In: *Science* 122 (1955), S. 108–111

[68] Garfield, E.: Citation Analysis as a Tool in Journal Evaluation. In: *Science* 178 (1972), Nr. 4060, S. 471–479

[69] Gibson, D. ; Kleinberg, J. ; Raghavan, P.: Inferring Web Communities from Link Topology. In: *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, 1998 (Structural Queries), S. 225–234

[70] Giles, C. L. ; Bollacker, K. ; Lawrence, S.: CiteSeer: An Automatic Citation Indexing System. In: Witten, Ian (Hrsg.) ; Akscyn, Rob (Hrsg.) ; III, Frank M. S. (Hrsg.): *Digital Libraries 98 - The Third*

ACM Conference on Digital Libraries. Pittsburgh, PA : ACM Press, June 23–26 1998, S. 89–98. – ISBN 0897919653

[71] Glance, N. ; Arregui, D. ; Dardenne, M.: Knowledge Pump: Supporting the Flow and Use of Knowledge. In: Borghoff, U. (Hrsg.) ; Pareschi, R. (Hrsg.): *Information Technology for Knowledge Management*, Springer Verlag, 1998

[72] Göbel, S.: *Literaturrecherche über Zitate*. 1996. – URL `http://www.math.fu-berlin.de/litrech/SCIndex/scindex.html`

[73] Göbel, S.: *Untersuchungen zur Mathematikliteratur in verschiedenen Datenbanken*. Oktober 1996. – URL `http://www.math.fu-berlin.de/~goebel/datenb.ps`

[74] Griffith, B. C. ; Small, H.: The structure of the scientific literatures I. Identifying and graphing specialities. In: *Science Studies* 4 (1974), S. 17–40

[75] Grossman, D. A. ; Frieder, O.: *Information Retrieval: Algorithms and Heuristics*. Kluwer Academic Publishers, 1998. – ISBN 0-7923-8271-4

[76] Guibas, Leonidas J. ; Mitchell, Joseph S. B. ; Roos, T.: Voronoi diagrams of moving points in the plane. In: *Proc. 17th Internat. Workshop Graph-Theoret. Concepts Comput. Sci.* Bd. 570, Springer-Verlag, 1992, S. 113–125

[77] Hadany, R. ; Harel, D.: A Multi-Scale Algorithm for Drawing Graphs Nicely / Weizmann Institute of Science, Faculty of Mathematics and Computer Sciences. Januar 1999 (CS99-01). – Technical Report

[78] Halasz, F. G. ; Moran, T. P. ; Trigg, R. H.: NoteCards in a Nutshell. In: *Conference Proceedings on Human Factors in Computing Systems and Graphics Interface (CHI/GI '87)*, 1987

[79] Halasz, F. G. ; Schwartz, M.: The Dexter Hypertext Reference Model. In: *Communications of the ACM* 37 (1994), Februar, Nr. 2, S. 30–39

[80] Hansen, K. M. ; Yndigegn, C. ; Grœnbæk, K.: Dynamic Use of Digital Library Material - Supporting Users with Typed Links in Open Hypermedia. In: *Proceedings of ECDL'99*, 1999

[81] Hendley, R. J. ; Drew, N. S.: *Visualisation of complex systems.* 1995

[82] Hendley, R. J. ; Drew, N. S. ; Wood, A. M. ; Beale, R.: Narcissus: Visualising Information. In: *Proceedings of IEEE Symposium on Information Visualisation*, URL `http://www.cs.bham.ac.uk/~amw/hyperspace/publications.html`, 1995, S. 90–96

[83] Hermann, M.: *Dokumentenexploration: Beziehungen analysieren und nutzen*, Universität Bonn, Institut für Informatik III, Diplomarbeit, 1999. – URL `http://www.informatik.uni-bonn.de/~myview/Diplomarbeiten/DA_Hermann.ps.gz`

[84] Hitchcock, S. ; Carr, L. ; Hall, W. ; Harris, S. ; Probets, S. ; Evans, D. ; Brailsford, D.: Linking electronic journals: Lessons from the Open Journal project. In: *D-Lib Magazine* (1998), Dezember. – URL `http://www.dlib.org/dlib/december98/12hitchcock.html`

[85] Hoppe, M.: *Erweiterung von Recherchemöglichkeiten für BibTEX-Datenbanken*, FernUniversität Hagen, Fachbereich Informatik, Diplomarbeit, 1999

[86] International DOI Foundation: *The Digital Object Identifier.* – URL `www.doi.org`

[87] International Organization for Standardization: *ISO Topic Maps Standard.* 1999. – URL `http://www.ornl.gov/sgml/sc34/document/0058.htm`

[88] Johnson, R. ; Woolf, B.: The Type Object Pattern. In: Martin, R. (Hrsg.) ; Riehle, D. (Hrsg.) ; Buschmann, F. (Hrsg.): *Pattern Languages of Program Design 3*, Addison-Wesley, 1998. – URL `http://www.awl.com/cseng/titles/0-201-31011-2`

[89] Jones, D. M.: *The Hypertext Bibliography Project.* 1995. – URL `http://theory.lcs.mit.edu/~dmjones/hbp/info.html`

[90] Kamada, T. ; Kawai, S.: An Algorithm for Drawing General Undirected Graphs. In: *Information Processing Letters* 31 (1989), S. 7–15

[91] Kaufmann, M. (Hrsg.) ; Wagner, D. (Hrsg.): *Drawing Graphs, Methods and Models.* Bd. *2025.* Springer, 2001. (Lecture Notes in Computer Science). – The book grow out of a Dagstuhl Seminar, April 1999. – ISBN 3-540-42062-2

[92] Kessler, M. M.: Bibliographic coupling between scientific papers. In: *American Documentation* 14 (1963), S. 10–25

[93] Klein, R.: Walking an Unknown Street with Bounded Detour. In: *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, 1991, S. 304–313

[94] Kleinberg, J.: Authoritative Sources in a Hyperlinked Environment. In: *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, Januar 1998, S. 668–677

[95] Kleinberg, J.: Authoritative Sources in a Hyperlinked Environment. In: *Journal of the ACM* 46 (1999), November, Nr. 5, S. 604–632

[96] Kleinberg, J. M. ; Kumar, S. R. ; Raghavan, P. ; Rajagopalan, S. ; Tomkins, A. S.: The Web as a Graph: Measurements, Models and Methods. In: Asano, T. (Hrsg.) ; Imai, H. (Hrsg.) ; Lee, D. T. (Hrsg.) ; Nakano, S. (Hrsg.) ; Tokuyama, T. (Hrsg.): *Proc. 5th Annual Int. Conf. Computing and Combinatorics, COCOON*, Springer-Verlag, Juli 1999 (Lecture Notes in Computer Science, LNCS 1627)

[97] Knuth, D. E.: *The Art of Computer Programming*. Bd. 3: Sorting and Searching. Reading, Massachussetts : Addison-Wesley, 1973

[98] Koch, M. ; Schönenberger, H. ; Galla, M.: Interoperable Community-Plattformen und Identitätsmanagement im Universitätsumfeld. In: Engelien, M. (Hrsg.) ; Homann, J. (Hrsg.): *Proc. Workshop GeNeMe2001 Gemeinschaften in Neuen Medien*, Josef Eul Verlag, Lohmar, 2001, S. 215–236. – URL `http://www11.in.tum.de/publications/pdf/Koch2001b.pdf`

[99] Kohonen, T.: Self-Organization of Very Large Document Collections: State of the Art. In: Niklasson, L. (Hrsg.) ; Bodén, M. (Hrsg.) ; Ziemke, T. (Hrsg.): *Proceedings of ICANN98, the 8th International Conference on Artificial Neural Networks* Bd. 1, Springer, 1998, S. 65–74

[100] Kopcsa, A. ; Schiebel, E.: Science and technology Mapping: A New Iteratio Model for Representing Multidimensional Relationships. In: *Journal of the American Society of Information Science* 49 (1998), Nr. 1, S. 7–17

[101] Krohn, U.: VINETA: Navigation through Virtual Information Spaces. In: *Proceedings of the Workshop on Advanced Visual Interfaces*, 1996, S. 49–58

[102] Kuhlen, R.: Moderation von elektronischen Foren bei netzbasierter Wissenskommunikation in einem virtuellen Wörterbuch / Universität Konstanz, Informationswissenschaft. URL `http://www.inf-wiss.uni-konstanz.de/People/RK/Texte/modfor.pdf`, 2000 (91-00). – Forschungsbericht

[103] Kumar, H. P. ; Plaisant, C. ; Schneiderman, B.: Browsing Hierarchical Data with Multi-Level Dynamic Queries and Pruning. In: *International Journal of Human-Computer Studies* (1995)

[104] Lamping, J. ; Rao, R. ; Pirolli, P.: A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In: *Chi '95 Proceedings* Xerox Palo Alto Research Center (Veranst.), 1995

[105] Lamport, L.: *LaTeX: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, 1994

[106] Landgraf, B.: *BibConsist: A Program to Check BibTeX Files for Inconsistencies*. 1997. – URL `http://wwwpi6.fernuni-hagen.de/wwwpi6/Forschung/BibRelEx/BibConsist.html.en#BIBCONS`

[107] Landgraf, B.: *Projekt BibRelEx: Objektmodell*. 1998

[108] Landgraf, B.: *Projekt BibRelEx: Pflichtenheft*. Februar 1998

[109] Landgraf, B.: 3D Graph Drawing. In: Kaufmann, M. (Hrsg.) ; Wagner, D. (Hrsg.): *Drawing Graphs, Methods and Models*, Springer, 2001, S. 172–192

[110] Landgraf, B. ; Handel, U.: *BibRelEx: Das Teilsystem Visual*. 2001

[111] Lasher, R. ; Cohen, D.: *A Format for Bibliographic Records*. 1995. – Request for Comments 1807

[112] Lawrence, S. ; Giles, C. L. ; Bollacker, K.: Digital Libraries and Autonomous Citation Indexing. In: *IEEE Computer* 32 (1999), Nr. 6, S. 67–71. – URL `http://www.neci.nj.nec.com/homepages/lawrence/papers/aci-computer98/aci-computer99.html`

[113] Lehner, B.: *KDE- und Qt-Programmierung*. Addison-Wesley, 2001

[114] Ley, M.: Die Trierer Informatik-Bibliographie DBLP / Universität Trier, FB 4. 1997. – Forschungsbericht

[115] Ley, M.: The ACM SIGMOD Anthology - a computer science retro-digitization project. In: *Proceedings of 7th Annual Meeting of the IuK Initiative Information and Communication of the Learned Societies in Germany, Cooperative Systems*, URL `http://www.zpid.de/iuk2001/program/abstracts/Ley.html`, 2001

[116] Liebl, A.: bibview: A graphical user interface to BibTEX. In: *TUGboat* 14 (1993), Dezember, Nr. 4, S. 390–395

[117] Liesenborghs, J.: *JThread Manual*. v1.0.0. : , 2001. – URL `http://lumumba.luc.ac.be/jori/jthread/manual.pdf`

[118] Lu, W. ; Janssen, J. ; Milios, E. ; Japkowicz, N.: Node Similarity in Networked Information Spaces / Faculty of Computer Science. September 2001 (CS-2001-03). – Forschungsbericht

[119] Lyardet, F.D.: Refining the Type Object Pattern: The Class Object Pattern. In: *The 4th Pattern Languagse of Programming Conference; Washington University Technical Report 97-34*, URL `http://www-lifia.info.unlp.edu.ar/~fer/classof.html`, 1997

[120] Lynch, J.: *Text Analysis with Compare*. 1995. – URL `http://www.english.upenn.edu/~jlynch/Computing/compare.html`. – Written for Stuart Curran's English 205/505, Electronic Literary Seminar

[121] Mackinlay, J. ; Robertson, G. ; Card, S.: The Perspective Wall: Detail and Context Smoothly Integrated. In: *Proceedings of the ACM SIGCHI '91 Conf.*, ACM Press, April 1991. – URL `http://www.xerox.com/PARC/dlbx/uir-papers/chi91-pw.ps`

[122] Mackinlay, J. D. ; Rao, R. ; Card, S. K.: An Organic User Interface For Searching Citation Links. In: *CHI '95 Proceedings* Xerox Palo Alto Research Center (Veranst.), 1995

[123] Marchiori, M.: The Quest for Correct Information on the Web: Hyper Search Engines. In: *Proceedings of the 6th International World-Wide Web Conference*, 1997, S. 265–276

[124] Marendy, P.: *A Review of Worl Wide Web searching techniques, focusing on HITS and related algorithms that utilise the link topology of the World Wide Web to provide the basis for a structure based search technology.* Juni 2001

[125] Maron, M. E. ; Kuhns, J. L.: On relevance, probabilistic indexing and information retrieval. In: *Journal of the ACM* 7 (1960), S. 216–244

[126] Marshakova, I.: A System of document connectionism based on references. In: *Nauchno-Tekhnicheskaya Informatsiya* 6 (1973), Nr. 2, S. 3–8. – (in Russian)

[127] Maurer, H. (Hrsg.): *HyperWave: The Next Generation Web Solution.* Reading, Massachusetts : Addison Wesley Longman, 1996. – ISBN 0-201-40346-3

[128] Max-Planck-Institut für Informatik: *AGD: A Library of Algorithms for Graph Drawing.* – URL `http://www.mpi-sb.mpg.de/AGD/`

[129] McCain, K. W.: Mapping authors in intellectual space: A technical overview. In: *Journal of the American Society of Information Science* 41 (1990), Nr. 6, S. 433–443

[130] Mehlhorn, K. ; Näher, S.: *LEDA: A Platform for Combinatorial and Geometric Computing.* Cambridge University Press, 2000

[131] Meyer, S.: *Natürliche Graphpartitionierung am Beispiel von Aufgabenmodellen in Unternehmensnetzwerken*, Universität Paderborn, Fachbereich Mathematik/Informatik, Diplomarbeit, 2000

[132] Michael, J.: Doppelgänger gesucht: Ein Programm für kontextsensitive phonetische Textumwandlung. In: *c't* 25 (1999)

[133] Mokotoff, G.: *Soundexing and Genealogy.* 1997. – URL `http://www.avotaynu.com/soundex.html`

[134] Monien, B. ; Ramme, F. ; Salmen, H.: A parallel simulated annealing algorithm for generating 3D layouts of undirected graphs. In: *Proceedings of the 3rd International Symposium on Graph Drawing (GD'95).* Springer Lecture Notes in Computer Science 1027, 1995, S. 396–408

[135] Mukherjea, S.: WTMS: a system for collecting and analyzing topic-specific Web information. In: *Computer Networks (Amsterdam, Netherlands: 1999)* 33 (2000), Juni, Nr. 1–6, S. 457–471. – URL `http://www9.org/w9cdrom/293/293.html`

[136] Mukherjea, S. ; Foley, J. D.: Visualizing the World-Wide Web with the Navigational View Builder. In: *Computer Networks and ISDN Systems* 27 (1995), April, Nr. 6, S. 1075–1087. – URL `http://www.elsevier.com/cas/tree/store/comnet/sub/1995/27/6/1447.pdf`

[137] Mukherjea, S. ; Foley, J. D. ; Hudson, S.: Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views. In: *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems* Bd. 1, ACM Press, 1995, S. 331–337. – URL `http://www.acm.org/sigchi/chi95/proceedings/papers/sm_bdy.htm`

[138] Mukherjea, S. ; Foley, J.D. ; Hudson, S.: Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views. In: *ACM SIGCHI '95*, ACM Press, Mai 1995. – URL `ftp://ftp.gvu.gatech.edu/pub/gvu/tech-reports/95-08.ps.Z`

[139] Nanard, J. ; Nanard, M.: Using Structured Types to Incorporate Knowledge in Hypertext. In: *Proceedings of Hypertext '91*, 1991

[140] Naumann, G.: *Ein System zur Verwaltung von benutzereigenen Datenbeständen als Ergänzung zu einer öffentlichen Literaturdatenbank unter periodischen Updates*, FernUniversität Hagen, Fachbereich Informatik, Diplomarbeit, 1996

[141] NEC Research Institute Inc. (NECI): *ResearchIndex: The NECI Scientific Literatur Digital Library.* – URL `http://www.neci.nec.com/~lawrence/researchindex.html`

[142] Neyer, G.: Map Labeling with Application to Graph Drawing. In: Kaufmann, M. (Hrsg.) ; Wagner, D. (Hrsg.): *Drawing Graphs, Methods and Models*, Springer, 2001, S. 172–192

[143] Niggemann, O.: *Visual Data Mining of Graph-Based Data*, University of Paderborn, Department of Mathematics and Computer Science, Dissertation, 2001. – URL `http://ubdata.uni-paderborn.de/ediss/17/2001/niggeman/`

[144] Noel, S.: *Data Mining and Visualization of Reference Associations: Higher Order Citation Analysis*, University of Louisiana, Lafayette, Dissertation, 2000

[145] Notthoff, M.: *Entwurf eines webbasierten Knowledge Building Environment für kooperatives Arbeiten*, Universität Dortmund, Diplomarbeit, 2000

[146] Obendorf, H.: *Vom Umgang mit XLinks - Konzepte für die Verwendung, Implementierung und Darstellung von XML Linking im Web*, Universität Hamburg, Diplomarbeit, 2001

[147] Offermanns, M.: *Visualisierung von Zitiergeflechten*, FernUniversität Hagen, Fachbereich Informatik, Diplomarbeit, 2000

[148] Oinas-Kukkonen, H.: Debate Browser – an argumentation tool for MetaEdit + environment. In: *Proceedings of the 7th European Workshop on Next Generation of CASE Tools (NGCT '96)*, 1996, S. 77–86

[149] Olsen, K. A. ; Korfhage, R. R. ; Sochats, K. M. ; Spring, M. B. ; Williams, J. G.: Visualization of a document collection with implicit and explicit links: The Vibe System. In: *Information Processing and Management* 29 (1993), Nr. 1, S. 69–81

[150] Page, L. ; Brin, S. ; Motwani, R. ; Winograd, Terry: The PageRank Citation Ranking: Bringing Order to the Web / Computer Science Department, Stanford University. URL `http://google.stanford.edu/~backrub/pageranksub.ps`, 1998. – Forschungsbericht

[151] Philips, L.: Hanging on the Metaphone. In: *Computer Language Magazine* 7 (1990), Nr. 12

[152] Philips, L.: The Double Metaphone Search Algorithm. In: *C/C++ Users Journal* 18 (2000), Nr. 6. – URL `http://www.cuj.com/articles/2000/0006/0006d/0006d.htm`

[153] Pirolli, P. ; Pitkow, J. ; Rao, R.: Silk from a Sow's Ear: Extracting Usable Structure from the Web. In: *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, ACM Press, 1996, S. 118–125. – URL `http://www.acm.org/sigchi/chi96/proceedings/papers/Pirolli_2/pp2.html`

[154] Pitkow, J. ; Pirolli, P.: Life, Death, and Lawfulness on the Electronic Frontier. In: *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, ACM Press, 1997, S. 383–390

[155] Popescul, A. ; Flake, G. ; Lawrence, S. ; Ungar, L. ; Giles, C.L.: Clustering and Identifying Temporal Trends in Document Databases. In: *Advances in Digital Libraries, ADL 2000*, 2000, S. 173–182

[156] Rada, R.: Hypertext Writing and Document Reuse: the Role of a Semantic Net. In: *Electronic Publishing: Organization, Dissemination and Distribution* 3 (1990), S. 3–13

[157] Raghupathi, W. ; Nerur, S.: Research Themes and trends in Artificial Intelligence: An Author CoCitation Analysis. In: *Intelligence* 10 (1999), Nr. 2

[158] Rijsbergen, C. J. V.: *Information Retrieval*. 2nd. Butterworths, 1979

[159] Robertson, G. ; Card, S. ; Mackinlay, J.: Information Visualization Using 3D Interactive Animation. In: *Communications of the ACM* 36 (1993), April, Nr. 4, S. 57–71. – URL `http://www.xerox.com/PARC/dlbx/uir-papers/iv-using-3d.ps`

[160] Robertson, G. ; Mackinlay, J. ; Card, S.: Cone Trees: Animated 3D Visuzalization of Hierarchical Information. In: *CHI '91 Proceedings*, April 1991, S. 189–194

[161] Röscheisen, M. ; Mogensen, C. ; Winograd, T.: Beyond Browsing: Shared Comments, SOAPs, Trails, and On-line Communities. In: *Third International World Wide Web Conference*, Elsevier Science Publishers B.V. North-Holland, 1995. – URL `http://www.igd.fhg.de/www/www95/proceedings/papers/88/TR/WWW95.html`

[162] Röscheisen, M. ; Mogensen, C. ; Winograd, T.: Shared Web Annotations as a Platform for Third-Party Value-Added Information Providers: Architecture, Protocols and Usage Examples / Computer Science Department, Stanford University. 1996 (CSDTR/DLTR). – Forschungsbericht

[163] Rupprecht, P.: *Evaluierung von Visualisierungssoftware für die Darstellung von Zitiergeflechten*, Fernuniversität Hagen, Fachbereich Informatik, Diplomarbeit, 1998

[164] Salton, G.: Associative Document Retrieval Techniques Using Bibliographic Information. In: *Journal of the ACM* 10 (1963), S. 440–457

[165] Salton, G.: *Automatic Information Organization and Retrieval*. New York : McGraw-Hill, 1968

[166] Salton, G.: Automatic Indexing Using Bibliographic Citations. In: *Journal of Documentation* 27 (1971), S. 98–110

[167] Salton, G. (Hrsg.): *The SMART retrieval system: experiments in automatic document processing*. Englewood Cliffs, US : Prentice-Hall, 1971. – 313–323 S

[168] Salton, G.: *Automatic Text Processing*. Addison-Wesley, 1989

[169] Salton, G. ; Wong, A.: Generation and Search of Clustered Files. In: *ACM Transactions on Database Systems* (1978), Dezember

[170] Sarkar, M. ; Brown, M. H.: Graphical Fisheye Views. In: *Communications of the ACM* (1994), Dezember

[171] Schickler, M. A. ; Mazer, M. S. ; Brooks, C.: Pan-Browser Support for Annotations and Other Meta-Information on the World Wide Web. In: *Proceedings of the 5th International World-Wide Web Conference*, Elsevier Science Publishers B.V. North-Holland, 1996. – URL `http://www5conf.inria.fr/fich_html/papers/P15/Overview.html`

[172] Schmidt, J. P.: *BibManage: Ein Tool zur Verwaltung von privaten und öffentlichen Versionen einer BibTEX-Literaturdatenbank*, Fern-Universität Hagen, Fachbereich Informatik, Diplomarbeit, 1999

[173] Shamos, M. I. ; Hoey, D.: Closest-Point Problems. In: *Proc. 16th Annu. IEEE Sympos. Found. Comput. Sci.*, 1975, S. 151–162

[174] Sim, S.: *Automatic Graph Drawing Algorithms*. 1996

[175] Singer, N.: *Science landscape may aid US intelligence services, funding agencies, researchers, and eventually libraries*. 1996. – URL `http://www.sandia.gov/LabNews/LN10-11-96/land.html`. – Sandia LabNews

[176] Singer, N.: *Science landscape may aid US intelligence services, funding agencies, researchers, and eventually libraries*. 1996. – URL `http://www.sandia.gov/media/mapping.htm`. – News Release

[177] Sleepycat Software Inc.: *Berkeley DB*. New Riders, 2001

[178] Small, H.: Co-citation in the scientific literature: a new measure of the relationship between scientific documents. In: *Journal of the American Society for Information Science* 24 (1973), S. 265–269

[179] Small, H.: Macro-Level Changes in the Structure of Co-Citation Clusters: 1983-1989. In: *Scientometrics* 26 (1993), Nr. 1, S. 5–20

[180] Small, H.: Update on Science Mapping: Create Large Document Spaces. In: *Scientometrics* 38 (1997), S. 275–293

[181] Smith, L. C.: Citation Analysis. In: *Library Trends* 30 (1981), Nr. 1

[182] Sokal, R. R. ; Michener, C. D.: A Statistical Method for Evaluating Systematic Relationships. In: *University of Kansas Science Bulletin* 38 (1958), S. 1409–1438

[183] Stein, B. ; Niggemann, O.: On the Nature of Structure and its Identification. In: *Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science*, Springer-Verlag, 1999 (Lecture Notes Comput. Sci.). – URL http://www.oliver-niggemann.de/work/publications/wg99.pdf

[184] Streitz, N. ; Haake, J. ; Hannemann, J. ; Lemke, A. ; Schuler, W. ; Schütt, H. ; Thüring, M.: SEPIA: cooperative hypermedia authoring environement. In: *Proceedings of the 3rd ACM Conference on Hypertext (HYPERTEXT '92)*, 1992, S. 11–22

[185] Stroustrup, B.: *The C++ Programm Language*. 3rd. Addison Weseley, 1997

[186] Sugiyama, K. ; Misue, K.: Drawing Graphs by Magnetic-Spring Model. In: *Journal on Visual Languages and Computing* 6 (1995), Nr. 3

[187] Sugiyama, K. ; Tagawa, S. ; Toda, M.: Methods for Visual Understanding of Hierarchical Systems. In: *IEEE Trans. Syste. Man Cybern.* SMC-11 (1981), Nr. 2, S. 109–125

[188] Sutter, H.: *Exceptional C++*. 1st. Addison-Wesley, 2000

[189] Terveen, L. ; Hill, W.: Finding and Visualizing Inter-Site Clan Graphs. In: *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems* Bd. 1, ACM Press, 1998, S. 448–455. – URL http://www.acm.org/pubs/articles/proceedings/chi/274644/p448-terveen/p448-terveen.pdf

[190] The VRML Consortium Incorporated: *The Virtual Reality Modeling Language*. 1997. – URL http://www.vrml.org/Specifications/VRML97/

[191] Thiel, S.: *LyberWorld—Eine 3D-Schnittstelle zur Exploration von Information*. 1995

[192] Trigg, R. H.: *A Netwwork-Based Approach to Text Handling*, University of Maryland, Dissertation, November 1983

[193] Tutte, W. T.: How to Draw a Graph. In: *Proc. London Mathematical Society* 13 (1963), S. 743–768

[194] Universität Konstanz: *Visone - analysis and visualization of social networks*. – URL http://www.visone.de/

[195] Voorhees, E. M.: Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. In: *Information Processing and Management* 22 (1986), Nr. 6, S. 465–476

[196] Voronoi, G. M.: Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième Mémoire: Recherches sur les parallélloèdres primitifs. In: *J. Reine Angew. Math.* 134 (1908), S. 198–287

[197] Wang, W. ; Rada, R.: Experiences with semantic net based hypermedia. In: *International Journal of Human-Computer Studies* 43 (1995), S. 419–439

[198] White, H. D. ; McCain, K. W.: Visualizing a Discipline: An Author Co-Citation Analysis of Information Science, 1972-1995. In: *Journal of the American Society for Information Science* 49 (1998), S. 327–355

[199] Wise, J. ; Thomas, J. ; Pennock, K. ; Lantrip, D. ; Pottier, M. ; Schur, A. ; Crow, V.: Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In: *Proceedings of Information Visualization ' 95 Symposium*, 1995, S. 51–58

[200] Wistrand, E.: Visualization Methods for Dynamic Queries Databases / Göteborg University. 1995 (SSKKII-95.01). – Forschungsbericht

[201] Wolff, J. E. ; Cremers, A. B.: The MYVIEW Project: a Data Warehousing Approach to Personalized Digital Libraries. In: *Proc. of the 4th Int. Workshop on Next Generation Information Technologies and Systems (NGITS'99)*, Springer-Verlag, 1999, S. 277–294. – URL `http://www.informatik.uni-bonn.de/~jw/publikationen/NGITS99_short_version.ps.gz`

[202] Wood, A. ; Beale, R. ; Drew, N. ; Hendley, B.: HyperSpace: A World-Wide Web Visualiser and its Implications for Collaborative Browsing and Software Agents. In: *Third International World-Wide Web Conference*, Elsevier Science B.V., 1995, S. 21–25

[203] Woodruff, A. ; Gossweiler, R. ; Pitkow, J. ; Chi, E. H. ; Card, S. K.: Enhancing a Digital Book with a Reading Recommender. In: *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems*, ACM Press, April 1–6 2000, S. 153–160. – URL `http://www.acm.org/pubs/articles/proceedings/chi/332040/p153-woodruff/p153-woodruff.pdf`

[204] Zedlitz, J.: Aus Schönberg wird ZÖNBAK: Moderne Soundexverfahren. In: *Computergenealogie: Magazin für Familienforschung* (2001). – Newsletter Nr. 7/2001

[205] Zedlitz, J.: Computer lernen besser hören: Verfeinerte Version des phonetischen Algorithmus Soundex. In: *Computergenealogie: Magazin für Familienforschung* (2001). – Newsletter Nr. 5/2001

[206] Zedlitz, J.: Wenn Computer hören lernen: Der phonetische Algorithmus Soundex. In: *Computergenealogie: Magazin für Familienforschung* (2001). – Newsletter Nr. 3/2001