

Institut für Visualisierung und Interaktive Systeme (VIS)

Bachelorarbeit Nr. 165

Webbasierte visuelle Analyse der Verbreitungswege historischer Abschriften

Marco Loureiro

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Thomas Ertl
Betreuer/in:	Dr. Steffen Koch, Dipl.-Inf. Tanja Blascheck
Beginn am:	28. August 2014
Beendet am:	24. Februar 2015
CR-Nummer:	H.5.2, J.5, H.3.5

Kurzfassung

Die Analyse von Verbreitungswegen historischer Texte ist ein Forschungsgebiet der Historiker. Diese Historiker versuchen herauszufinden, welche Texte von welchen Quellen abgeschrieben oder überliefert wurden. Dazu zählt auch herauszufinden, welche Textsammlungen, die heute nicht mehr existieren, als Zwischenstationen gedient haben. Diese wissenschaftliche Arbeit beschreibt die Entwicklung eines webbasierten Tools zur Darstellung und Analyse von Verbreitungswegen von Briefen Papst Gregors VII. Dazu werden die zu Grunde liegenden Daten in Form einer XML-Datenbank mit Hilfe von JavaScript und der Bibliothek D3 ausgelesen und visuell dargestellt. Die XML-Datenbank enthält Informationen über die Kollektionen, in denen die Briefe enthalten sind. Es sind auch Hypothetische Kollektionen enthalten, von denen man ausgeht, dass sie existiert haben. Ein weiterer Teil der XML-Datenbank enthält Informationen über die Briefe, die in diesen Kollektionen enthalten sind. Hierzu werden unter anderem Daten über die Verlässlichkeit der Briefe, ob der Brief nachweislich ein Brief aus dieser Zeit ist, die Kollektionen die diese enthalten und den Platz in der Kollektion angegeben. Um eine einfache Verfolgung der Verbreitung zu gewährleisten, werden die Kollektionen in einem Knoten-Kanten Graphen angezeigt, der in Schichten unterteilt ist. Mithilfe von Interaktion können die Knoten, also die Kollektionen, ausgeklappt werden und die Briefe eingesehen werden. In ähnlicher Weise sind die Verbindungen zwischen den Kollektionen zusammengefasst. Die Art der Verbindungen kann zwischen Schichtkanten und Übersichtskanten umgeschaltet werden, um einen Überblick ohne Kantenkreuzungen zu ermöglichen. Die Überlegungen und Ansätze, die zu dieser Lösung geführt haben, werden im Folgenden präsentiert. Dabei werden zuerst einige Grundlagen für das Verständnis der Arbeit eingeführt. Im Anschluss werden verwandte Arbeiten vorgestellt, um diese Arbeit von bisherigen Tools abzugrenzen. Nachfolgend wird das Konzept hinter dieser Arbeit aufgezeigt und die konkrete Implementierung dieses Konzepts dargelegt. Zum Abschluss wird das Ergebnis dieser Arbeit im Bezug auf die vorher bestehende Visualisierung abgewogen.

Inhaltsverzeichnis

1	Einleitung	9
2	Grundlagen	11
2.1	Knoten-Kanten Graphen	11
2.2	Graphen Layouts	12
2.3	Heuristiken	15
2.4	Baryzentrische Heuristik	16
2.5	Sugiyama Algorithmus	17
2.6	Kollektionen	21
3	Verwandte Arbeiten	23
3.1	Layout	23
3.2	Interaktion	26
4	Konzept	27
4.1	Anforderungsanalyse	27
4.2	Layout	29
4.3	Kanten Probleme	30
4.4	Interaktion	31
4.5	Farbgestaltung	32
5	Implementierung	35
5.1	XML Datenbank	35
5.2	Geschichtetes Layout	36
5.3	Kantenführung	37
5.4	Briefe	38
5.5	Öffnen von Kollektionen	38
6	Ergebnis	41
7	Zusammenfassung und Ausblick	43
	Literaturverzeichnis	45

Abbildungsverzeichnis

2.1	Gerichteter Graph	12
2.2	Ungerichteter Graph	12
2.3	Haarball	13
2.4	Hierarchisches Layout	14
2.5	Radiales Layout	15
2.6	Kräftebasierendes Layout	16
2.7	Baryzentrische Heuristik	17
2.8	Sugiyama Startgraph	18
2.9	Sugiyama Zyklenentfernung	19
2.10	Sugiyama Schichtzuweisung	19
2.11	Sugiyama Kantengruppierung	20
2.12	Sugiyama Kantensortierung	21
2.13	Sugiyama X-Koordinaten	21
3.1	<i>Class Blueprint</i>	24
3.2	SHriMP	24
3.3	SHriMP Spring Layout	25
3.4	Circos	25
3.5	SHriMP Details	26
4.1	Kollektionen Typen	28
4.2	Erdmann Stammbaum	28
4.3	Konzept Übersicht	29
4.4	Übersichtskanten	30
4.5	Initiale Ansicht	31
4.6	Geöffnete Kollektion	32
4.7	Briefauswahl	33
6.1	Ursprüngliche Visualisierung	41

Verzeichnis der Listings

5.1	Fragment der XML-Datenbank eines Kollektionen XML-Eintrags	35
5.2	Fragment der XML-Datenbank eines Brief XML-Eintrags	36
5.3	Berechnung baryzentrischer Wert	37

1 Einleitung

Ein wichtiger Bereich der Informationsvisualisierung ist die visuelle Analyse. In diesem Bereich geht es darum, durch Computer visuell dargestellte Sachverhalte durch einen menschlichen Verstand analysieren zu lassen. Hier wird der Vorteil von Computern, Daten schnell verarbeiten und darstellen zu können, mit dem Vorteil des menschlichen Verstandes verbunden, Muster in einer solchen Visualisierung erkennen zu können. Der Gegenstand dieser Arbeit ist es, eine Visualisierung für eine solche Analyse von Verbreitungswegen historischer Texte bereitzustellen.

Im Themengebiet der Analyse historischer Texte spielen deren Verbreitungswege eine wichtige Rolle. Dabei wird oft gefragt: woher kommt ein Dokument und welche Kollektionen liegen zwischen zwei bekannten Kollektionen? In dieser Arbeit geht es dabei speziell um die Verbreitungswege der Briefe von Papst Gregor VII. Zu diesen Briefen bestehen bereits Vermutungen und Tatsachen, was Verbreitungswege bestimmter Briefe angeht. Diese sollen nun mittels einer Visualisierung dargestellt werden, die dem Benutzer auch erlaubt, Kollektionen in dieser Visualisierung genauer zu analysieren. Dazu wird in dieser Arbeit im folgenden Kapitel „Grundlagen“ zunächst verschiedene Begriffe eingeführt, die für das weitere Verständnis der Arbeit wichtig sind. Im Anschluss werden in Kapitel 3 „Verwandte Arbeiten“ ähnliche Tools, zu dem in dieser Arbeit entwickelten Visualisierung, vorgestellt und Unterscheidungsmerkmale analysiert. In Kapitel 4 „Konzepte“ werden die grundlegenden Konzepte hinter der Visualisierung vorgestellt. Daraufhin wird in Kapitel 5 „Implementierung“ die konkrete Umsetzung der Konzepte in ein Programm präsentiert. In Kapitel 6 „Ergebnis“ wird ein kurzer Vergleich mit einer ursprünglichen Visualisierung zu diesem Themengebiet vorgenommen. Zum Schluss wird in Kapitel 7 „Zusammenfassung und Ausblick“ eine Zusammenfassung der Arbeit gegeben und mögliche Verbesserungen zur entwickelten Visualisierung vorgestellt.

2 Grundlagen

In diesem Kapitel werden Begriffe und Algorithmen eingeführt, die für den weiteren Verlauf der Arbeit benötigt werden. Dabei werden zuerst Knoten-Kanten Graphen, zur Einführung in dieses Themengebiet, und die verschiedenen Layouts dieses Graphentyps vorgestellt. Dies dient dem Überblick über die verschiedenen Layouts eines Knoten-Kanten Graphen. Im Anschluss werden Heuristiken allgemein und die baryzentrische Heuristik eingeführt, da diese im weiteren Verlauf der Arbeit zur Berechnung verwendet wird. Im letzten Teil der Grundlagen wird der Sugiyama Algorithmus zum Berechnen eines hierarchischen Layouts aufgezeigt, da dieser ein Vorreiter im Gebiet der hierarchischen Layouts war.

2.1 Knoten-Kanten Graphen

Ein Knoten-Kanten Graph G wird definiert als die Menge $G = (V, E)$, wobei V die Menge der Knoten und E die Menge der Kanten bezeichnet [Tam13]. Die Knoten stellen hierbei die zu untersuchenden oder darzustellenden Objekte dar. Die Kanten bezeichnen die Verbindungen, die zwischen diesen Objekten bestehen. Die Anzahl der Knoten wird als $|V| = n$ und die Anzahl der Kanten als $|E| = m$ definiert. Im Rahmen der Definition von Kanten muss man zwischen gerichteten und ungerichteten Graphen unterscheiden. Bei einem gerichteten Graph besteht die Verbindung zwischen zwei Knoten nur in eine Richtung, d.h. eine Kante $e_1 = (v_1, v_2) : e_1 \in E, v_1, v_2 \in V$ beschreibt eine Verbindung vom Knoten v_1 zum Knoten v_2 wie in Abbildung 2.1, allerdings besteht dadurch keine Verbindung vom Knoten v_2 zum Knoten v_1 . Dies müsste durch eine zusätzliche Kante $e_2 = (v_2, v_1)$ definiert werden. Die Kanten in einem gerichteten Graphen sind also ein geordnetes Tupel [Tam13]. Anders verhält sich dies bei ungerichteten Graphen. Hier würde es ausreichen, wie in Abbildung 2.2 zu sehen, die Kante e_1 zu definieren, um eine Verbindung von v_1 nach v_2 und umgekehrt festzulegen. Bei ungerichteten Graphen werden Kanten also als ungeordnetes Tupel [Tam13] dargestellt.

In einem Knoten-Kanten Graph hat jeder Knoten einen sogenannten Ausgangs- und Eingangsgrad. Der Ausgangsgrad beschreibt die Anzahl der Kanten, die den Knoten verlassen, während der Eingangsgrad die Anzahl der Kanten beschreibt, die in den Knoten gehen. Bei ungerichteten Kanten ist der Eingangsgrad äquivalent zum Ausgangsgrad. Weiterhin bezeichnet man mit $N(v)$ die Nachbarschaft vom Knoten v , also alle mit v direkt verbundenen Knoten.

Mit Knoten-Kanten Graphen kann der Benutzer leicht Aufgaben erfüllen, die etwas mit der Verfolgung von Pfaden zu tun hat. zum Beispiel die Analyse eines Flusses, also welche Knoten transitiv miteinander verbunden sind. Allerdings ist dies nur bis zu einem gewissen Grad möglich. Die Skalierbarkeit solcher Graphen ist nicht sonderlich hoch. So kann es bei einem dichten Graphen, also einem Graph mit vielen Kanten, mit wenigen Knoten (<100) schon zu einem sogenannten „Haarball“ [GFC05] kommen. Dies bezeichnet einen Graphen, dessen Kanten eine so hohe Anzahl an Kreuzungen und

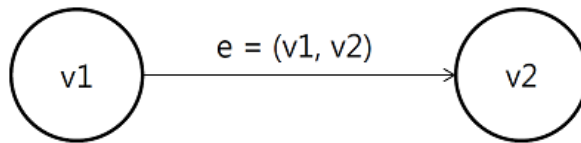


Abbildung 2.1: Gerichteter Graph mit Kante $e = (v_1, v_2)$. Ein gerichteter Graph bedeutet, dass die Definition der Kante e eine Verbindung vom Knoten v_1 zum Knoten v_2 aufbaut. Die Verbindung v_2 zu v_1 besteht dadurch allerdings nicht. Die Kanten werden in gerichteten Graphen also als geordnetes Tupel definiert.

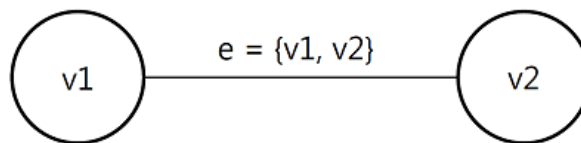


Abbildung 2.2: Ung gerichteter Graph mit Kante $e = \{v_1, v_2\}$. Anders als im gerichteten Fall, bedeutet die Definition der Kante e hier eine Verbindung vom Knoten v_1 zum Knoten v_2 **UND** andersrum. Das bedeutet, dass die Kanten in ungerichteten Graphen als ungeordnetes Tupel, also als Menge, definiert werden.

Überdeckungen hat, dass man keine pfadbezogenen Aufgaben lösen kann, wie in Abbildung 2.3 zu sehen. Bei einem Graph mit vielen Knoten kann man ebenso auf Probleme stoßen, zum Beispiel ist es bei einer hohen Zahl von Knoten schwer, alle Knoten in einem Bildschirm darzustellen, sodass eine Analyse in diesem Fall kaum durchzuführen ist.

2.2 Graphen Layouts

Zur Darstellung von Knoten-Kanten Graphen gibt es verschiedene Layouts, das heißt verschiedene Berechnungsarten der Knotenpositionen. Diese Layouts lassen sich grundsätzlich in drei Hauptgruppen einordnen:

- hierarchische/geschichtete Layouts
- radiale/zirkuläre Layouts
- kräftebasierende Layouts

Diese werden im Folgenden vorgestellt.

2.2.1 Hierarchisches/geschichtetes Layout

Die erste Darstellungsform von Graphen ist ein hierarchisches, auch geschichtetes, Layout. Dabei werden die Knoten wie in Abbildung 2.4 entsprechend ihrer hierarchischen Einordnung in horizontale

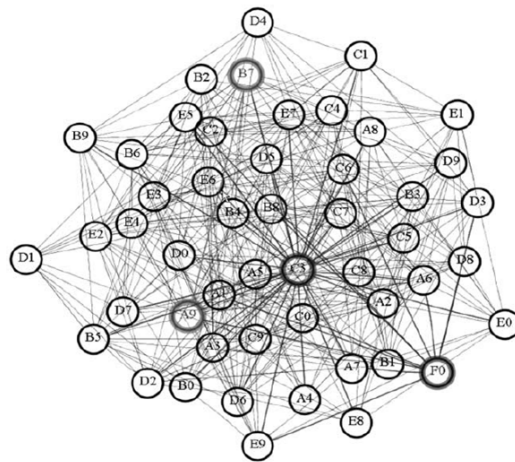


Abbildung 2.3: Dies ist ein Graph aus Ghoniem et al. [GFC05], an dessen Beispiel die schlechte Skalierbarkeit von Knoten-Kanten Graphen dargestellt wird. Dieses Bild hat den Namen „Haarball“ für einen solchen Graphen geprägt. Der Name stammt daher, dass die hohe Anzahl an Kanten und Kantenkreuzungen zu einer starken Überdeckung führen. Dadurch kann man in diesem Graphen keine einzelnen Verbindungen mehr erkennen, wodurch der Eindruck eines Haarballs entsteht.

Schichten eingeteilt. Im Falle einer Einordnung von oben nach unten stehen die Knoten der ersten Hierarchieebene auch in der obersten Schicht der Darstellung. Dadurch kann dieses Layout schön Abläufe darstellen und intuitiv an den Benutzer übermitteln. Das Problem an dieser Intuitivität ist allerdings, dass Knoten in einer unteren Schicht unwichtiger vorkommen, da sie unterhalb anderer Knoten liegen, was durch die Daten möglicherweise nicht intendiert ist. Dadurch könnte der Benutzer ein falsches Bild von diesen dargestellten Daten bekommen. Ein weiteres Problem von geschichteten Layouts ist ihre ineffiziente Platzausnutzung. Dies wird vor allem zum Problem, wenn es im gegebenen Datensatz eine volle Ebene in der Hierarchie gibt. Diese wird dann im geschichteten Layout breit und könnte zu Schwierigkeiten mit der Darstellbarkeit auf dem Bildschirm führen. Zum Beispiel könnte die Zahl der Knoten in einer Ebene so groß sein, dass es nicht möglich ist diese auf einer Bildschirmbreite darzustellen, ohne die Knoten einer anderen Ebene so klein machen zu müssen, dass diese nicht mehr zu erkennen sind. Selbst wenn dies möglich ist, wird in den Ebenen mit wenigen Knoten sehr viel Platz verschwendet.

2.2.2 Radiales/zirkuläres Layout

Eine weitere Darstellungsart von Graphen ist die radiale oder auch zirkuläre Repräsentation. Hier werden die Knoten ausgehend von einem Mittelpunkt auf einen einzelnen oder auf mehrere konzentrische Kreise platziert. Für kleinere, weniger dichte Graphen eignet sich ein einzelner Kreis zur Darstellung, bei dichteren Graphen müssen die Knoten auf mehrere konzentrische Kreise verteilt werden. Bei einem einzelnen Kreis zur Darstellung werden benachbarte Knoten möglichst nahe

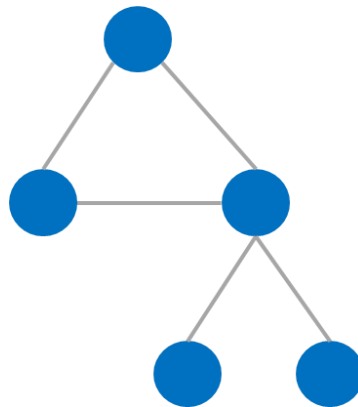


Abbildung 2.4: Beispiel für ein Hierarchisches Layout. Hier kann man drei horizontale Schichten des Layouts identifizieren. Je nach Lage in der Hierarchie werden die Knoten in die Schichten eingeordnet. Die Kanten werden dann zwischen die dazugehörigen Knoten eingezeichnet. Hier ist die oberste Schicht also die oberste Ebene der Hierarchie, je weiter die Schicht unten liegt, desto tiefer steigt man auch in der Hierarchie des darzustellenden Datensatzes hinab.

beieinander auf dem Kreis platziert, um lange Kanten und Kantenkreuzungen zu verhindern. Dieses Layout hat die Eigenschaft, dass Knoten durch die Platzierung in einem Kreis ohne zentrales Element alle gleich wichtig wirken, wodurch der Nutzer alle Knoten gleich gründlich untersucht, da kein zentrales Element den Fokus von den anderen Knoten nimmt. Dies hat eine Neutralitätseigenschaft der Knoten zur Folge.

Im Falle mehrerer Kreise wird bei jedem Schritt in der Hierarchie der Radius des Kreises vergrößert. Dadurch entsteht eine Darstellung wie in Abbildung 2.5, die im Zentrum die oberste Hierarchieebene repräsentiert und nach außen gehend die Hierarchie nach unten durchläuft. Allerdings geht bei dieser Abänderung die Eigenschaft der Neutralität verloren, da die Knoten auf verschiedene Radien verteilt werden. Außerdem ist die Laufzeit für die Berechnung einer radialen Layouts niedrig, von Tamassia et al. [Tam13] wird ein Algorithmus von Six und Tollis [ST99][ST06] gezeigt, der ein solches Layout mit einem Kreis in Laufzeit $O(m)$ berechnen kann und dadurch linear von der Eingabegröße der Kanten abhängt. Aber auch diese Layouttechnik hat Schwächen. Bei großen Datensätzen kann es schwer sein, den zentralen Knoten des Layouts zu finden. Außerdem kann bei einer ungünstigen Verteilung auf den Kreisen eine große Anzahl von Kantenkreuzungen und unter anderem lange Kanten entstehen, welche die Lesbarkeit des Darstellung stark beeinträchtigen. Um dies zu verhindern werden benachbarte Knoten auch auf dem Kreis benachbart platziert. Weiterhin wird in den Algorithmen, wie von Tamassia et al. in [Tam13] beschrieben, versucht durch Umordnen der Knoten Kantenkreuzungen zu verhindern.

2.2.3 Kräftebasierendes Layout

Die dritte Gruppierung von Knoten-Kanten Graphdarstellungen bilden die auf Kräften basierenden Layouts. Hierbei werden die Knotenpositionen von physikalischen Kräften bestimmt [VLKS⁺11], die

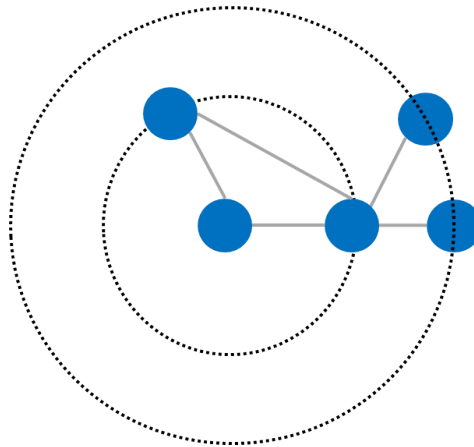


Abbildung 2.5: Der Graph aus Abbildung 2.4 im radialen Layout dargestellt. Die gestrichelt gezeichneten Kreise dienen zur Veranschaulichung der radialen Anordnung der Knoten. Diese werden mit absteigender Hierarchie um einen Kreis nach außen gehend eingefügt, dadurch wird jeweils eine Ebene der Hierarchie pro Kreis eingefügt. Dadurch liegt im Zentrum der Kreise der einzelne Ursprungsknoten der Hierarchie, falls dieser existiert. Wenn die erste Hierarchie mehrere Knoten enthält, wird diese auf den ersten Kreis aufgetragen und der Mittelpunkt bleibt leer.

auf die Knoten und die Kanten angewendet werden. Ein erster solcher Algorithmus wurde von Eades [Ead84] vorgeschlagen. Dabei werden den Knoten Kräfte zugeteilt, die sich gegenseitig abstoßen, während die Kantenenden Kräfte erhalten, die sich anziehen, zum Beispiel Federkräfte. Dadurch entsteht eine Repräsentation des Graphen, in welcher direkt verbundene Knoten einen festen Abstand haben, während nicht verbundene Knoten sich gegenseitig abstoßen, wie beispielhaft in Abbildung 2.6 zu sehen. Hier wurden die Knotenkräfte auf eins festgelegt, während die Anziehung durch die Länge der Kante repräsentiert wird.

Da die Berechnung der Knotenpositionen durch Kräfte, also durch ein physikalisches Modell, aufgebaut wird, ist es sehr intuitiv. Durch die bestehenden Kräfte in einem solchen Layout ist möglich, dieses dynamisch zu gestalten. Somit passt sich der Rest des Layout beispielsweise an, wenn man einen Knoten in der Darstellung verschiebt. Diese Dynamik muss allerdings durch eine hohe Laufzeit erkauft werden. Ein von Harel und Sardas [HS98] verbesserter Algorithmus, der von Tamassia et al. in [Tam13] vorgestellt wird und versucht keine Kantenkreuzungen zu erzeugen, benötigt eine Laufzeit von $O(n^3)$.

2.3 Heuristiken

Heuristiken berechnen Lösungen, die nicht immer optimal sind, sich dieser aber annähern [Wei07]. Dafür haben sie den Vorteil, dass sie einen geringeren Rechen- und Laufzeitaufwand benötigen, als nicht-heuristische Berechnungen. Daher werden sie besonders häufig dort eingesetzt, wo die optimale

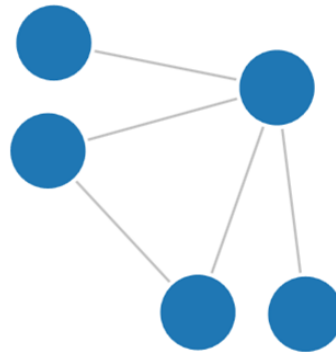


Abbildung 2.6: Der Graph aus Abbildung 2.4 in einem Kräfte basierenden Layout. In diesem Layout werden den Knoten Kräfte zugeordnet, wodurch sich Knoten gegenseitig abstoßen, die Startpositionen der Knoten sind zufällig. Die Kantenenden erhalten dahingegen anziehende Kräfte. Dadurch entsteht eine Darstellung des Graphen, in dem Knotenpaare mit einer Kante zusammen bleiben und Knotenpaare ohne Kanten sich mit der zugeordneten Kraft abstoßen. In diesem Beispiel haben die Knoten eine abstoßenden Kraft vom Wert eins, während die Anziehungskraft durch die Länge der Kanten repräsentiert wird. Wenn man die abstoßenden Kräfte erhöhen würde, wären die Knotenpaare mit einer Kante im gleichen Zustand, während die anderen Knoten weit auseinander geschoben werden würden.

Lösung einer Fragestellung nicht unbedingt notwendig ist, um damit die nötigen Ressourcen zur Berechnung der Lösung eines Algorithmus zu verringern.

2.4 Baryzentrische Heuristik

Die baryzentrische Heuristik von Sugiyama et al. [STT81] wird von Bastert und Matuszewski in [BM01] als gute Variante vorgestellt, um Kantenkreuzungen in einem Graphen mit geschichtetem Layout zu verringern. Dazu wird angenommen, dass es ,um wenige Kantenkreuzungen zu haben, von Vorteil ist, verbundene Knoten nahe beieinander zu platzieren. Dazu wird für jeden Knoten der baryzentrische Wert ausgerechnet, der sich aus den X-Koordinaten ergibt. Daraus ergibt sich folgende Formel, die den Durchschnitt aller X-Koordinaten der Nachbarn eines Knoten berechnet, nach dem die Knoten dann sortiert werden. Der Knoten u befindet sich in der Ebene V_2 , die Knoten v in der Ebene V_1 :

$$bary(u) = \frac{1}{deg(u)} * \sum_{v \in N(u)} \pi_1(v)$$

Hierbei beschreibt $deg(u)$ den Ausgangsgrad des Knoten u , $N(u)$ die Nachbarn von u , also alle mit u verbundenen Knoten, und $\pi_1(v)$ die Permutation der Knoten v in der Schicht V_1 , das heißt die aktuelle Anordnung aller Knoten v in dieser Schicht. In Abbildung 2.7 ist das Schema für eine baryzentrische Heuristik abgebildet. Diese Heuristik ist besonders praktisch, da für ihre Berechnung keine Matrix

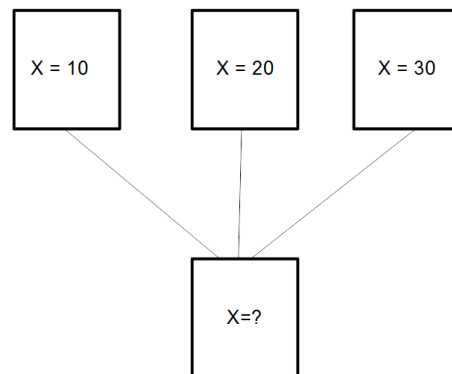


Abbildung 2.7: Diese Abbildung stellt das Schema zur Berechnung einer baryzentrischen Heuristik dar. Die drei Kästen oben haben hier feste Koordinaten. Die X-Koordinate des unteren Kastens soll berechnet werden. Mit Hilfe der Formel

$$bary(u) = \frac{1}{deg(u)} * \sum_{v \in N(u)} \pi_1(v)$$

kann man nun die X-Koordinate berechnen. In diesem Beispiel ist $deg(u) = 3$, da der untere Kasten den Grad drei besitzt. Die Summe ergibt sich aus den bestehenden X-Koordinaten der anderen Kästen, also

$$\sum_{v \in N(u)} \pi_1(v) = 10 + 20 + 30 = 60.$$

Daraus folgt für die X-Koordinate des unteren Kastens: $bary(u) = \frac{1}{3} * 60 = 20$.

für die Anzahl der Kantenkreuzungen errechnet werden muss, was für viele andere Heuristiken für diesem Zweck berechnet werden muss, weil die Anzahl der Kantenkreuzungen für die Berechnung der Heuristik keine Rolle spielen. Dadurch wird bei dieser Heuristik Laufzeit eingespart, sodass inklusive der Umsortierung der Knoten nach der Berechnung eine Laufzeit von $O(|V_2| * \log|V_2|)$ [KW01], wobei $|V_2|$ für die Anzahl der Knoten in der Schicht V_2 steht. Außerdem ist noch anzumerken, dass diese Heuristik eine Lösung ohne Kantenkreuzungen findet, wenn solch eine Lösung existiert.

2.5 Sugiyama Algorithmus

Die Algorithmen zur Berechnung geschichteter Layouts werden häufig als Sugiyama Stil Algorithmen bezeichnet, benannt nach Kozo Sugiyama, dem Entwickler des ersten Algorithmus dieser Art [STT81]. Diese Algorithmen sind aus 5 Teilschritten aufgebaut, die von Tamassia et al. [Tam13] beschrieben werden:

- Zyklenentfernung
- Schichtzuordnung
- Kantengruppierung

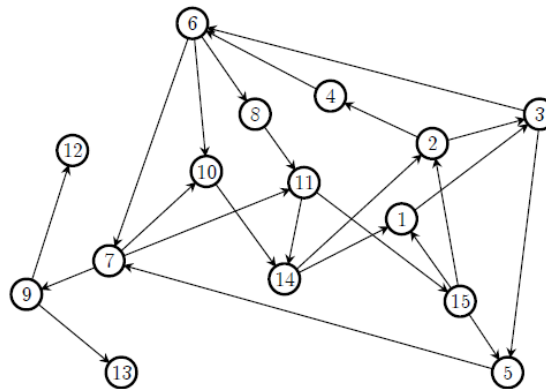


Abbildung 2.8: Dies ist der Startgraph aus [Tam13] der für die weite Erläuterung der Schritte, mit Ausnahme der Kantengruppierung, verwendet wird. Dieser Graph ist ein gerichteter Graph mit Zyklen.

- Knotensortierung
- Zuweisung X-Koordinaten

Diese Schritte werden im Folgenden anhand des in Abbildung 2.8 zu sehenden Graphen verdeutlicht, mit Ausnahme des Kantengruppierung Schrittes, der optional ist.

Im ersten Schritt werden enthaltene Zyklen im Graph entfernt, dies kann durch Umkehrung der Richtung einer Zyklus verursachenden Kante oder durch deren Löschung aus dem Graphen geschehen, zu sehen in Abbildung 2.9.

Der zyklenlose Graph wird für den zweiten Schritt des Algorithmus benötigt, die Einordnung der Knoten in verschiedene Schichten. Mit Zyklen wäre diese Einordnung nicht eindeutig, da man in einem Graph mit Zyklen nicht eindeutig festlegen kann, wo der Ursprung oder das Ende eines Kantenverlaufs ist. In diesem Schritt werden die Knoten in ihre jeweilige Schicht eingeordnet. Die Schicht wird durch die Rolle der Knoten in der Hierarchie festgelegt. So kann zum Beispiel ein Knoten der nur ausgehende Kanten hat in der obersten Schicht eingeordnet werden, wohingegen ein Knoten mit ausschließlich eingehenden Kanten in die unterste Schicht einsortiert werden kann. Der Rest der Knoten wird ausgehend von diesen eingefügten Knoten aufgebaut. Dabei wird darauf geachtet, dass Kanten möglichst nur benachbarte Schichten miteinander verbinden. Falls dies für eine Kante nicht möglich ist, werden sogenannte „Dummieknoten“ in die übersprungenen Schichten eingefügt, die als Zwischenknoten dienen, um diese Eigenschaft zu erhalten, dieser Vorgang kann in Abbildung 2.10 beobachtet werden.

Nach dem Einfügen der Knoten können die im ersten Schritt gelöschten Kanten wieder eingefügt und die umgekehrten wieder umgedreht werden. Diese Kanten sind der Richtung der restlichen Kanten entgegengesetzt, die durch die Einteilung der Knoten in Schichten entstanden ist. Das Wiedereinfügen der Kanten kann aber auch im letzten Schritt erfolgen, wie im gegebenen Beispiel zu sehen in

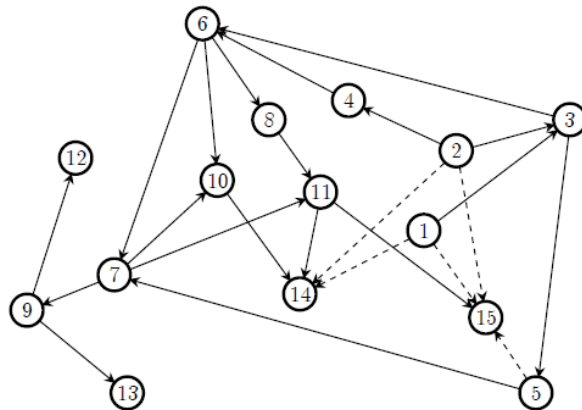


Abbildung 2.9: Dies ist der zyklensfreie Startgraph aus [Tam13]. Zur Entfernung der Zyklen wurden die zyklerzeugenden Kanten umgedreht, diese werden hier als gestrichelten Kanten dargestellt.

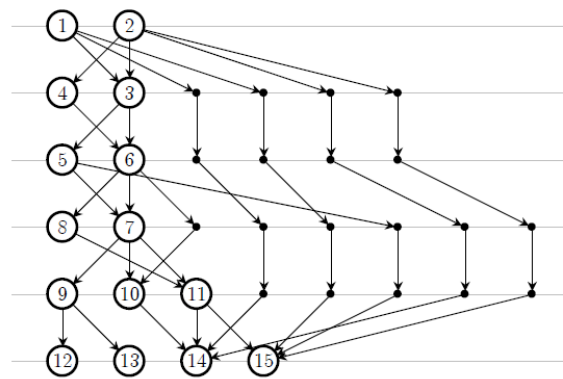


Abbildung 2.10: In dieser Abbildung sieht man die Knoten, nummerierte Kreise, und „Dummieknoten“, schwarze Punkte, für die Schichtzuordnung des Algorithmus aus [Tam13]. Man kann erkennen, dass die Dummieknoten den Kanten eine bessere Route durch den Graphen vorgeben, als die direkte Verbindung der realen Knoten.

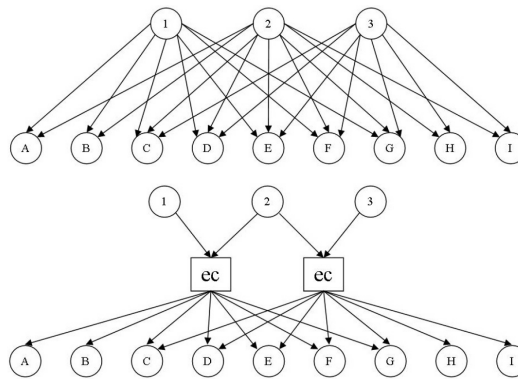


Abbildung 2.11: Diese Abbildung stellt den Graphen im Kantengruppierungsschritt aus [Tam13] dar. Oben ist der Graph ohne Kantengruppierung zu sehen. Unten drunter sind die beiden Kantengruppierungsknoten „ec“ eingefügt. Man kann die Verbesserung der Darstellung durch die verkürzten Kanten und die dadurch verringerten Kantenkreuzungen erkennen. Da dieser Schritt optional ist, wird er im zusammenhängenden Beispiel der anderen Abbildungen nicht angewandt.

Abbildung 2.13

Als optionaler Schritt kann daraufhin eine Kantengruppierung („Edge Concentration“) vorgenommen werden. Dazu wird eine zusätzlich Schicht an der gewünschten Stelle eingefügt, die mit mit sogenannten Kantengruppierungsknoten („Edge Concentrators“) gefüllt werden. Diese werden dazu genutzt, die Darstellung übersichtlicher zu gestalten, wie es in Abbildung 2.11 gemacht wurde.

Im Anschluss werden die Knoten in ihren Schichten in ihrer Reihenfolge umgeordnet, zu sehen in Abbildung 2.12. Dies geschieht zur Minimierung der Kantenkreuzungen im zu erzeugenden Graphen. Die Anzahl der Kantenkreuzungen spielt eine wichtige Rolle für die Lesbarkeit des Graphen, da viele Kreuzungen unter anderem die Verfolgung von Pfaden erschweren. Ein Layout zu finden, in dem die minimale Anzahl von Kantenkreuzungen auftritt, ist ein NP hartes Problem, weshalb die Algorithmen hier auf einen heuristischen Ansatz zurückgreifen. Eine praktische Heuristik ist für diese Anwendung zum Beispiel die baryzentrische Heuristik, siehe Abschnitt 2.4.

Als letzten Schritt im Algorithmus wird für jeden Knoten noch die X-Koordinate berechnet. Diese wird wenn möglich so gewählt, dass der Knoten mittig zwischen den Knoten liegt, mit denen er eine Verbindung hat. Die positive Auswirkung die daraus entsteht, dass ein Knoten in Mitte seiner Nachbarn liegt, ist, dass man dadurch möglichst gerade Kanten zwischen den Knoten erhält, wie in Abbildung 2.13. Diese geraden Kanten vereinfachen das Verfolgen von Pfaden.

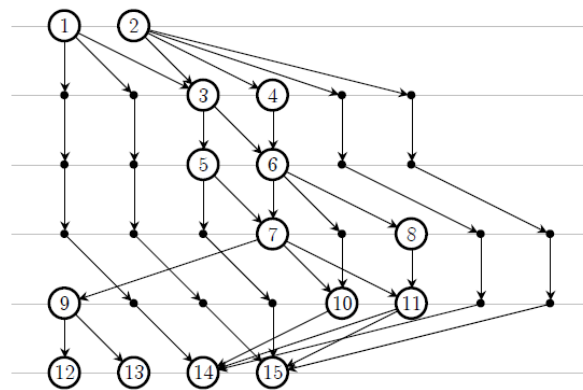


Abbildung 2.12: Diese Abbildung stellt den Graphen im nach dem Umsortieren der Knoten aus [Tam13] dar. Man kann erkennen, dass zum Beispiel die Knoten drei und vier in der Reihenfolge im Vergleich zum Schichtzuweisungsschritt vertauscht wurden.

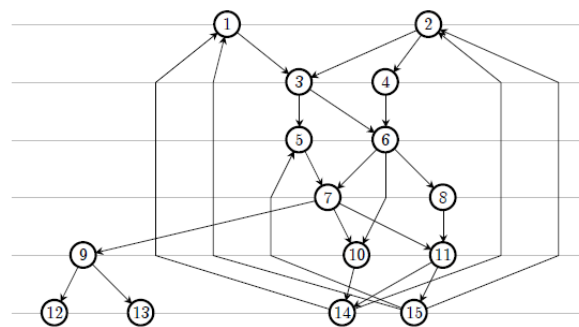


Abbildung 2.13: Der gezeigte Graph aus [Tam13] befindet sich hier am Ende des Algorithmus. Man kann erkennen, dass die Knoten sich an anderen X-Koordinaten befinden, als in den Schritten zuvor, um die Kanten möglichst gerade und frei von Kantenkreuzungen zu halten. Außerdem kann man zum Beispiel die Kante (15,1) erkennen, die wieder in die ursprüngliche Richtung umgekehrt wurde und somit gegen die Hauptrichtung des Graphen zeigt.

2.6 Kollektionen

Kollektionen beschreiben die Sammlung von Briefen in dieser Arbeit. Dabei wird zwischen hypothetischen Kollektionen und realen Kollektionen unterschieden. Reale Kollektionen sind Kollektionen, die bekanntermaßen existiert haben. Dahingegen sind hypothetische Kollektionen solche, von denen nicht nachgewiesen wurde, dass sie existiert haben. Es ist nur soviel sicher, dass es zwischen zwei realen Kollektionen eine Lücke gibt, in der eine Kollektion existiert haben muss. Eine hypothetische Kollektion ist also eine Theorie über eine Kollektion, die möglicherweise diese Lücke füllen kann.

3 Verwandte Arbeiten

3.1 Layout

Im Bezug auf das Layout dieser Arbeit können verschiedene Ansätze unterschieden werden, die sich auf das Layout beziehen. Auf die verschiedenen Anwendungsgebiete der vorgestellten Tools soll hier nicht eingegangen werden.

Das CodeCrawler Tool von Lanza et al. [LDGP05] verwendet in der sogenannten *Class Blueprint* Ansicht, wie die Visualisierung dieser Arbeit, ebenfalls ein hierarchisches Layout zur Darstellung von Softwaremetriken. Diese Ansicht ist, den Autoren zufolge, die meist bekannte und stellt Metriken, wie zum Beispiel Vererbung von Klassen, dar. Genaueres zum *Class Blueprint* kann in [DL05] nachgelesen werden. In CodeCrawler werden dabei Hierarchien von Klassen angezeigt, wobei Unterklassen unterhalb von Oberklassen liegen. Dadurch entsteht eine Art geschichtetes Layout, in dem Klassen und ihre Details, wie Methoden, dargestellt werden. Anders als in dieser Arbeit werden die Klassen, also die Elemente einer Schicht, weiter in interne Schichten unterteilt. Dies ist in dieser Arbeit nicht notwendig, da es innerhalb der Elemente einer Schicht keine weitere Unterscheidungen gibt, die eine interne Schichtung begründen könnte. Eine Beispielformalisierung kann man in Abbildung 3.1 sehen.

Genau zu erkennen sind hier die zwei Schichten, Schicht A mit dem großen Rechteck, Schicht B mit den drei kleineren darunter. Jedes dieser Elemente ist weiter in vertikale Schichten unterteilt.

Einen ähnlichen Ansatz verwendet auch das Tool SHriMP (*Simple Hierarchical Multi-Perspective*). Hier wird wie bei CodeCrawler ein hierarchisches Layout zur Darstellung von Untersystemen wie Klassen verwendet. Anders als bei CodeCrawler und dieser Arbeit, werden die verschiedenen Details allerdings in einzelne Fenster ausgelagert. Wie in Abbildung 3.2 zu sehen, wird im obersten Fenster (A) ein Überblick geboten. Die restlichen Fenster im roten Kasten stellen jeweils Details von ausgewählten Elementen dar.

Einen anderen Layout Ansatz stellen die Autoren von SHriMP ebenfalls in [SM95] als alternatives Layout vor. Dieser Ansatz verwendet das *Spring* Layout von Eades aus [Ead84] um die Subsysteme in einem kräftebasierenden Layout wie auf der linken Seite in Abbildung 3.3 zu sehen.

Den letzten Ansatz, der hier in Betracht gezogen wird, ist der von Krzywinski et al. im Tool Circos [KSB⁺09] verwendete zirkuläre Ansatz. Dieses Tool wurde ursprünglich dafür entwickelt, genomische Zusammenhänge darzustellen, ist allerdings auf viele andere Themengebiete anwendbar. Hier werden die Elemente äquidistant auf Kreise um einen Mittelpunkt platziert. Ein Beispiel für eine Visualisierung mittels Circos in einem anderen Themengebiet als Genome, kann in Abbildung 3.4 gesehen werden. Diese Abbildung stammt aus [AAWN13] und stellt dar, welche Internetseiten auf andere verweisen. Das Kriterium ist hierbei die Sprache der verlinkenden und der verlinkten Seite.

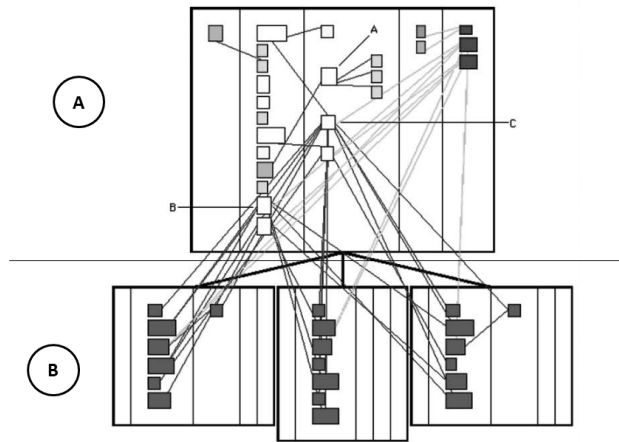


Abbildung 3.1: In dieser beispielhaften Visualisierung einer Hierarchie mit drei Unterklassen aus [DL05] kann man die zwei horizontalen Schichten erkennen, in die die vier Klassen eingeordnet wurden. Die Oberklasse ist in der ersten Schicht (A), die Unterklasse in der zweiten (B). Jede Klasse ist weiterhin in vertikale Schichten aufgeteilt. Diese stellen verschieden Stadien in der Aufrufreihenfolge dar [DL05].

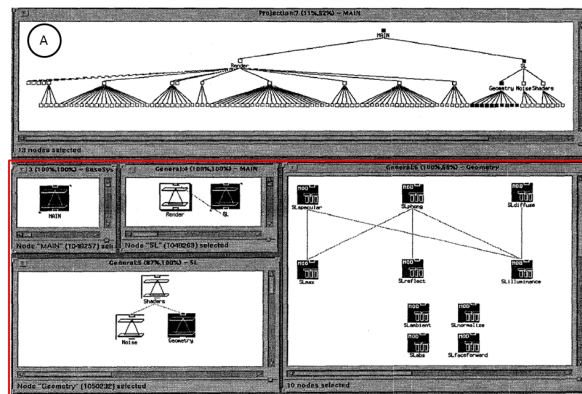


Abbildung 3.2: Beispielhafte Visualisierung eines Subsystems mittels SHriMP aus [SM95]. Die längliche Ansicht (A) stellt den Überblick über das Subsystem dar. Die anderen Fenster im roten Kasten stellen jeweils die Detailansicht zu geöffneten Unterknoten des Systems dar, die tieferegehende Analysen ermöglichen.

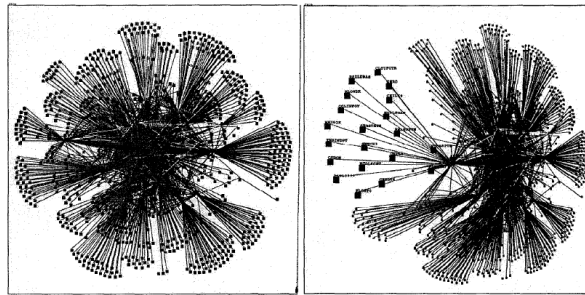


Abbildung 3.3: In dieser Visualisierung eines Untersystems mittels SHriMP aus [SM95] mit einem kräftebasierenden Layout kann man links den erzeugten Graph erkennen. Auf der rechten Seite wurden mittels Interaktion einige Knoten geöffnet, um Details wie Labels darzustellen.

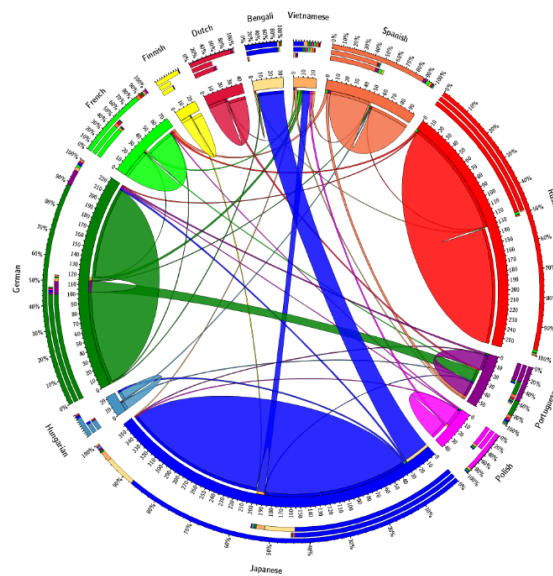


Abbildung 3.4: Diese Visualisierung stammt von AlNoamany et al. aus [AAWN13] und stellt dar, welche Internetseiten in einer gewissen Sprache auf Internetseiten in einer anderen oder gleichen Sprache verweisen. Dazu liegen auf dem Kreis die verschiedenen Sprachen, die analysiert wurden. Die Verbindungen zeigen die Verlinkungen der jeweiligen Sprache an.

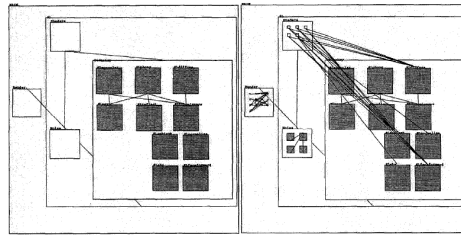


Abbildung 3.5: In dieser Abbildung aus [SM95] sieht man die zwei Ansichten eines Subsystems. Links sind noch keine Subsysteme geöffnet. Auf der rechten Seite wurden einige Subsysteme geöffnet, um weitere Details anzuzeigen.

Der kräftebasierende und der zirkuläre Layoutansatz sind für diese Arbeit allerdings nicht geeignet, da die gegebenen Daten eine Hierarchie von oben nach unten darstellen. Diese Begebenheit kann durch die beiden Ansätze nicht intuitiv an den Benutzer vermittelt werden. Zu diesem Zweck eignet sich das hierarchische beziehungsweise geschichtete Layout weit mehr.

3.2 Interaktion

In Abbildung 3.5 von Storey et al. ist auf der linken Seite ein Subsystem mit geschlossenen Details zu sehen. Auf der rechten Seite wurden die drei kleinen Rechtecke, die weitere Subsysteme darstellen, geöffnet, um weitere Details anzuzeigen [SM95]. Dieses Prinzip wird auch in dieser Arbeit angewandt, wenn mittels Interaktion die gewünschten Kollektionen geöffnet werden sollen, um die Details, also die Briefe, anzuzeigen. Das gleiche Prinzip kann auch auf der rechten Seite in Abbildung 3.3 beobachtet werden, da mittels Öffnung des Knoten neue Details angezeigt werden, wie hier die Labels der Knoten.

Die Interaktionsmöglichkeit wird auch in dieser Arbeit verwendet, da man durch das Verstecken von Details dem Benutzer einiges an überflüssigen Informationen und visueller Überladung ersparen kann. Zusätzlich kann man dadurch einen guten Überblick über die gesamte Größe der Daten geben. Der Unterschied zwischen der Interaktion von [SM95] und der dieser Arbeit liegt allerdings darin, dass in dieser Arbeit die Kanten der übergeordneten Elemente ausgeblendet werden. Dies wird in der Interaktion von Storey et al. nicht getan, dies ist wohl durch das unterschiedliche Einsatzgebiet der Tools bedingt. In dieser Arbeit spielen die übergeordneten Verbindungen keine Rolle mehr, wenn Details angeschaut werden. In einem Softwaresystem könnten diese Verbindungen allerdings noch wichtig sein, wodurch man diese auch bei der Anzeige von Details beibehält.

4 Konzept

In diesem Teil der Arbeit werden die grundlegenden Konzepte hinter der später folgenden Implementierung vorgestellt. Dazu werden zuerst die Ergebnisse einer Anforderungsanalyse aufgezeigt. Im Anschluss wird das Konzept des grundsätzlichen Layouts der Visualisierung vorgestellt. Im weiteren Verlauf werden Konzepte zur Lösung verschiedener Probleme des verwendeten Layouts dargelegt und das Konzept der Interaktion in dieser Arbeit eingeführt. Zum Abschluss werden die Hintergründe hinter der Farbgestaltung der Visualisierung aufgezeigt.

4.1 Anforderungsanalyse

Zur Erstellung des hier beschriebenen Tools wurden vor Beginn der Programmierarbeit die Anforderungen an das Tool zusammengestellt. Dabei wurden folgende Anforderungen aufgestellt:

- A1: XML-Datenbank einlesen
- A2: Überblick auf Kollektionsebene
- A3: Hypothetische Kollektionen kennzeichnen
- A4: Analyse der Briefverbindungen
- A5: Kollektion einfügen
- A6: Geeignetes Interaktionskonzept
- A7: Klassische Ansicht nach Erdmann [Erd36, S.27]
- A8: Export als XML

Die Daten die in diesem Werkzeug visualisiert werden, sind in einer XML-Datenbank abgespeichert. Diese Datenbank wird zu Beginn eingelesen (A8) und die extrahierten Informationen werden zur Darstellung genutzt. Da das Tool für die Analyse von Verbreitungswegen päpstlicher Briefe zwischen Kollektionen genutzt werden soll, ist es zunächst einmal wichtig, dem Benutzer einen Überblick (A2) über die zur Analyse zur Verfügung stehenden Kollektionen zu geben. Dabei ist es wichtig, diesen Überblick innerhalb einer Bildschirmbreite darzustellen, um dem Benutzer unpraktisches scrollen nach rechts zu ersparen. Innerhalb der Kollektionen wird zwischen solchen Kollektionen unterschieden (A3), die real existiert haben, Abbildung 4.1 a), oder hypothetisch erstellt wurden, Abbildung 4.1 b). Um etwaige Theorien innerhalb der Visualisierung zu testen, ist eine Funktion zum einfügen neuer Kollektionen (A5) sinnvoll. Durch dieses Feature wird es dem Analyst vereinfacht, sich in der Visualisierung zurechtzufinden. Dafür sorgen zusätzlich Verbindungen auf Kollektionsebene, die anzeigen welche Kollektionen über Briefe überhaupt in Verbindung stehen. Um die Analyse der

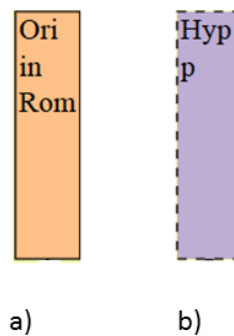


Abbildung 4.1: In dieser Abbildung kann man das Konzept zur Darstellung der zwei zu unterscheidenden Kollektionentypen sehen. Es wird zwischen realen Kollektionen, dargestellt wie a), und hypothetischen Kollektionen, dargestellt wie b), unterschieden. Der gestrichelte Rand der hypothetischen Kollektion stellt die Unsicherheit darüber dar, ob die Kollektion wirklich existiert hat. Die qualitative Farbgebung dient zur Unterscheidung zweier Gruppen von Kollektionen ohne diese zu werten.

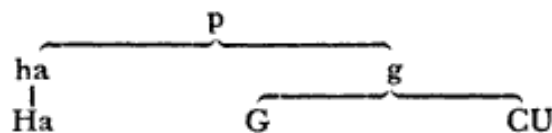


Abbildung 4.2: Stammbaum Darstellung von Erdmann aus [Erd36, S.27]. Dieses Schema gilt als eine Art Norm für die visuelle Darstellung von Verbreitungswegen im Themenbereich der Analyse von Verbreitungswegen. Das Schema stellt dar, dass die Sammlungen *ha* und *g* von *P* abstammen, da *ha* und *g* Teile aus *P* enthalten, aber durch weitere Teile ergänzt wurden [Erd36]. Die Kollektion *ha* wurde durch *Ha* und die Kollektion *g* durch *G* und *CU* ergänzt.

Briefe zu unterstützen, werden dann die Verbreitungspfade (A4) zwischen den Briefen angezeigt. Diese Verbreitungspfade werden mit Hilfe der extrahierten Angaben aus der XML Datenbank unter dem Punkt „ff“ bei den Briefen definiert. Dies kennzeichnet den sogenannten *fons formalis* (im Folgenden *ff*), was die Kopiervorlage des betrachteten Briefes beschreibt. Um diese Visualisierung im Stile von Erdmanns Stammbaum in [Erd36, S.27] zu halten (A7), der in Abbildung 4.2 dargestellt ist, wird das Layout hierarchisch ausgelegt.

Um dem Benutzer zu erlauben, die neu eingefügten Theorien und Änderungen abzuspeichern, ist es sinnvoll, diese Erkenntnisse wieder als XML-Datenbank zu exportieren (A8).

Um dem Benutzer stets eine übersichtliche Darstellung anzubieten, werden die verschiedenen Ansichten per Interaktion (A6) angezeigt und auch wieder versteckt.

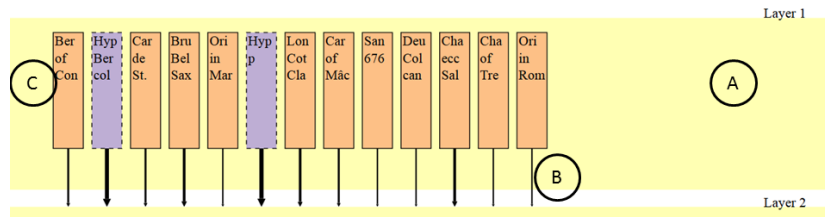


Abbildung 4.3: In dieser Abbildung sind die drei wichtigsten Teile des Überblicks abgebildet. Ein Teil der Visualisierung sind die Schichten (A), in die Kollektionen (C) einordnet werden. Die sogenannten „Schichtkanten“ (B) dienen dazu, einen Überblick über die Menge der weitergegebenen Briefe zu liefern.

4.2 Layout

Da die Visualisierung in Form einer Knoten-Kanten Darstellung erfolgen soll, um im Stile von Erdmanns Stammbaum zu bleiben, fallen andere Repräsentationen aus der Betrachtung heraus. Kräftebasierende Layouts sind für diese Anwendung nicht geeignet, da die Daten streng hierarchisch sind und somit eine „zufällige“ Anordnung nicht passend ist. Die radialen Layouts sind für die Daten ebenfalls unpassend, da es sich um eine zeitlich geordnete Datenstruktur handelt, die durch eine kreisförmige Darstellung weniger intuitiv dargestellt werden kann. Die hierarchischen Layouts bieten sich dagegen an, für diese Visualisierung genutzt zu werden.

Durch die Einordnung in Schichten, siehe Abbildung 4.3 (A), kann leicht und intuitiv der Platz der Kollektionen in der Verbreitungshistorie angezeigt werden. Dazu werden die Kollektionen, wie weiter unten beschrieben, sinnvoll in Gruppen eingeteilt, die es erlauben, die Kollektionen getrennt in verschiedene Schichten einzuordnen. Unter den Kollektionen können vier Gruppen ausgemacht werden. Die erste Gruppe von Kollektionen sind diejenigen die als Vorlage für andere Abschriften gedient haben, allerdings selbst keine Vorlage hatten. Die zweite Gruppe von Kollektionen hat sowohl Briefe erhalten, als auch weiterverbreitet. Als nächste Gruppe können die Kollektionen identifiziert werden, die einen Brief erhalten haben, allerdings selbst nicht als Abschrift gedient haben. Die letzte Gruppe wird durch die Kollektionen gebildet, deren Briefherkunft unbekannt ist und auch an keinen bekannten Weiterverbreitungen beteiligt waren. Dadurch kann man 3-4 Schichten identifizieren, die in der Visualisierung repräsentiert werden sollen.

Briefe sind in den zugehörigen Kollektionen enthalten. Um den Benutzer jedoch nicht mit einer Flut von Kollektionen und Briefen zu überfordern, werden zuerst nur die Kollektionen gezeigt, wie in Abbildung 4.3 (C) zu sehen. Dann werden mittels Interaktion durch den Benutzer Details zu Briefen und Kollektionen geliefert, um tiefer gehende Analysen möglich zu machen.

Einen ähnlichen Ansatz wird auch auf die Verbindungen bzw. die Kanten der Visualisierung angewendet. Initial werden nur aggregierte Kanten angezeigt, die die Anzahl der weitergegebenen Briefe darstellt, indem ein Pfeil mit entsprechender Breite auf die nächste Schicht zeigt („Schichtkanten“), wie in Abbildung 4.3 (B) zu beobachten. Die nächste Stufe von Kanten verbindet bereits Kollektionen miteinander. Diese stellen dar, welche Kollektionen über Verbreitungswege überhaupt in Verbindung stehen („Übersichtskanten“), diese sind in Abbildung 4.4 für die Verbindung zweier Kollektionen

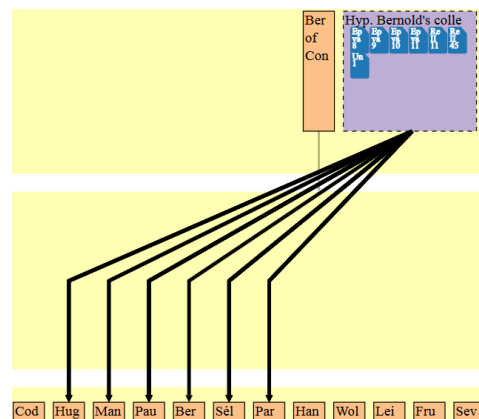


Abbildung 4.4: Die Abbildung zeigt wie die Übersichtskanten einer geöffneten Kollektion aussehen. Man kann in dieser Abbildung auch das Führen der Kanten erkennen. Diese Kanten enthalten einen Knick, der dazu genutzt wird, einen ansprechenderen Eintrittswinkel in die Zielkolektionen zu erreichen. Außerdem kann man die Schichtenkante erkennen, die durch das Öffnen einer Kollektion in den Hintergrund gerückt wurde, um die Analyse der Übersichtskanten nicht zu behindern.

dargestellt. Die Ansicht der Verbreitungswege der einzelnen Briefe stellt die letzte Stufe der Kanten dar („Briefkanten“).

4.3 Kanten Probleme

Im Folgenden werden Probleme beleuchtet, die durch die Kanten zwischen den einzelnen Teilen der Visualisierung entstehen, und deren Lösungskonzepte erklärt. Ein Problem bei der Visualisierung mittels eines Knoten-Kanten Graphen stellen häufig die Kanten dar. Kanten sind schwerer zu lesen, wenn sie in einem kleinen Winkel auf den Zielknoten treffen oder Knoten schneiden. Außerdem gibt es in Knoten-Kanten Graphen häufig Probleme mit Kantenkreuzungen, die eine Hürde für die Skalierbarkeit und Lesbarkeit darstellen.

Die Darstellung der Verbreitungswege von Briefen über verschiedene Kollektionen hat ohne Gegenmaßnahmen genau die vorher genannten Probleme. Durch die nebeneinander liegenden Kollektionen in einer Schicht treffen einige Kanten mit einem flachen Winkel auf die Knoten. Auch die Kantenkreuzungen sind bei dieser Arbeit ein Problem, welches die Verfolgung von Verbreitungswegen kaum durchführbar macht.

Um dem Problem der flachen Kanten entgegenzuwirken, wird eine sogenannte Kantenführung („edge routing“) verwendet, die in Abbildung 4.4 angewendet wurde. Damit werden Kanten nicht einfach mit einer geraden Linie dargestellt, sondern durch zusätzliche Punkt hindurchgeführt. Dadurch wird es möglich, Kanten an anderen Objekten der Visualisierung vorbeizuführen und flach eingehende Kanten so zu führen, dass der letzte Teil einer Verbindung steil in den Knoten eingeht.

Ein wichtiges Konzept Kantenkreuzungen zu verhindern ist das Umsortieren von Knoten. Das Umsortieren von Knoten in dieser Arbeit wird mittels der baryzentrischen Heuristik, siehe Grundlagen,

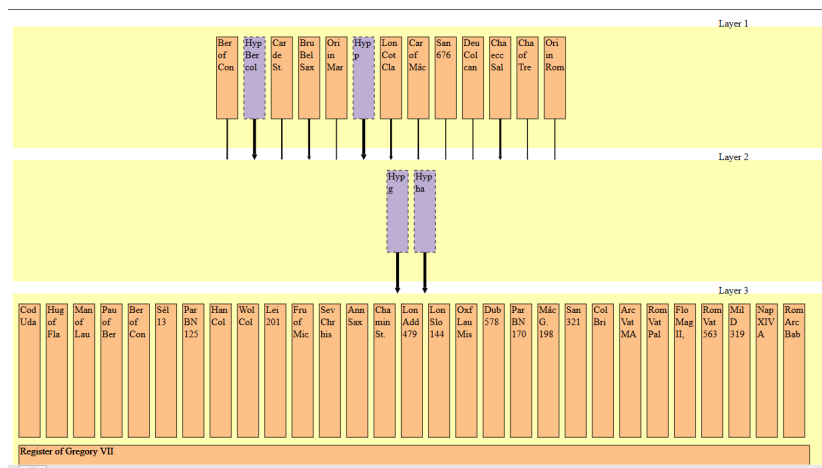


Abbildung 4.5: In dieser Abbildung sieht man die initiale Ansicht. Man erkennt die Kollektionen in ihren Schichten und die Schichtkanten.

durchgeführt. Durch die Verwendung einer Heuristik werden Ressourcen eingespart, wodurch die Interaktivität des Werkzeugs nicht unterbrochen wird. In dieser Visualisierung wird für die Berechnung der Heuristik angenommen, dass nur die Kollektionen der obersten Ebene verschiebbar sind. Da diese Kollektionen die meisten ausgehenden Verbindungen haben, ist dies ausreichend, um die Anzahl der Kantenkreuzungen zu minimieren. Da die Briefe einen festen Platz innerhalb der Kollektionen haben, dürfen diese nicht umsortiert werden. Diese wandern also mit den Kollektionen mit. Da die Übersichtskanten allerdings durch die Briefverbindungen definiert sind, verbessert sich durch die Umsortierung auch die Darstellung der Briefkanten. Nach der Berechnung der Heuristik werden die Knoten derart umsortiert, dass Kollektionen in räumlicher Nähe zu den Kollektionen liegen, zu denen eine Verbindung existiert. Dadurch werden die Kantenkreuzungen reduziert oder sogar vollständig entfernt, wenn dies auf den Daten möglich ist.

4.4 Interaktion

Das wichtigste Konzept für diese Arbeit ist die Interaktion. In dieser Arbeit ist es besonders wichtig, dem Benutzer per Interaktion die Möglichkeit zu geben, sich bestimmte Teile der Visualisierung genauer anzuschauen. Um dem Benutzer anzuzeigen, dass mit einem Gegenstand der Visualisierung eine Interaktion möglich ist, verändert der Mauszeiger seine Form.

Die initiale Darstellung, siehe Abbildung 4.5, zeigt dem Benutzer die Kollektionen in ihren Schichten und die Schichtkanten. Zur Interaktion wird dem Benutzer eine Funktion zur Verfügung gestellt, die es ihr/ihm gestattet, einzelne Kollektionen zu öffnen und somit deren Briefe anzeigen zu lassen, wie in Abbildung 4.6 zu sehen. Kollektionen die geschlossen bleiben, verstecken ihre Briefe und vermeiden so eine unnötige Überladung der Visualisierung. Zusätzlich zum Verstecken und Anzeigen von Briefen, kann der Benutzer sich per Interaktion unterschiedliche Kanten anzeigen lassen. Ist er an einer bestimmten Kollektion interessiert, kann er diese öffnen und sich somit die von dieser Kollektion



Abbildung 4.6: In dieser Abbildung wird eine offene Kollektion dargestellt. Durch das Öffnen der Kollektion werden die Brieficons sichtbar, welche die Briefe darstellen, die diese Kollektion enthält

ausgehenden Übersichtskanten anzeigen lassen. Dies rückt gleichzeitig die Schichtenkanten der anderen Kollektionen in den Hintergrund. Mit einer weiteren Interaktion kann der Benutzer einen Brief aus einer geöffneten Kollektion auswählen. Dadurch werden die Briefkanten des ausgewählten Briefes angezeigt und die Übersichtskanten aller anderen Kollektionen ausgeblendet, wie in Abbildung 4.7 zu sehen. Dadurch kann der Benutzer sich ausschließlich auf die gewünschten Details konzentrieren, ohne durch überflüssige Objekte behindert zu werden.

Wenn der Benutzer sich entschließt eine andere Kollektion anschauen zu wollen, kann er die geöffneten Kollektionen wieder schließen, wodurch die Briefkanten ebenso verschwinden, um dem Benutzer die mühselige Arbeit zu ersparen, jeden Brief einzeln abzuwählen. Es besteht zusätzlich die Möglichkeit einzelne Briefe abzuwählen.

4.5 Farbgestaltung

Eine wichtige Rolle in jeder Visualisierung spielt die Gestaltung der Elemente ebendieser. Diese darf dem Benutzer keine Intuitionen bieten, die in den Daten nicht enthalten sind. Somit ist für diesen Fall eine Farbgebung im Bereich der qualitativen Farbgebung richtig, da in der Visualisierung selbst keine Angaben über etwa die Verlässlichkeit oder Qualität des Briefes/der Kollektion gemacht werden. Diese dient ausschließlich der Verfolgung des Verbreitungsweges. Allerdings gibt es trotzdem ein paar Unterscheidungsmerkmale innerhalb der Visualisierung. Dazu werden zuerst einmal die verschiedenen Schichten des Layouts farblich abgehoben, um den Nutzer diese anzuzeigen, allerdings ohne diese zu sehr in den Fokus zu rücken. Dazu wird ein leichter orange Ton (HEX #ffffb3) verwendet. Zusätzlich unterscheiden sich die Kollektionen, je nachdem ob es sich um eine real existierende, dunkel orange (HEX #fdc086) gefärbt, oder um eine hypothetische Kollektion, violett (HEX #beaed4) gefärbt, handelt, da diese Information für die Analyse berücksichtigt werden muss. Ein weiterer Aspekt der Farbgebung liegt bei den Briefen. Diese werden anders dargestellt, je nachdem ob der Brief derzeit ausgewählt ist oder nicht. Ein ausgewählter Brief wird grün (HEX #33a02c) hervorgehoben, während nicht ausgewählte Briefe blau (HEX #1f78b4) sind.

Schlussendlich ist noch auf allgemeine Vorsichtsmaßnahmen bei der Farbwahl zu achten. So sollte es dem Benutzer möglich sein das Werkzeug zu benutzen, auch wenn dieser z.B. an einer rot-grün Schwäche leidet.

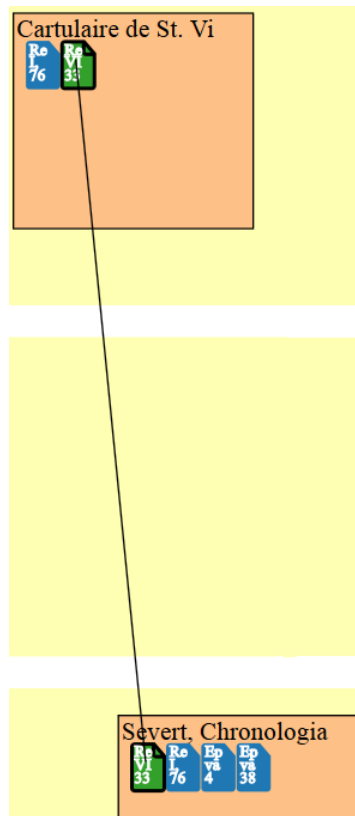


Abbildung 4.7: Das grün ausgefüllt Brieficon wurde in dieser Abbildung vom Benutzer ausgewählt. Durch das Auswählen dieses Briefes wurden die Übersichtskanten ausgeblendet und eine neue Briefkante gezeichnet. Diese führt vom ausgewählten Brief zu allen weiteren Briefen, die eine Abschrift des ausgewählten Briefes darstellen.

5 Implementierung

Die Implementierung der vorher genannten Konzepte wurde mittels der JavaScript Bibliothek D3 [BOH11] in der Version 3 vorgenommen. D3 steht für „Data Driven Documents“ und ist speziell dafür geeignet Dokumente anhand von Daten zu verändern.

5.1 XML Datenbank

Die Daten für die Visualisierung werden durch das Auslesen einer XML Datenbank gewonnen. Die für die Visualisierung wichtigen Teile sind so aufgeteilt, dass es zwei Hauptblöcke für die Kollektionen und die Briefe gibt. Für eine Kollektion existiert immer ein Eintrag, wie in Listing 5.1 zu sehen, in dem Eigenschaften wie Status, Art und die XML ID gespeichert sind. Dieser Eintrag enthält dann weitere Untereinträge, in denen tiefer gehende Informationen wie Name, Autor und ein Kommentar stehen.

```
1 <collection status="real" xml:id="c002" collection_type="letter_collection">
2   <commonly_known_as>Codex Udalrici</commonly_known_as>
3
4   <bibliographical_reference>Johann Georg von Eckhart, Corpus historicum medii aevi sive scriptores res in orbe universo... volume 2
5     (1723)</bibliographical_reference>
6   <commentary> A new edition is being prepared by Klaus Naß from the MGH.</commentary>
7
8 </collection>
```

Listing 5.1: Fragment der XML-Datenbank eines Kollektionen XML-Eintrags

Die Briefeinträge sind ähnlich aufgebaut und in Listing 5.2 dargestellt. Hier stehen Informationen, wie die Kollektionen, die den Brief enthalten, und die Position des Briefes in der entsprechenden Kollektion. Durch diese Informationen ist es überhaupt erst möglich die Briefe den Kollektionen zuzuordnen. Im Programm wird über die *d3.xml* Funktion auf die Datenbank zugegriffen. Im folgenden werden die für die Visualisierung wichtigen Informationen über die Briefe und Kollektionen aus dieser extrahiert und in verschiedenen Datenstrukturen gespeichert. Die Einträge über Kollektionen werden über den Befehl *getElementsByTagName("collection")* aus der XML Datenbank herausgefiltert und abgespeichert. Über des gleichen Befehl wird die Datenbank auf die Briefeinträge abgesucht und diese ebenfalls abgespeichert. Somit gibt es für jeden Brief und jede Kollektion nun einen eigenen Datensatz.

5 Implementierung

```
1 <letter double_entry="double_entry_false" dictatus_pape="dictatus_pape_false" status="genuin" document_type="normal_letter" xml:id="l029"
2   tradition_type="vagans" edition="Cowdrey">
3
4   <page_in_edition from="74" to="76"/>
5
6   <commonly_known_as>Ep. vagans 29</commonly_known_as>
7
8   <also_known_as type_of_name="jl-number">JL 5130</also_known_as>
9
10  <included_in id_of_collection="c043">
11    <position_in_collection>
12      <em type="number" value="35"/>
13      <em type="folio" value="21v"/>
14    </position_in_collection>
15    <textual_features/>
16    <rubric/>
17  </included_in>
18
19  <date_of_issue>
20    <tei:date when-iso="1079-04"/>
21  </date_of_issue>
22
23  <abstract>To the monks of <tei:placeName>Monte Cassino</tei:placeName>. <tei:persName ref="#greg-7">Gregory</tei:persName>
24    raises the
25    interdict which he had placed upon their church and desires their prayers.</abstract>
26
27  <address_lines xml:lang="la">
28    <intitulatio>Gregorius episcopus servus servorum Dei</intitulatio>
29    <adressatio>venerabili congregacioni Casinensi cenobii</adressatio>
30    <benedictio>salutem et apostolicam benedictionem</benedictio>
31  </address_lines>
32
33  <incipit xml:lang="la">Nuper, dilectissimi fratres</incipit>
34
35  <datum_line not_there="true"/>
36
37  <commentary> </commentary>
38 </letter>
```

Listing 5.2: Fragment der XML-Datenbank eines Brief XML-Eintrags

5.2 Geschichtetes Layout

Die Kollektionen werden durch ihre Rolle im Verbreitungsprozess der Briefe in Schichten eingeteilt. Diese Einteilung wird durch die extrahierten Angaben aus der XML Datenbank unter dem Punkt „*ff*“ bei den Briefen festgelegt. Dieses Attribut zeigt an von welcher Kollektion der Brief an die derzeit analysierte Kollektion weitergegeben wurde. Mit Hilfe dieser Angaben kann man vier Typen unterscheiden: die Kollektionen, die

- ausschließlich Briefe ohne *ff* enthalten, aber als *ff* bekannt sind
- Briefe mit *ff* enthalten und als *ff* bekannt sind
- Briefe mit *ff* enthalten und nicht als *ff* bekannt sind
- ausschließlich Briefe ohne *ff* enthalten und nicht als *ff* bekannt sind

Aus diesen Informationen kann man den Kollektionen, entsprechend dem in Kapitel 4.2 erklärten Layoutkonzept, eine Schicht zuordnen. Als nächsten Schritt für den Aufbau der Visualisierung wird die Breite der Knoten in der Darstellung berechnet. Zu diesem Zweck wird die maximale Breite des Bildschirms mittels `screen.MaxWidth` abgefragt. Da sich in der dritten Ebene die meisten Kollektionen mit Verbindungen befinden, wird die Maximalbreite des Bildschirms durch die Anzahl der Kollektionen in dieser Ebene geteilt. Dadurch wird erreicht, dass alle Kollektionen der ersten drei Ebenen jeweils auf einer Y-Koordinate liegen können.

Um den Knoten Werte für die X-Koordinate zuordnen zu können, werden X-Koordinaten vorberechnet, sodass die darzustellenden Kollektionen zentriert sind. Es werden dazu ausgehend von der Anzahl, einem Offset und der Breite die Werte für die X-Koordinate so errechnet, dass die Kollektionen zentriert liegen. Nun werden alle Knoten zunächst einmal an diese Positionen eingefügt. Da dadurch allerdings unschöne Kantenkreuzungen erzeugt werden, die wie in Kapitel 4.3 erläutert vermindert werden können, werden um diese zu verringern die Knoten in der obersten Ebene mittels der baryzentrischen Heuristik umsortiert. Dabei erhält jede Kollektion in dieser Ebene einen Wert, der sich aus den X-Koordinaten der verbundenen Knoten in den darunterliegenden Ebenen ergibt. Die X-Koordinaten werden aufaddiert und dann durch die Anzahl der Knoten geteilt. Dadurch ergibt sich die in Listing 5.3 zu sehende Implementierung:

```

1 degree = ConnectedTo[node].length; // degree beschreibt den Grad des Knotens "node"
2 for (var k = 0; k < ConnectedTo[node].length; k++){ // Über alle Knoten laufen die mit "node" verbunden sind
3     sumofX = sumofX + OverviewCollX[ConnectedTo[node][k]]; // Alle X-Koordinaten der verbundenen Knoten aufaddieren
4 }
5 value[node] = 1/degree * sumofX; // Den Wert der aufsummierten X-Koordinaten durch den Grad des Knoten teilen

```

Listing 5.3: Berechnung baryzentrischer Wert

Diese Werte („value[node]“) werden abgespeichert und mittels einer *sort* Funktion der Größe nach absteigend sortiert. Außerdem werden die X-Koordinaten der Knoten in der obersten Ebene ebenfalls der Größe nach absteigend sortiert. In einem weiteren Schritt wird jeder Kollektion entsprechend ihres Wertes ein X-Wert zugewiesen. Dadurch werden Kollektionen mit einem niedrigen baryzentrischen Wert an eine niedrige X-Koordinate geschoben, während Kollektionen mit einem hohen Wert eine hohe X-Koordinate zugewiesen bekommen. Somit liegen Knoten nahe bei Knoten zu denen eine Verbindung existiert, was Kantenkreuzungen verringert.

5.3 Kantenführung

Die Kantenwege werden mittels eines Zwischenpunktes verbessert. Dies wird derart implementiert, dass eine Kante die eine gewisse Steigung unterschreitet einen zusätzlichen Zwischenpunkt erhält, wie in Abbildung 4.4. Dieser verbessert den Winkel zur verbundenen Kollektion und vereinfacht die Verfolgung der Kanten. Der Zwischenpunkt erhält den gleichen X-Wert wie der Zielpunkt der Kante. Der Y-Wert des Zwischenpunktes ergibt sich aus dem Durchschnitt der Y-Koordinate des Start- und Zielknotens, um 20 Pixel nach unten versetzt. Dadurch liegt der neue Zwischenpunkt genau über dem Zielpunkt, wodurch die Kante zwischen diesen beiden Punkten gerade ist, was die Klarheit der Darstellung verbessert.

5.4 Briefe

Die Briefe werden in der Visualisierung durch ein kleines Brief Icon repräsentiert. Dieses Icon wird aus einer separaten Datei gewonnen und für jeden einzelnen Brief repliziert. Die Briefe werden gruppiert, um das Ein- und Ausblenden dieser mittels Interaktion mit den Kollektionen einfacher zu gestalten. Dazu wird für jede Kollektion eine Gruppe eröffnet, in welcher dann die einzelnen Briefe eingeordnet werden. Nun kann man die Anzeige der Briefe manipulieren, indem man das Attribut *style("visibility")* der Briefgruppe auf *"visible"* oder *"hidden"* setzt. Für die einzelnen Brief Icons wurde eine *on("click")* Funktion implementiert. Diese hebt das geklickte Icon hervor und stellt die Verbindungen des ausgewählten Briefes zu Briefen dar, die durch den ausgewählten Brief weiterverbreitet wurden.

5.5 Öffnen von Kollektionen

Nachdem dem Nutzer ein Überblick gegeben wurde, hat dieser die Möglichkeit Verbreitungswege genauer zu analysieren. Dazu wurde eine Funktion implementiert, die es dem Nutzer erlaubt die Kollektionen zu öffnen. Dies wurde mittels einer *on("click")* Funktion umgesetzt, die sich auf die Repräsentation der Kollektionen als Rechtecke bezieht. Wenn der Benutzer nun auf eine Kollektion einen Linksklick mit der Maus ausführt, wird diese Funktion aufgerufen. Dadurch öffnet sich die Kollektion, d.h. das Rechteck wird breiter und die vorher versteckten Briefsymbole werden sichtbar. Außerdem werden die vorher aggregierten Kanten aufgespalten, um dem Nutzer genau anzuzeigen, mit welchen Kollektionen die Geöffnete verbunden ist. Diese Kantendicke hängt mit der Anzahl der Briefe zusammen, durch die die Kollektionen in Verbindung stehen, wie in Abbildung 4.4 zu sehen. Da eine lineare Skalierung sehr unschöne Unterschiede zwischen den Pfeilbreiten verursacht hat, wurde schlussendlich eine Logarithmische Skalierung verwendet. Zusätzlich werden die Übersichtskanten, die in der gleichen Ebene mit der zu öffnenden Kollektion liegen mittels der *style(„opacity“)* Funktion ausgegraut. In der *on("click")* Funktion laufen einige Dinge im Hintergrund ab, die im Folgenden erläutert werden.

Das Öffnen einer Kollektion hat große Auswirkungen auf die restlichen Kollektionen in der Visualisierung. Wenn eine Kollektion am rechten Rand geöffnet wird, ist dies der einfachste Fall. Dann muss nur die Breite der geöffneten Kollektion verändert werden. Ein Problem bei diesem Fall ist eine komplett gefüllte Schicht. Beim Öffnen einer Kollektion in einer solchen Schicht ragen die Kollektionen über den Bildschirmrand hinaus. Eine Lösung für dieses Problem wäre zum Beispiel, den angezeigten Bereich beim Öffnen einer solchen Kollektion automatisch nach rechts zu scrollen. Bei einer Kollektion am linken Rand wird jede weitere Kollektion rechts davon um die zusätzliche Breite der geöffneten Kollektion verschoben. Falls mehr als eine Kollektion geöffnet ist, muss gezählt werden und diese zusätzliche Breite berücksichtigt werden. Im Allgemeinen muss dann noch unterschieden werden, ob die Kollektionen überhaupt auf der selben Ebene liegen.

Weiterhin hat das Öffnen einer Kollektion Auswirkungen auf die Koordinaten aller anderen Objekte innerhalb der Visualisierung. So muss bei jeder Öffnung der Wert der Kanten und der Briefe angepasst werden, sodass sie sich mit verschieben.

Die selben Berechnungen müssen auf umgekehrten Weg ausgeführt werden, wenn eine Kollektion

geschlossen wird. Hierbei muss man noch Sonderfälle beachten. Falls die Kollektion die letzte geöffnete war, müssen alle Übersichtskanten den *style("visibility", "hidden")* erhalten, um sie auszublenden. Weiterhin müssen die Schichtkanten wieder dargestellt werden.

6 Ergebnis

In diesem Kapitel wird ein abschließender Vergleich zwischen der ursprünglichen Visualisierung der Daten und der in dieser Arbeit entwickelten Visualisierung vorgenommen.

Im Vergleich mit der ursprünglichen Visualisierung, zu sehen in Abbildung 6.1, wurde das grundsätzliche Prinzip, also das geschichtete Layout, auf Wunsch des Auftraggebers beibehalten. Ein Problem in der ursprünglichen Visualisierung war dabei, dass die unterste Schicht nicht weiter unterteilt wurde, wie es bei dieser Arbeit geschehen ist. Dadurch wurde diese in einem solchen Maße breit, dass der Benutzer weit nach rechts scrollen musste, um die am weitesten rechts liegende Kollektion zu sehen. Außerdem gab es durch diese Breite Kanten, die zu lang waren, um zu Analysezielen verwendet werden zu können. Dieses Problem konnte durch die Unterteilung der untersten Schicht in Kollektionen mit bekannten Verbindungen und Kollektionen ohne Verbindung verbessert werden. Weiterhin wurden die Kollektionen ohne Verbindung nach einer Bildschirmbreite umgebrochen. Dadurch passt der Überblick über alle Kollektionen nun auf eine einzige Bildschirmbreite. Der Nachteil der dadurch entsteht, nämlich die vergrößerte Höhe der Visualisierung, ist nicht schwerwiegend, da die unteren Kollektionen ohnehin keine Verbindungen besitzen. Die Verbindungen in der Visualisierung dieser Arbeit sind durch die verringerte Breite nun kürzer und so besser lesbar und verfolgbar. Dies wurde auch dadurch erreicht, dass Interaktion in größerem Maße in die Visualisierung eingebaut wurde, welche vorher nur gering vorhanden war.

Eine Funktion in der alten Visualisierung machte es möglich, Kanten durch *hovern* über die Briefe hervorzuheben, was in der Visualisierung dieser Arbeit aktuell noch nicht möglich ist, aber eine

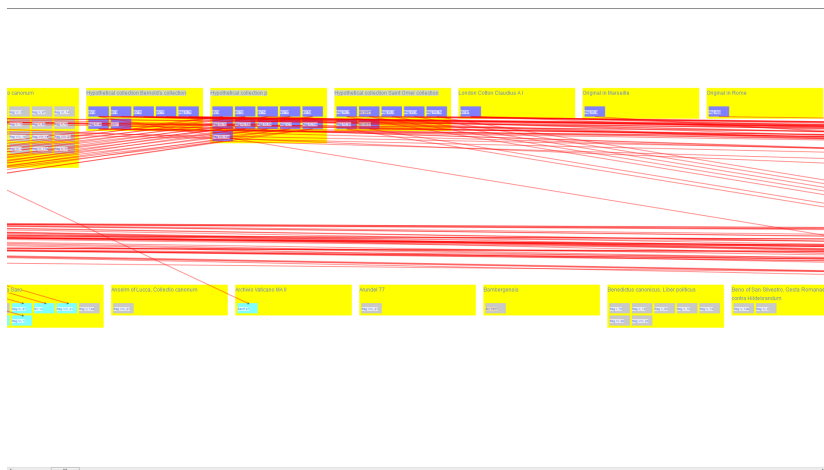


Abbildung 6.1: In dieser Abbildung kann man die ursprüngliche Visualisierung des Auftraggebers sehen, die für die Analyse der Verbreitungswege verwendet wurde.

nützliche Interaktionsmöglichkeit darstellt. In der ursprünglichen Visualisierung wurden alle Briefe und Kanten von Anfang an angezeigt. In der neuen Visualisierung ist es dem Benutzer möglich, die Briefe mittels Öffnen und Schließen von Kollektionen anzuzeigen oder zu verstecken. Zusätzlich wurden Schicht- und Übersichtskanten eingebaut, die dem Benutzer einen Überblick über die Gesamtsituation geben, ohne direkt alle Details anzuzeigen. Somit ist die Visualisierung nicht mehr voller Kanten, diese können vom Benutzer manuell angezeigt werden lassen. Ein Nachteil dieser Herangehensweise ist allerdings der erhöhte Aufwand alle Kanten anzeigen zu lassen, wenn dies vom Benutzer gewünscht wird.

7 Zusammenfassung und Ausblick

In dieser Arbeit wurden zum Beginn in den Grundlagen die nötigen Grundkenntnisse zum Verständnis vorgestellt. Dabei ging es vor Allem um Knoten-Kanten Graphen. Diese wurden zunächst eingeführt und dann deren verschiedene Layouts vorgestellt. Weiterhin wurde der Sugiyama Algorithmus kurz zusammengefasst, da dieser der Grundstein für das verwendete geschichtete Layout bildet. Im Anschluss wurde diese Arbeit in Zusammenhang mit anderen Tools gestellt, die ebenfalls Knoten-Kanten Graphen verwenden. Dabei wurde erklärt, wie diese Arbeit sich von diesen Tools unterscheidet oder sich ähnelt und warum dem so ist.

Daraufhin wurden die Konzepte vorgestellt, mit deren Hilfe die Anforderungen an die Visualisierung umgesetzt werden sollten. Dabei wurde ein geschichtetes Layout gewählt, um die Vorstellung des Weiterverbreitens von oben nach unten intuitiv darzustellen. Im weiteren Verlauf der Konzepte wurde vorgestellt, dass die Kanten in solchen Graphen ein Problem darstellen und wie man diese lösen kann. Ein Konzept war hierfür auch die Interaktion, die eine wichtige Rolle dabei spielt, dem Benutzer nicht zu viele Informationen auf einmal zu geben. Zum Abschluss wurde das Konzept zur Gestaltung der Elemente in der Visualisierung vorgestellt.

Im Folgenden wurde die konkrete Umsetzung der Konzepte in der Implementierung mittels D3 erklärt. Zuerst wurde dabei auf die XML-Datenbank eingegangen, aus der die Daten für die Visualisierung stammen und wie man diese ausliest. Daraufhin wurde die Implementierung des geschichteten Layouts vorgestellt. Nach dem Layout wurde die Implementierung des Konzeptes der Kantenführung präsentiert, mit dem ,die im Konzeptkapitel vorgestellten, Kantenprobleme verringert wurden. Im weiteren Verlauf wurde auf die Elemente in der Visualisierung, namentlich die Briefe und Kollektionen, eingegangen.

Im Ergebniskapitel wurde die ursprüngliche Visualisierung mit der dieser Arbeit verglichen.

Ausblick

Durch das Feedback des Auftraggebers konnten noch Verbesserungen herausgefunden werden, die in dem Tool gewünscht wurden. So kann man über weitere Unterschiede in der Farbgebung der Kollektionen noch andere Attribute reflektieren. Ein Beispiel wäre das hervorheben von Kollektionen, die als *fons formalis* eingetragen sind, um die Schichtzuordnung klarer zu machen. Ein *fons formalis* beschreibt die Kopiervorlage für einen Brief.

Eine weitere Idee ist die Einführung einer zweiten Ansicht, in die der Benutzer eine Art chronologische Anordnung der Kollektionen anzeigen lassen kann. Damit würden die Kollektionen in eine nachvollziehbarere Reihenfolge geordnet werden, was allerdings möglicherweise die Lesbarkeit reduziert. Deshalb sollte die hier entwickelte Ansicht mit der chronologischen zusammen genutzt werden.

Eine zusätzliche Funktion wäre das Einfügen von eigenen Kollektionen in der Visualisierung. Diese

Funktion würde die Anforderung A5 aus Kapitel 4.1 erfüllen, die in dieser Arbeit nicht mehr umgesetzt wurde. Diese Kollektionen sollen dann innerhalb der Visualisierung mit Briefen und Verbindungen bestückt und in die XML Datei exportiert werden.

Anders als in der aktuellen Visualisierung wurde die Idee eingebracht, die Kollektionengröße mit der Anzahl der Briefe skalieren zu lassen, die sich in dieser befindet. Dadurch könnte der Benutzer mit einem Blick erkennen, welche Kollektion viele oder wenige Briefe enthält.

Eine weitere Funktion, die dem Benutzer weitere Personalisierungsmöglichkeiten geben würde, ist die manuelle Veränderung der Kantenfarbe. Diese Funktion würde es dem Benutzer erlauben, einzelne Kanten mittels einer Farbpalette selbst einzufärben und sich diese somit zu markieren.

Grundsätzlich können diesem Programm weitere Funktionen hinzugefügt werden, um die ungenutzten Daten aus der XML-Datei einzubauen und diese somit komplett zu repräsentieren. Dabei sollte man aber darauf achten, dass die Visualisierung nicht zu überladen wird.

Literaturverzeichnis

- [AAWN13] Y. AlNoamany, A. Alsum, M. C. Weigle, M. L. Nelson. Who and What Links to the Internet Archive. *CoRR*, abs/1309.4016, 2013. URL <http://arxiv.org/abs/1309.4016>. (Zitiert auf den Seiten 23 und 25)
- [BM01] O. Bastert, C. Matuszewski. Layered drawings of digraphs. In M. Kaufmann, D. Wagner, Herausgeber, *Drawing graphs*, S. 87–120. Springer, 2001. (Zitiert auf Seite 16)
- [BOH11] M. Bostock, V. Ogievetsky, J. Heer. D³ data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011. (Zitiert auf Seite 35)
- [DL05] S. Ducasse, M. Lanza. The class blueprint: visually supporting the understanding of glasses. *Software Engineering, IEEE Transactions on*, 31(1):75–90, 2005. doi:10.1109/TSE.2005.14. (Zitiert auf den Seiten 23 und 24)
- [Ead84] P. Eades. A heuristics for graph drawing. *Congressus numerantium*, 42:146–160, 1984. (Zitiert auf den Seiten 15 und 23)
- [Erd36] C. Erdmann. *Die Bamberger Domschule im Investiturstreit*. Beck, 1936. (Zitiert auf den Seiten 27 und 28)
- [GFC05] M. Ghoniem, J.-D. Fekete, P. Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4(2):114–135, 2005. (Zitiert auf den Seiten 11 und 13)
- [HS98] D. Harel, M. Sardas. An algorithm for straight-line drawing of planar graphs. *Algorithmica*, 20(2):119–135, 1998. (Zitiert auf Seite 15)
- [KSB⁺09] M. I. Krzywinski, J. E. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 2009. doi:10.1101/gr.092759.109. URL <http://genome.cshlp.org/content/early/2009/06/15/gr.092759.109.abstract>. (Zitiert auf Seite 23)
- [KW01] M. Kaufmann, D. Wagner. *Drawing graphs: methods and models*, Band 2025. Springer, 2001. (Zitiert auf Seite 17)
- [LDGP05] M. Lanza, S. Ducasse, H. Gall, M. Pinzger. CodeCrawler-an information visualization tool for program comprehension. In *Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on*, S. 672–673. IEEE, 2005. (Zitiert auf Seite 23)
- [SM95] M.-A. Storey, H. A. Muller. Manipulating and documenting software structures using SHriMP views. In *Software Maintenance, 1995. Proceedings., International Conference on*, S. 275–284. IEEE, 1995. (Zitiert auf den Seiten 23, 24, 25 und 26)

- [ST99] J. M. Six, I. G. Tollis. Circular drawings of biconnected graphs. In *Algorithm Engineering and Experimentation*, S. 57–73. Springer, 1999. (Zitiert auf Seite 14)
- [ST06] J. M. Six, I. G. Tollis. A framework and algorithms for circular drawings of graphs. *Journal of Discrete Algorithms*, 4(1):25–50, 2006. (Zitiert auf Seite 14)
- [STT81] K. Sugiyama, S. Tagawa, M. Toda. Methods for Visual Understanding of Hierarchical System Structures. *Systems, Man and Cybernetics, IEEE Transactions on*, 11(2):109–125, 1981. doi:10.1109/TSMC.1981.4308636. (Zitiert auf den Seiten 16 und 17)
- [Tam13] R. Tamassia. *Handbook of graph drawing and visualization*. CRC press, 2013. (Zitiert auf den Seiten 11, 14, 15, 17, 18, 19, 20 und 21)
- [VLKS⁺11] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, D. W. Fellner. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. In *Computer graphics forum*, Band 30, S. 1719–1749. Wiley Online Library, 2011. (Zitiert auf Seite 14)
- [Wei07] M. Weinard. *Analyse von Heuristiken*. Müller, 2007. (Zitiert auf Seite 15)

Alle URLs wurden zuletzt am 11.02.2015 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift