



TECHNISCHE
UNIVERSITÄT
DRESDEN



INTERACTIVE
MEDIA LAB
DRESDEN

Fakultät Informatik Institut für Software- und Multimediatechnik, Multimedia-Technologie

Dokumentation

INFOVIS, TEAM D

Paula Schöley, Robert Morawa, Gilbert Röhrbein, Alexandra Weiß

Eingereicht am 8.Juli 2013

1 EINFÜHRUNG

1.1 Kanji Schriftsystem

Die japanische Schrift besteht aus drei Alphabeten, wobei eines aus Logogrammen – sogenannten Kanji – besteht und die anderen beiden Silbenalphabete sind. Kanji stammen aus der chinesischen Sprache und wurden von den Japanern übernommen und haben sich seitdem parallel entwickelt. In einem Satz werden Kanji unter anderem für den Wortstamm und die Silbenalphabete für grammatikalische Beugung (beispielsweise Negation oder Zeitformen) verwendet. Hiragana und Katakana, also die Silbenalphabete, können aber auch als Ersatz für ein Kanji verwendet werden, indem sie die Aussprache (die Lesung) des Kanji beschreiben.

Kanji bestehen aus einzelnen Elementen, sogenannten Radikalen, die miteinander kombiniert werden können. Wenige der Radikale sind bereits bedeutungstragend, wie beispielsweise das Radikal 氵, welches für Kontext „Wasser“ steht. Somit ist es beispielsweise in den Wörtern für „Flut“ (氾濫) oder auch „Meer“ (海) enthalten.

1.2 Tube-Maps

Tube-Maps oder Transit Maps (deutsch: Liniennetzplan) sind eine Form der Visualisierung, die für Linien öffentlichen Nahverkehrs – also Busse, U- und Straßenbahnen – verwendet wird. Bisher werden Tube-Maps nur von Menschen gefertigt, da viele verschiedene Faktoren das Aussehen und die Eigenschaften der Tube-Map beeinflussen.

Eigenschaften

- Es existieren Stationen, die verschiedene Ausprägungen, wie „Endstation“ oder „Umstiegsmöglichkeiten“ besitzen.
- Stationen liegen auf Linien, die diese miteinander gemeinsam haben. Linien entsprechen den verlegten Gleisen beziehungsweise Routen, die die einzelnen Stationen verbinden. Es gibt die Möglichkeit, verschiedene Arten von Linien visuell zu unterscheiden, um zum Beispiel zwischen Straßenbahn- und Buslinie zu unterscheiden. Des Weiteren dürfen die Linien nur horizontal, vertikal oder im 45° Winkel zueinander sein.
Linien können sich ab einer Station aufteilen, um zu zwei alternativen Endstationen zu fahren. Hinzu kommen Ringbahnen, bei denen die Endstation der Anfangsstation entspricht.
- In [Stott, 2008] werden als weiterer Bestandteil der Map topographische Metadaten, also Orientierungspunkte wie Flüsse, Flughäfen, Bahnhöfe, oder sehr wichtige touristische Orte genannt. Dabei helfen sie bei der Orientierung, indem geographische Bezüge zwischen Stationen und Landmarks hergestellt werden.
- Die Beschriftung der vorhandenen Stationen und Linien eines Liniennetzes spielt eine wichtige Rolle beim Kennenlernen und Verinnerlichen der vorhandenen Linien. Häufig wird eine Farbcodierung für die Beschriftung der Linien verwendet, während dies bei Stationen nicht der Fall ist und diese direkt mit Text versehen werden.
- Zonen stellen in Tube-Maps meist Tarifzonen dar, können aber auch weitere Eigenschaften visualisieren, die sich durch Entfernung kodieren lassen.

Weitere Verwendungszwecke

Wie in [Stott, 2008] beschrieben, gibt es auch weitere Anwendungsgebiete für Tube-Maps, wie Projektpläne, Karten von thematischen Verbindungen zwischen Büchern [O'Reilly, 2003], Aufbau von Webseiten, sowie Metabolic Pathways. Eigene Recherche des Teams hat ähnliche Ergebnisse ergeben, wobei nicht immer bei allen Visualisierungen klar war, warum eine Visualisierung in Form einer Tube-Map gewählt wurde oder sich die Daten besonders dafür eignen.

2 MOTIVATION

2.1 Warum Kanji?

Die japanische Schrift ist streng hierarchisch und modular aufgebaut. Der Datensatz ist begrenzt, da keine neuen Kanji „erfunden“ werden, und er lässt sich reduzieren auf sogenannte Jouyou Kanji. Jouyou sind Zeichen, die besonders häufig im Alltagsleben verwendet werden, und die somit ein Großteil der Japaner beherrscht. Im Vergleich zur Gesamtzahl aller Kanji (das Team verwendet eine Datei mit 12000 Kanji) beschränken sich die Jouyou auf knapp 2000 Zeichen, was eine starke Reduzierung des Datensatzes ermöglicht. Diese Reduzierung ist nicht nur von praktischer Bedeutung, es ist außerdem für Interessierte, die die Sprache lernen wollen, ein guter Startpunkt.

2.2 Warum eine Tube-Map für Kanji?

Durch den modularen Aufbau von Kanji können Radikale als Endstationen interpretiert werden, deren Linien durch mehrere Stationen (also Kanji) verlaufen. Dadurch ergibt sich der Aufbau eines Kanji anhand der Linien, die die Station bedienen.

Lernende können somit existierendes Wissen über Radikale und Kanji zur Orientierung verwenden, oder aber neue Zeichen kennenlernen, indem sie der Linie eines bekannten Radikals folgen. Anhand der Häufigkeitsstatistik, die für die Alltags-Kanji vorhanden ist, kann ein Benutzer sich anhand häufiger Kanji darin enthaltene Radikale merken und diese verwenden, um weitere Kanji zu lernen.

Wer soll es am Ende verwenden? Die Anwendung richtet sich primär an solche, die Kanji lernen möchten. Dabei wird davon ausgegangen, dass entweder ein spezielles Kanji gelernt oder nachgeschlagen wird, oder auch ein generelles Interesse an den Zeichen besteht. Soll beispielsweise das Kanji für „Baum“ gelernt werden, kann man weitere Kanji lernen, die sehr ähnlich dazu sind. [Rose, 2010] beschreibt eine Lernstrategie, die für Kanji eine besonders gute Anwendung findet. Bei dieser Strategie spielt neben visueller Ähnlichkeit auch das sogenannte „grouping“ (also die Bildung von Verknüpfungen zwischen Vokabeln basierend auf Bedeutung oder Konzepten) eine Rolle. Diese Erkenntnisse lassen vermuten, dass Tube-Maps sich zum Lernen von Kanji eignen.

3 HERAUSFORDERUNGEN

Da Haltestellen einen festen Ort haben, der sich nicht ändert, wird dieser als sehr wichtiger Faktor („initial embedding“) für das Layouting in „Automatic Layout of Metro Maps using Multicriteria Optimisation“ von [Stott, 2008] beschrieben. Dabei kann sich die Position geringfügig verändern, zum Beispiel werden Entfernungen zwischen Haltestellen auf dem Plan teilweise nicht in dem eigentlich vorhanden Abstand dargestellt, sondern der Abstand wird angeglichen, um ein uniformes Aussehen des Plans zu gewährleisten. Diese Verortung existiert bei Kanji in diesem Maße nicht und daher müssen Richtungen und Positionen durch einen Algorithmus festgelegt werden.

Ähnlich verhält es sich mit der Kreuzung von Linien außerhalb von Haltestellen. Vor allem bei Schienenverkehr findet eine solche Kreuzung selten statt, meist bedienen Linien stattdessen mindestens eine gemeinsame Haltestelle. Da eine physische Begrenzung dieser Art bei dem verwendeten Datensatz nicht vorhanden ist, müssen Linien dort aufwendig per Routingalgorithmus gelegt werden, um Überschneidungen soweit möglich zu vermeiden.

Da der Datensatz mit knapp 2000 Kanji deutlich größer ist als beispielsweise die Anzahl der Straßenbahnhaltestellen der Dresdner Verkehrsbetriebe (259 laut [DVB-AG, 2011]), musste vor der Implementierung ein Konzept entwickelt werden, wie die Daten automatisch in Form einer Tube-Map korrekt dargestellt werden können.

4 DESIGNPROZESS

4.1 Auswahl eines passenden Datensatzes

In der ersten Phase des Designprozesses hat das Team sich mit mehreren Datensätzen beschäftigt, die mit Hilfe einer Tube-Map visualisiert werden können. Dabei wurden die Datensätze mittels Abstimmung auf vier reduziert, woraufhin sich jedes Teammitglied mit einem Thema beschäftigt hat. Die Ergebnisse wurden im Team vorgestellt und nach Abstimmung und Diskussion das Thema für die Visualisierung gewählt.

Webseiten Wie im Abschnitt 1.2 beschrieben lässt sich die Struktur einer Webseite mittels einer Tube-Map darstellen. Diese Darstellung ist vor Allem für große und komplexe Webseiten von Vorteil, da ein Benutzer sich so einfach eine Übersicht über alle vorhandenen Inhalte verschaffen kann. Problematisch dabei ist jedoch, die Eigenschaften der Tube-Map deutlich hervorzuheben und nicht einen Graphen als Ergebnis zu erhalten. Durch Verlinkungen von Webseiten lässt sich das Ergebnis nicht mehr als Tube-Map darstellen.

Die Idee des Teammitgliedes war daher, häufig besuchte Webseiten und nicht die Struktur einer Webseite darzustellen. Oft besucht ein Benutzer einen Fundus von Seiten im Laufe des Tages. Des Weiteren könnten Zonen für die grobe Klassifizierung des Inhalts der Seiten eingeführt werden, wie beispielsweise „Nachrichten“ oder „Zeitvertreib“. Dabei könnten Linien auch bestimmten Schlagwörtern, zum Beispiel „Technik“, zugeordnet werden, um zusammen mit den Zonen Seiten zu beschreiben, die primär über Neuigkeiten im Bereich von Technik und IT berichten.

Bildeigenschaften Ein weiterer Gedanke verfolgte das Ziel, die Eigenschaften von Bildern wie zum Beispiel Gemälden darzustellen. Hierbei würden die Linien Eigenschaften des Gemäldes und die Stationen das Gemälde selbst darstellen. Es wäre eine Möglichkeit, Muster in dem Gebrauch z.B. von Farbe unter Malern oder Fotografen im Zusammenhang mit ihrer Strömung, der Zeit oder des Bildthemas zu finden. Dabei würde die Tube-Map sowohl auf X- und Y-Achse in Zonen aufgeteilt. Die X-Achse wäre in diesem Beispielkonzept eine Zeitachse, die Y-Achse weist die im Werk dominierenden Farben aus. Sie würden den typischen Zonen einer Tube-Map entsprechen.

Bei diesem Layout würden die Achsen je nach Inhalt gestreckt oder gestaucht und die Linien stetig von einer Hauptlinie abzweigen. Die Auswahl an Bildeigenschaften, die man so kodieren könnte, ist groß. Als Interaktionsmöglichkeit böte sich an, Künstler oder Skalen zu wechseln sowie diverse Filter oder Neuskalierung zu ermöglichen.

Parallele Handlungsstränge Ein dritter Ansatz beschäftigte sich mit parallel verlaufenden Handlungssträngen wie sie beispielsweise in der Fernsehserie Game of Thrones und dessen Buchvorlage vorkommen. Es würde Zuschauern oder Fans der Serie eine Möglichkeit geben, die bisherige Handlung zu rekapitulieren und bisherige Aufeinandertreffen von Charakteren darzustellen. Einzelne Hauptcharaktere erhalten hierbei eine Tube-Map Linie, Kapitel bilden Stationen und Handlungsorte die Zonen der Tube-Map. Diese würde entlang der X-Achse nach von links nach rechts verlaufen, grob an einem Zeitstrahl orientiert oder aber sich radial in alle Richtungen ausbreiten, wobei das die Vergleichbarkeit erschweren würde.

Beschriftungen von Stationen könnten in dieser Tube-Map eine Wortgruppe als Zusammenfassung des Kapitels liefern. Man könnte die Stationen bei Point of View-Kapiteln den PoV auch in der Farbe des jeweiligen Charakters festhalten. Icons könnten Ereignisklassen wie Tode oder kämpferische Auseinandersetzungen symbolisieren. Interaktionen könnten hier die Filterung, das Eingrenzen eines Zeitbereiches und ähnliches beinhalten.

4.2 Entwicklung des Programmkonzepts

Die folgenden Konzepte stellen die Ideen des Teams vor der Implementierung, also in der Konzeptionierungsphase dar und sind deshalb nicht immer exakt so in der Implementierung umgesetzt. Sie spiegeln also alle Ideen und Vorstellungen des Teams wider.

Der gesamte Kanjiraum

Der erste Gedanke war, dass man die komplette Zahl Kanji anzeigen könnte. Aufgrund der hohen Zahl an Radikalen würde jedoch das Einfärben der einzelnen Linien als wichtiges visuelles Element von Tube-Maps nicht möglich. Würde man jede Radikallinie einfärben, würden sich die Farben kaum voneinander abheben. Eine Idee war, dies mittels Hervorhebung zu lösen, sodass man die Radikallinie selbst oder alle Linien eines Kanji per Selektion einfärben kann. Kleine Labels, die bei traditionellen Tube-Maps genutzt werden, um die Verkehrslinien innerhalb der Karte zu markieren, werden analog dazu genutzt, um die Radikallinien zu bezeichnen.

Innerhalb der Ansicht navigierte man vor allem mit Zooming und Panning. Eine Minimap, die die Position des Fensters in der Karte angibt, sorgt hierbei für eine grobe Übersicht. Ideen wurden entworfen, die Kanji hier nach den in ihnen enthaltenen Radikalen zu clustern und diese Cluster erst mittels semantischem Zoom aufzulösen, um die überbordende Komplexität zu verringern und die Analyse nach dem Radikalkriterium zu erleichtern. Mit Filtern hat der Nutzer die Möglichkeit, die Menge der Kanji einzuschränken, hierfür eignen sich Attribute wie die unterschiedlichen japanischen Lesungen, die Bedeutungen, Strichzahl oder Häufigkeit des Kanji in japanischen Zeitungsartikeln. Eine Suche sollte Kanji anzeigen, die den ausgewählten Kriterien entsprechen, von dieser Suchliste wäre eine Autonavigation zum gesuchten Kanji möglich gewesen. Das Kanji wird für eine kurze Zeit farblich hervorgehoben, während alle Kanji, auf die die Suchkriterien zutreffen, mit dickerem Rand gezeichnet werden.

Mittels Hover hat der Nutzer die Möglichkeit, sich ein gewähltes Detail eines Kanji in Form eines Tooltips oder Labels darzustellen. Da ein einfacher Hover bei einer komplexeren Tube-Map rasch zu ungewünschten Details führen kann, wurde es mit einer kurzen Verweilzeit von ca. einer halben Sekunde kombiniert. Um Vergleichbarkeit der Kanji herzustellen, führten wir eine ausblendbare Detailtabelle ein. Dort wird pro Spalte ein ausgewähltes Kanji angezeigt, in den Zeilen die dazugehörigen Details. Auch hier ist es möglich, sich zu den dargestellten Kanjis navigieren zu lassen.

Die Central Station-Ansicht

Nach längerem Überlegen wurde klar, dass die bisherige Struktur des Konzeptes nicht zielführend ist oder sein wird. Man erhält nicht einmal einen groben Überblick über die Kanji, da die Verknüpfungen der Kanji durch Radikale viel zu komplex ist, um sie auf einer Tube-Map abzubilden. Zudem fehlen unter anderem die für eine Tube-Map charakteristischen farbigen Linien. Hinzu kam eine Überlegung, wie Kanji selbst gelernt werden: Normalerweise wird ein Kanji zum Lernen vorgegeben. Mittels der Tube-Map könnte man „verwandte“ Kanji darstellen, die dieselben Radikale teilen wie das zum Lernen ausgewählte. Auf diese Weise soll man das Kanji besser in seinen Kontext einbetten können.

In diesem Konzept gibt es einen „Hauptbahnhof“ (Central Station). Dieses Kanji wird in einer vorherigen Übersicht ausgewählt, in der alle Kanji gelistet sind und gefiltert werden können. Diese Central Station unterscheidet sich signifikant: sie ist größer und zeigt alle Details des Kanji in der Station an. Es werden nur Linien zu den Radikalen angezeigt, die in dem Kanji enthalten sind. Da ein Kanji maximal neun Radikale enthält, ist es nun möglich die Linien mit Farben zu unterscheiden. Auch wird ein Großteil der möglichen Überschneidungen auf diese Weise entfernt. Die Radikallinien führen durch alle Kanji, in denen sie enthalten sind.

Ansonsten bleiben viele Ideen des ursprünglichen Konzeptes erhalten. Farbige, kleine Radikale kennzeichnen die zugehörigen Linien. Kanji können in eine Detailtabelle eingetragen, durchsucht und gefiltert werden. Ebenso bleibt die Autonavigation erhalten. Jedoch entfällt auf diese Weise das Hervorheben der Linien. Stationenbezeichner erfahren eine kleine Änderung. In der Central Station existiert nun die Möglichkeit, Detailkategorien auszuwählen, die dort angezeigt werden.

Zudem wird eine weitere Interaktionsmöglichkeit mit den Kanji eingeführt. Man kann nun ein Kanji anwählen, um es zu einer neuen Central Station werden zu lassen. Radikallinien, die auch in der neuen Ansicht vorhanden sind, sollen beim Übergang nach Möglichkeit nur minimal verändert werden, um die Orientierung des Nutzers zu unterstützen. Neue Radikallinien werden in das Layout eingefügt. Nicht mehr benötigte Radikallinien werden ausgeblendet. Eine Liste der bisher ausgewählten Central Station Kanji wird in der Central Station selbst angezeigt.

4.3 Codierung visueller Variablen

Zu Beginn der Designphase hat das Team die in der Vorlesung „Interaktive Informationsvisualisierung“ besprochenen visuellen Variablen auf die Tauglichkeit dieser für Tube-Map untersucht. Dabei fiel auf, dass viele der möglichen Variablen entweder nicht sinnvoll oder sogar nicht umsetzbar sind. Die visuellen Variablen Position, Ausrichtung, Form und Krümmung können für Codierung von Information verwendet nicht werden, da sie sehr stark von der Gestaltung der Tube-Map abhängen. Wie in 4.1 angedeutet, könnten X- und Y-Achse zwar für die Codierung der Strichzahl von Kanji verwendet werden, allerdings konnten diese im Layoutalgorithmus nicht zusätzlich umgesetzt werden. Hinzu kommt, dass zum Beispiel die Position von Stationen bei Tube-Maps bereits gegeben und dadurch fest ist und deshalb vom Nutzer nicht anders interpretiert wird.

Visuelle Variable	Verwendung
Linienlänge	keine, nicht charakteristisch für Tube-Maps
Linienbreite	Rahmen gesuchter Kanji: dicker; breite Linienbündel: mehr gemeinsame Radikale
Größe	Central Station: groß; Station: normal
Farbton	Radikal
Intensität	Filter
Verbindung	Zugehörigkeit zu Radikalen

4.4 Allgemeine Interaktionsaufgaben von Shneiderman

Die Interaktionsaufgaben werden von [Shneiderman, 1996] wie folgt benannt: Overview, Zoom, Filter, Details-on-demand, Relate, History, Extract. In den folgenden Untersektionen wird beschrieben wie die Tube-Map Visualisierung alle Interaktionsaufgaben umsetzt.

Overview

Nachdem der Nutzer in der ersten Ansicht ein für ihn interessantes Kanji gewählt hat, wird dieses als zentraler Punkt in einer Tube-Map gezeichnet. Nun ist es für ihn möglich durch Zoomen den maximalen Überblick über alle dargestellten Kanji und deren Verbindungen zu erhalten. In dieser Zoomstufe lassen sich einzelne Kanji nicht mehr ablesen, da sie zu klein sind, aber die Radikallinien lassen sich in ihrer Länge vergleichen und deren gesamter Verlauf besser nachvollziehen. Außerdem sieht er in welchem Bereich besonders viele Radikallinien zwischen den Stationen wechseln.

Zoom

In der Tube-Map ist es dem Nutzer möglich einen kleineren Teil der Visualisierung zu betrachten indem er hineinzoomt oder herauszoomt. Die genaueste Betrachtung der Beschaffenheit eines Kanjis ist auf der kleinsten Zoomstufe gegeben. Den größten Überblick erhält man auf größter Zoomstufe wie bereits in 4.4 beschrieben. Möchte der Nutzer den Ausschnitt der Visualisierung verschieben, kann er dies mit einer Pan Geste umsetzen. Auf diese Weise könnte er zum Beispiel eine Radikallinie verfolgen.

Filter

Am unteren Rand der Visualisierung befindet sich ein Formular. Mit diesem Formular können alle Kanji, die gerade angezeigt werden, nach den Kriterien Aussprache, Bedeutung, Strichanzahl, Schuljahr und Verwendungshäufigkeit gefiltert werden. Kanji, welche die Kriterien nicht erfüllen, werden transparenter dargestellt (Brushing). Verbindungen werden in diesem Modus nur angezeigt, wenn sie zwischen zwei Kanji besteht, welche nicht transparent sind. Wenn der Nutzer wieder zur Ausgangssituation zurückkehren möchte, aktiviert er den Reset Button, welcher Formulareingaben und Veränderungen in der Visualisierung rückgängig macht.

Details-on-demand

Mehr Informationen über ein spezifisches Kanji zu erhalten, ist auf mehreren Wegen möglich. Für ein schnelles und oberflächliches Informationsbedürfnis fährt der Nutzer über das Kanji und erhält dessen Bedeutung in einer Textbox neben diesem angezeigt. Durch einen einfachen Klick auf die Textbox kann er es wieder entfernen. Bei größerem Interesse klickt der Nutzer auf das Kanji und es wird in der Tabelle am unteren Bildschirmrand eingefügt. In der Tabelle sind nun alle Informationen über das Kanji ablesbar. Nach einem MouseOver über das Tabellenkanji erscheint ein Button zum entfernen des Kanjis aus der Tabelle. Um die aktuelle Tube-Map Ansicht nicht verschieben zu müssen, wurde die Tabelle scrollbar gemacht. Klickt der Nutzer doppelt auf ein Kanji zeichnet sich die Tube-Map mit diesem Kanji als zentrales Kanji neu.

Relate

Die Eigenschaften von Kanji können in der Tabelle am unteren Bildschirmrand verglichen werden. Siehe dazu [4.4](#). Theoretisch können Kanjis auch durch die Textboxen mit der Bedeutung des jeweiligen Kanjis verglichen werden. Praktisch allerdings gestaltet es sich durch die teilweise große Distanz zwischen den Kanji in der Tube-Map als schwierig, weswegen die Tabelle eine gute Alternative darstellt.

History

In einer Visualisierung soll es möglich sein, Aktionen rückgängig zu machen und zu vorherigen Ansichten zurückzugelangen. In der Tube-Map verändert nur das Wählen eines neuen Kanjis als zentralen Punkt der Visualisierung die Ansicht nachhaltig. Im unteren Bereich des zentralen Kanjis wird eine Liste aller Kanji angezeigt, die bereits zentrales Kanji waren. Wenn der Nutzer zu einem Kanji zurückkehren möchte, kann er in dieser Liste das entsprechende Kanji selektieren.

Extract

Im Abschnitt [4.4](#) wurde die Funktionalität des Formulars als Filter beschrieben. Einen anderen Effekt haben die Kriterien auf die Visualisierung, wenn statt Filtern die Suche gewählt wird. Kanji, die auf die Suchkriterien zutreffen, werden hervorgehoben. Der Nutzer hat also die Gelegenheit sowohl hervorhebende als auch ausblendende Interaktionen durchzuführen. Die ersten Suchergebnisse werden als Liste unter dem Formular angezeigt und durch Anklicken fokussiert, d.h. der Ausschnitt der Tube-Map verschiebt sich so dass das Kanji mittig positioniert ist. Anschließend wird es farbig hervorgehoben damit der Nutzer es schneller ausfindig machen kann.

5 IMPLEMENTIERUNG

Die Software ist eine vollständig Client-seitige Webanwendung. Diese Entscheidung wurde in Abwägung gegenüber C++ mit Qt 5 getroffen. Ausschlaggebende Argumente waren die Verfügbarkeit von CSS und SVG zur Darstellung im Gegensatz zu imperativen und objektorientierten Ansätzen sowie das einfache Zeigen und Teilen mit anderen Menschen über einen Link und einen Browser.

5.1 Entwicklungsumgebung

Mit der Entscheidung zur Webanwendung war die erste Frage, welche Programmiersprache die vorherrschende sein sollte. Zwischen JavaScript, CoffeeScript, Java und Haxe fiel die Wahl auf [CoffeeScript 1.6.2](#). Stark typisierte Sprachen, die zu JavaScript kompilieren wie Java und Haxe, haben das Problem, dass JavaScript Bibliotheken wie d3.js, einen abschreckenden Aufwand erfordern, die Interfaces zu definieren, so dass der Vorteil der starken Typisierung auch genutzt werden kann.

CoffeeScript ist JavaScript in sofern überlegen, da es ein Klassen-Pattern bereits in die Sprache integriert hat und auch vor anderen Unklarheiten in JavaScript schützt.

Zur Darstellung selbst wird [SVG 1.1](#) und [CSS 2010](#) verwendet, um mit Hilfe von CSS schnell visuell ansprechende Effekte und Transitionen nach Bedarf zu definieren und sie vom Browser bereits effizient umgesetzt werden. CSS Effekte sind jedoch nur auf DOM Elemente anwendbar, was wiederum SVG als Darstellungsframework verlangt. [d3.js 3.1.6](#) ist darauf ausgelegt, DOM-Elemente basierend auf Daten zu erstellen. Andere Darstellungsbibliotheken für Webanwendungen legen entweder ihren Fokus nicht auf SVG und CSS oder sind nicht datengebunden.

Zwischen Firefox 23 und [Chromium 28](#) fiel die Wahl auf letzteren, da der Chromium Browser SVG mit Text effizient darstellt. Der Firefox stößt hier an eine Grenze.

Weitere verwendete Technologien sind [require.js 2.1.6](#) um die Software in Module zu teilen, [reactor.js commit cdbf994](#) als marginale Abhängigkeit, die lediglich für einen simplen Fall ausprobiert wurde, [GNU Make 3.81](#) um typische Entwicklungsprozesse zu automatisieren und [Python 3.2.3](#) als lokaler Webserver.

5.2 Tube-Map Layout

Die Software für ein ansehnliches Layout für die Tube-Map zu implementieren, gestaltete sich als sehr schwierig. Die Doktorarbeit von [\[Stott, 2008\]](#) nahmen wir als Inspiration, jedoch eignete sich diese Arbeit lediglich für einen groben Ansatz. Im Detail mussten viele andere Entscheidungen getroffen werden und andere Ansätze ausprobiert werden.

Prinzipiell besteht der Layout Prozess aus zwei Phasen. Zuerst wird eine initiale Einbettung bestimmt. Dabei sollen Positionen der Knoten bestimmt werden und wie diese untereinander verbunden sind. Danach wird der entstandene Graph optimiert. Dies geschieht durch eine kontinuierlich Bewertung und Veränderung um eine bessere Bewertung zu erzielen. Die Bewertung richtet sich nach mehreren Kriterien und Regeln.

Konkret ist das Ergebnis der initialen Einbettung, dass das zentrale Kanji der Mittelpunkt ist. Die Radikale des zentralen Kanji werden hier als relevante Radikale bezeichnet. Von dem zentralen Kanji ausgehend, denkt man sich für jedes relevante Kanji Strahlen in 90° und 45° Winkel zueinander. Für jedes relevante Radikal werden die Kanji genommen, die dieses Radikal enthalten und eingeteilt in solche, die von allen relevanten Radikalen nur eins enthalten (Lo-Kanji), und solche, die mehrere relevante Radikale enthalten (Hi-Kanji). Hi-Kanji werden unter den Strahlen verteilt, Lo-Kanji am entsprechenden Strahl hinten angefügt. Alle Kanji haben nun eine Position. Um Kanji miteinander zu verbinden, werden für jedes relevante Radikal Verbindungen nacheinander gezogen. Vom zentralen Kanji aus werden alle Hi-Kanji des zum Radikal zugehörigen Strahls verbunden und anschließend wird jeder Strahl durchgegangen und alle dortigen Hi-Kanji, die das aktuelle Radikal enthalten, nacheinander verbunden. Zum Schluss wird noch eine Verbindung zu den dazugehörigen Lo-Kanji gezogen.

Die initiale Einbettung ist ausgesprochen wichtig für eine erfolgreiche Optimierung. Die Optimierung nach mehreren Kriterien wird von [\[Stott et al., 2011\]](#) beschrieben. Ein Teil dieser Optimierung ist in der vorliegenden Software implementiert, jedoch in entscheidenden Aspekten abgeändert um schneller zu einem guten Ergebnis zu kommen, die Entwicklung zu vereinfachen und auf spezielle Bedürfnisse der Kanji-Tube-Map einzugehen. Die Optimierung findet parallel in einem [Worker](#) statt um die Interaktivität der Anwendung nicht zu behindern. Kontinuierlich wird ein Knoten im Graph ausgewählt und geprüft ob in der Umgebung eine Position für den Knoten ist, bei der die Bewertung des Graphen besser wird. In die Bewertung fließen Regeln und Qualitätskriterien ein. Weniger Regeln zu verletzen ist stets besser als eine höhere Qualität zu erreichen.

Folgende Regeln sollen möglichst nicht verletzt werden:

- keine Kanten, dürfen unter einem Knoten hindurchführen, die nicht zu dem Knoten selbst gehören (`wrongEdgesUnderneath`)
- der Knoten darf dem Knoten für das zentrale Kanji nicht zu nahe kommen (`tooNearCentralNode`)

- keine Kanten, die andere Kanten kreuzen (`edgeCrossings`)

Folgende Qualitätskriterien flossen in die Version der Software ein, die zur finalen Präsentation gezeigt wurde:

- aneinander liegende Kanten sollen möglichst gerade sein (`lineStraightness`)
- Kanten sollen möglichst kurz sein (`lengthOfEdges`)

LITERATURVERZEICHNIS

- [DVB-AG, 2011] DVB-AG (2011). <http://www.dvb.de/de/Die-DVB-AG/Zahlen-Daten/>.
- [O'Reilly, 2003] O'Reilly (2003). <http://www.oreilly.de/artikel/routemap.pdf>.
- [Rose, 2010] Rose, H. (2010). Kanji learning of Japanese language learners on a year-long study exchange program at a Japanese university: An investigation of strategy use, motivation control and self-regulation. PhD thesis, University of Sydney.
- [Shneiderman, 1996] Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In Visual Languages, 1996. Proceedings., IEEE Symposium on, pages 336–343. IEEE.
- [Stott et al., 2011] Stott, J., Rodgers, P., Martinez-Ovando, J. C., and Walker, S. G. (2011). Automatic metro map layout using multicriteria optimization. Visualization and Computer Graphics, IEEE Transactions on, 17(1):101–114.
- [Stott, 2008] Stott, J. M. (2008). Automatic Layout of Metro Maps using Multicriteria Optimisation. PhD thesis, University of Kent.