

3. Workshop Historische Netzwerkforschung,

Wien, 13./14.11.2010

Vortrag Visualisierung dynamischer Netzwerke mit VISIONE

Dr. Ulrich Eumann, NS-Dokumentationszentrum der Stadt Köln

Gliederung

1 Ansätze zu einer Dynamischen Netzwerkvisualisierung

1.1 Statische Darstellung dynamischer Netzwerke

1.2 Animation mit Pajek und dem SVG-Format

1.3 Animation mit SoNIA, NEVADA, dyNet und Gephi

2 Animation dynamischer Netzwerke mit VISIONE

2.1 Bedienung von VISIONE

2.2 Abfilmen der Animation mit Screencast-Programm

3 Vor- und Nachteile verschiedener Verfahren

1 Ansätze zu einer Dynamischen Netzwerkvisualisierung

Netzwerkgrafiken sind normalerweise statisch, Momentaufnahmen zu einem bestimmten Zeitpunkt. Manchmal sind wir aber mehr an der *Entwicklung* eines Netzwerks interessiert als an einem bestimmten *Zustand*; in unserem Kölner Forschungsprojekt zum Beispiel an der Entwicklung der Kenntnis der Kölner Gestapo von bestimmten Widerstandsgruppierungen. Das Medium der Animation sollte aber auf Fälle beschränkt werden, in denen die spezifischen Vorteile der Animation gegenüber der statischen Darstellung zum Tragen kommen. Schließlich passen Animationen nicht sonderlich gut zu den üblichen Medien historiografischer Wissensproduktion. Der Vortrag besteht aus 3 Abschnitten, der mittlere Abschnitt ist der Hauptteil.

1.1 Statische Darstellung dynamischer Netzwerke

Dieses Verfahren eignet sich nur für ganz bestimmte Netzwerkstrukturen und ganz bestimmte Fragestellungen. Je mehr Knoten desto unübersichtlicher wird die Grafik.

Unser Beispiel ist ein Aussagenetzwerk, in dem die waagerechten Positionen der Knoten die Tage anzeigen, an denen der Name eines Mittäters zum ersten Mal fällt. Auf diese Weise können wir den Ermittlungsfortschritt deutlich machen, zeigen (etwa am Beispiel Paul

Matenia), wer wie lange seine Widerstands-Genossen geschützt hat und wessen Verurteilung oder Verhaftung auf wessen Aussage zurückgeht.

Um dort hin zu gelangen, müssen wir das von einem Layout-Algorithmus erzeugte Netzwerk zerstören, da wir die X-Achse der Visualisierung zweckentfremden müssen, um über sie den Zeitablauf zu visualisieren. Dazu rechnen wir die Abstände zwischen den Zeitpunkten in X-Koordinaten um. Diese können wir dann in der Net-Datei mit den dort berechneten X-Koordinaten austauschen.

1.2 Animation mit Pajek und dem SVG-Format

Der Königsweg zur dynamischen Visualisierung ist aber die Animation.

Ihre Grundlage sind die einzelnen „Frames“, Einzelbilder, unserer zeitlich aufeinander folgenden Netzwerk-Zustände, die zu einer Animation verbunden werden, in der Knoten und Kanten zu dem Zeitpunkt auf dem Bildschirm auftauchen oder verschwinden, an dem die dazu gehörigen Personen im historischen Netzwerk aufgetaucht oder verschwunden sind. Mit **Pajek** (<http://pajek.imfm.si>) und **Inkscape** (<http://www.inkscape.org>) kann man relativ einfach eine dynamische Netzwerk-Animation als Präsentation erstellen. Man erstellt eine Grafik für das Gesamtnetzwerk und löscht in **Inkscape** die von **Pajek** als verschiedene **Classes** gespeicherten einzelnen Zeiträume in rückwärtiger Richtung aus dem Netzwerk. Die dabei entstehenden Zeitpunkt-Visualisierungen lassen sich dann mit **PowerPoint** oder **Impress** zu einer Präsentation verbinden. Mit dem freien Open-Source Animationsprogramm **Pencil** (<http://www.pencil-animation.org/>) lässt sich aus den einzelnen Graphen eine Animation erstellen, die in das verbreitete Flash-Format exportiert werden kann.

Die weitaus elegantere Lösung ist die *Animation mit SVG*. Ihr Hauptnachteil liegt darin, dass es bis auf den Nischen-Browser Opera kaum Programme gibt, die die SVG-Animationen ohne besondere Plug-ins darstellen können. Konverter von SVG-Animationen in das Flash-, AVI- oder GIF-Format habe ich noch keine entdecken können (vgl. http://de.wikipedia.org/wiki/Scalable_Vector_Graphics).

Die Animation mit SVG ist recht aufwändig. Nachdem **Pajek** das Gesamtnetzwerk erstellt und als SVG-Datei exportiert hat, müssen wir im **XML-Editor** jedes Element *einzel*n animieren, das heißt jede Zeiteinstellung eines Elements per Hand eintragen.

Es lässt sich allerdings Einiges über ein paar Suchen- und Ersetzen-Vorgänge automatisieren.

Die animationsrelevanten SVG-Befehle `animate` und `set` umfassen eine Vielzahl von Anweisungen, zum Beispiel den Startzeitpunkt einer Animation, ihre Dauer oder auslösende Ereignisse (z.B. Mausklick). Sie beziehen sich auf bestimmte Eigenschaften von Elementen, etwa auf den Radius unseres Knotens in vertikaler Richtung, und ändern die darin festgehaltenen Werte über die Attribute `from` und `to`, die den Zustand des Elements am Beginn und am Ende der Animation beschreiben. Der Browser berechnet dabei die Zwischenstufen für eine flüssige Animation während der Durchführung selbst.

Im Detail habe ich das Vorgehen in meinem Manual zur Netzwerkvisualisierung beschrieben, das ich gern Interessierten zur Verfügung stelle.

1.3 Animation mit SoNIA, NEVADA, dyNet und Gephi

Ein weiterer Weg zur Animation dynamischer Netzwerke ist das auf JAVA aufsetzende Package **SoNIA** (Social Network Image Animator) (<http://www.stanford.edu/group/sonia/>). SoNIA kann mit den NET-Dateien arbeiten, die **Pajek** produziert. Die Einträge zu den Knoten müssen allerdings in eckigen Klammern um Angaben über den Zeitraum ergänzt werden. Der hinzugefügte Ausdruck [21-39] bedeutet zum Beispiel, dass ein Knoten in der 21. Zeiteinheit in der Animation auftaucht und bis zur 39. Zeiteinheit zu sehen bleibt.

SoNIA-Animationen lassen sich in das Flash- und das QuickTime-Format exportieren. Die Netzwerke, die **SoNIA** animiert sind allerdings recht filigran, die Elemente werden nicht sanft animiert, sondern abrupt eingeblendet. Außerdem lässt sich kein Titel einfügen, statt dessen muss die laufende Uhr oben links hingenommen werden.

NEVADA (Network Visualization and Analysis for Dynamic Networks) setzt ebenfalls auf JAVA auf und ist eine Entwicklung einer Informatiker-Projektgruppe am Lehrstuhl für Software-Technik der Uni Trier (<http://www.st.uni-trier.de/gd/>). Durch sieben verschiedene Menüs lassen sich die Eigenschaften von Knoten, Kanten und Layouts festlegen und die Analyse bestimmter Werte durchführen.

Die Visualisierung dynamischer Netzwerke lässt sich nur mit dem Import einer NET-Datei starten, weil dann überhaupt erst ein Button angezeigt wird, der das Hinzuladen weiterer Sequenzen erlaubt, den späteren Frames unserer Animation. Jeder Frame unserer Animation muss daher einzeln in **Pajek** erstellt werden und dann nach **NEVADA** geladen werden.

Die fertige Animation kann dann – nicht ganz problemlos - in das SVG-Format exportiert werden. Da es keine Exporteinstellungen gibt, kann man z.B. die Labels der Knoten nicht exportieren. Angesichts gravierender Bugs im Programm und der letzten Beta-Version vom März 2009 ist eher von einer Verwendung abzuraten.

Außerdem gibt es eine „pilot version“ von **dyNet**, einem neuen Software-Projekt zur „dynamic network visualization“ (<http://www.markowetzlab.org/software/networks.html>). Leider wird hier wieder ein neues XML-basiertes Datenformat eingeführt (XGMML). Die Informationen auf der Website sind sehr knapp, das Programm nicht intuitiv bedienbar. Es soll aber demnächst ein Aufsatz des Programmierers erscheinen. Vielleicht sieht man dann, in welche Richtung es geht.

Auch **Gephi**, ein französisches Visualisierungsprojekt auf JAVA-Basis (<http://gephi.org/>), kann dynamische Netzwerke visualisieren. Das Programm, das jetzt in der Version 0.7 beta vorliegt, verwendet für die Animation Dateien im GEXF-Format (Graph Exchange XML Format), in denen Start- und Endpunkt der Zeitintervalle angegeben werden, in denen bestimmte Knoten vorhanden sind.

2 Animation dynamischer Netzwerke mit VISIONE

Wir erfassen bei der Auswertung von Gestapoakten als ein Attribut der Verbindungen den Zeitpunkt oder Zeitraum, an dem diese stattfand. Da die Gestapo sich nur für die Zeitpunkte einiger spezifischer Aktivitäten interessiert hat, wir aber von kontinuierlichen Widerstandsverbindungen ausgehen, müssen wir hochrechnen: Liegen für mehr als 50 Prozent der Monate zwischen dem ersten und dem letzten Monat einer Verbindung Daten vor, gehe ich von einer permanenten Verbindung aus und ergänze die fehlenden Monate. Aus der Excel-Datei mit al-

len – vorliegenden und hochgerechneten - Verbindungen erstellen wir dann die Netzwerke der einzelnen Monate.

2.1 Bedienung von VISIONE

Seit der Version 2.6 von **VISIONE** – einem Visualisierungsprojekt der Uni Konstanz (<http://visone.info/>), gibt es einen Layout-Algorithmus: *Dynamic Layout* und damit die Möglichkeit der Animation dynamischer Netzwerke. Grundlage für die Animation sind auch hier Frames mit verschiedenen Stadien eines Netzwerks. Das Vorzugsformat der Frames ist GraphML, ein XML-basiertes Dateiformat für die Repräsentation von Graphen, VISIONE verarbeitet aber auch Gedcom-, Net- oder DL-Dateien. Ein anderer Weg bestünde darin, in VISIONE ein Netzwerk vom Startzeitpunkt ausgehend aufzubauen und für jeden Frame eine GraphML-Datei zu speichern.

Die von uns geladenen Frames verknüpfen wir über das zweitrechte Icon in der Menüleiste („opens the network collection manager“) zu einer Network Collection. In dem Menü **Network Collections** vergeben wir einen Namen und wählen aus einer Liste von Dateien diejenigen aus, die gemeinsam animiert werden sollen.

An diesem Punkt stellt sich die grundsätzliche Frage, ob die Position der Knoten über die Animation hinweg eher stabil sein soll oder ob jedes Frame-Netzwerk sein eigenes Layout behalten soll. Wenn das entschieden ist, wählt man für die geladenen Frame-Netzwerke unter dem Reiter **visualization** das Node Layout *Dynamic Layout*. Hier lässt sich unter **stability** die Stabilität der Knotenposition zwischen null und hundert Prozent festlegen. Hier wird außerdem entschieden, welcher Referenztyp das Layout der einzelnen Frame-Netzwerke bestimmt, das Frame-Netzwerk mit den meisten Knoten beispielsweise oder das vereinte Gesamtnetzwerk (*aggregation network*), das man sich auch gesondert anzeigen lassen kann. Dann klickt man auf **layout!** und alle Frame-Networks werden gleichförmig angeordnet.

Wenn alle Frames geladen und verknüpft sind, kann man mit dem Icon ganz rechts („animate between successive networks“) die Animation erstellen lassen, die in einem gesonderten Fenster abläuft. Das Animationsfenster zeigt oben links die Bezeichnung der Network Collection. Unten links lässt sich die Geschwindigkeit des Ablaufs verändern. Daneben haben wir zwei Buttons für das Durchgehen der einzelnen Frames vorwärts und rückwärts und den Button mit dem sich die Animation starten lässt. Weiter rechts sehen wir die Nummer des gerade dargestellten Frames und die Gesamtanzahl der Frames.

2.2 Abfilmen der Animation mit Screencast-Programm

Die Animation lässt sich leider nicht speichern oder exportieren. Man braucht vorerst noch ein Programm mit Screencast-Option (Bildschirm-Aufnahme), um die Animation abzufilmen. Dafür hat sich **CamStudio**, das relativ einfache und daher auf vielen Rechnern gut arbeitende Open-Source-Screencast-Programm als brauchbar erwiesen (<http://camstudio.org/>). Man klickt auf **Region** und wählt den ganzen Bildschirm oder eine in Pixel-Koordinaten anzugebende Region, die abgefilmt werden soll, klickt auf den großen roten Punkt und lässt dann in **VISIONE** die Animation ablaufen. Am Ende der Animation klickt man auf das große blaue Quadrat und stoppt die Animation. **CamStudio** erstellt dann eine AVI-Datei, die in jedem Media-Player laufen sollte, kann aber das Video auch in das Flash-Format exportieren.

Mit Programmen wie dem **AVI-Splitter** (<http://www.brizsoft.com/avisplit/>) kann man das Video hinterher schneiden. Der Export von Animationen steht bei den Programmierern von **VISIONE** bereits auf der Liste der *planned features*.

3 Vor- und Nachteile verschiedener Verfahren

Der Nutzen einer *statischen Darstellung* ist begrenzt. Für Personen, die an Spring-Embedder-Layouts gewöhnt sind, ist die Grafik zunächst vielleicht eher verwirrend als erhellend. Intuitiv erschließt sich die Visualisierung nicht so leicht wie ein Netzwerk-Graph. Sie hat dennoch ihre Vorzüge, jedenfalls so lange, wie wir in der Geschichtswissenschaft noch vorrangig über bedrucktes Papier kommunizieren.

Die Grenzen der Visualisierung dynamischer Netzwerk mit Programmen wie **SoNIA**, **Nevada** oder **dyNet** liegen auf der Hand: Die Anzahl der Knoten ist sehr begrenzt, die optischen Gestaltungsmöglichkeiten sehr gering. **Nevada** enthält darüber hinaus zahlreiche Bugs und wird nicht mehr weiterentwickelt. Die Entwicklung von **dyNet** werde ich allerdings weiterverfolgen und es lohnt sich, alle paar Monate zu schauen, ob auf der Softwareseite der INSNA-Homepage (<http://www.insna.org/software>) neue Programme angekündigt werden.

Die Vorgehensweise mit **Pajek** und SVG bietet sicherlich die meisten Gestaltungsoptionen, was allerdings durch einen enormen Zeitaufwand erkauft wird. **VISONE** ist demgegenüber ein rundes Paket mit einem sehr guten Verhältnis von Aufwand und Ertrag. Leider lassen sich zu den einzelnen Graphen in **VISONE** meines Wissens keine Attributdateien zuladen. Vor allem vermisst man natürlich eine Schnittstelle für den Export von Netzwerkanimationen. Wenn man neben einem Umwandeln der Animation in das Flash-Format auch eine Ausgabe in das SVG-Format programmiert, hätten wir eine beinahe rundum gelungene Software für die Visualisierung dynamischer Netzwerke.

Wer als Historiker den Zeitfaktor ernst nimmt, muss sich überlegen, seine Visualisierungen in Bewegung zu bringen. Das schränkt natürlich ihren Anwendungsbereich ein, so lange bedrucktes Papier das Hauptkommunikationsmittel ist. Da wir für unsere große Widerstandsausstellung im Jahr 2015 einen Monitor planen, in dem wir die Wellen von Zerschlagung und Wiederaufbau der KPD als dynamische Netzwerk-Animation präsentieren wollen, werden wir uns wahrscheinlich noch länger mit den hier aufgeworfenen Fragen befassen. Aber auch sonst kann die intellektuelle Auseinandersetzung mit der Modellierung des Faktors Zeit in Netzwerken sehr inspirierend sein, wie wir hier ja nun zwei Tage lang erlebt haben.