

SOCIAL NETWORK DAY

ORGANISATIONALE NETZWERKANALYSE

www.ansna.org/konferenz

31. Mai 2008
TU Wien

SoNIA – Social Network Image Animator

Martina Salzl
0105988

Abstract: Die Arbeit wird im Rahmen der Lehrveranstaltung „Theorie und Praxis der Gruppenarbeit“, an der TU-Wien, verfasst. Im Zuge dieser Ausarbeitung wird das Projekt „SoNIA“ der Stanford-University vorgestellt. Zu Beginn der Arbeit wird der „Social Network Image Animator“ vorgestellt. Dabei werden auf die Anwendung und auf die daraus gewonnen Dateien speziellen eingegangen. Für den Einsatz ist es unerlässlich, die zugrundeliegenden Eingabeformate detailliert zu erläutern. Des Weiteren soll durch das Lesen dieser Arbeit, die Anwendung des Tools für Einsteiger erleichtert werden.

1 Einleitung

Bei SoNIA handelt es sich um ein auf Java basierendes Tool zur Visualisierung von dynamischen Netzwerkdaten. Es wird von dynamischen Netzwerkdaten gesprochen, falls zusätzlich zu den Beziehungsinformationen zwischen den einzelnen Einheiten noch weitere Informationen berücksichtigt werden. Als diese könnte unter anderem die relative Reihenfolge angesehen werden.

Das Ziel, welches durch SoNIA verfolgt wird ist, dass die dynamischen Netzwerkinformation mittels unterschiedlicher Datenformate einzulesen und diese in weiterer Folge graphisch aufzubereiten. Die Aufbereitung der Daten kann zum Beispiel als animierte Darstellung oder als Bild exportiert werden. Die dafür eingelesenen Datensätze können anhand unterschiedlicher Parameter verändert werden.

1.1 Problemstellung

Das Ziel dieser Arbeit ist es, dass Tool „SoNIA“ vorzustellen und die Funktionsweise des Tools anhand eines kleinen Anwendungsbeispiels zu zeigen. Dieses soll den zukünftigen Usern den Einstieg in die Arbeit mit „SoNIA“ erleichtern. Um dies zu ermöglichen ist der Aufbau des Tools zu erläutern, der Aufbau der Input-Files muss erläutert werden und die verschiedenen Ausgabeformate müssen erläutert werden.

2 Anwendung des Tools

Innerhalb dieses Abschnittes wird die Anwendung des Tools beschrieben.

2.1 Input-Files

Bei der Entwicklung des Tools wurde darauf geachtet, dass das Einlesen von sämtlichen Datenformate ermöglicht wird. Im Rahmen der Ausarbeitung wird dabei auf zwei wichtigsten Formate genauer eingegangen.

2.1.1 .son

Das .son-Format soll die Einschränkungen des .net-Formates ausgleichen und es soll des Weiteren das Speichern und das Importieren der Netzwerkattribute erleichtern. Zusätzlich ist dieses Format darauf ausgerichtet, dass es möglichst einfach ist, aus anderen Anwendungen Exportfiles zu generieren. Das Import-File ist in Form einer Liste aufgebaut, wobei diese zwischen Knoten und Kanten getrennt wird. Die Einträge für jeden Datensatz sind durch einen Tabulatorabstand zu trennen und sie werden durch die jeweiligen Spaltenüberschriften definiert. Es ist zusätzlich darauf hinzuweisen, dass sämtliche im Inputfile enthaltenen Daten optional sind und die Reihenfolge der Datensätze nicht festgelegt ist. Die erforderlichen Attribute sind durch einen identifizierenden Tag und Zeit-Koordinaten für den entsprechenden Event versehen. Dadurch handelt es sich um flexibles Format.

Die Idee des .son-Parsers ist es, dass der Aufbau des Files auf den Spalten basiert und nicht wie im .net-Paser auf Token. Aufgrund dessen ist dieser Filetyp einfacher zu erstellen und die Konvertierung von Datensätzen in diesen Filetyp wird vereinfacht. Allerdings ist dieses Format mühsamer zu schreiben. SoNIA bietet die Möglichkeit zum Speichern der Daten in diesem Format.

Die ersten Zeilen des Input-Files können Kommentare enthalten, welche durch „/“ gekennzeichnet sind. Diese Kommentare werden in weiterer Folge in das Log-File übernommen. Im Anschluss an die Kommentare werden die Spaltenüberschriften, welche der Knotenbenennung entsprechen, definiert. Diese sind durch einen Tabulatorabstand voneinander abzugrenzen. Das erste Element muss „NodeID“ bzw. „AlphaID“ genannt werden. Durch dieses Element findet der Parser die entsprechende Zeile. Das Element „NodeID“ entspricht einem Schlüsselement, welches zum Darstellen von den Beziehungen zwischen den Knoten und Kanten benötigt wird. Wird im .son-File NodeID verwendet, so ist dem Knoten ein Integer-Wert zuzuweisen. Es besteht des Weiteren die Möglichkeit die Identifizierung mittels des Elements „AlphaId“, wobei diesem Element ein String zugewiesen werden kann. Die Ausgangsbasis für die Übersetzung des .son-Files stellt die Definition der Spalten dar. Kann ein File nicht entsprechend dieser Angaben geparkt werden, so wird eine Fehlermeldung ausgegeben. Alle weiteren Bezeichnungen können in einer beliebigen Reihenfolge erfolgen bzw. müssen diese nicht im File berücksichtigt werden. Es ist allerdings darauf hinzuweisen, sobald ein Element definiert wurde, muss dieses in den Datensätzen angegeben werden, das heißt, in jede Spalte muss ein Wert eingetragen werden und darf nicht frei gelassen werden.

Nachdem die Knotenbezeichnungen und deren Datensätze definiert wurden, können die Kanten definiert werden. Hierbei ist darauf zu achten, dass das erste Element mit „FromId“

bezeichnet wird. Somit erkennt der Parser, dass die Definition der Knoten abgeschlossen ist. Alle weiteren Einträge unterliegen keiner bestimmten Reihenfolge.

Im Anschluss an die theoretische Ausarbeitung des .son-Files wird ein Beispiel gezeigt.

2.1.2 Beispiel eines .son-Files

NodeId	Label	StartTime	EndTime	NodeSize	NodeShape	ColorName	BorderWidth	BorderColor
1	129473	0.0	42.0	5.0	square	black	1.5	
2	129047	0.0	42.0	5.0	square	black	1.5	
3	132996	0.0	42.0	5.0	square	black	1.5	
4	145242	0.0	42.0	5.0	square	black	1.5	
5	127535	0.0	42.0	5.0	square	black	1.5	
6	127319	0.0	42.0	5.0	square	black	1.5	
7	129801	0.0	42.0	5.0	square	black	1.5	
8	104456	0.0	42.0	5.0	square	black	1.5	

FromId	ToId	StartTime	EndTime	ArcWeight	ArcWidth	ColorName
2	1	0.135	0.135	0.2	1.6	black
4	2	0.135	0.135	0.2	1.6	black
2	4	0.135	0.135	0.2	1.6	black
4	3	0.135	0.135	0.2	1.6	black
2	6	0.135	0.135	0.2	1.6	black
6	5	0.135	0.135	0.2	1.6	black
8	7	0.135	0.135	0.2	1.6	black
4	8	0.135	0.135	0.2	1.6	black
2	8	0.135	0.135	0.2	1.6	black
4	1	0.135	0.135	0.2	1.6	black
2	1	0.135	0.135	0.2	1.6	black
4	2	0.135	0.135	0.2	1.6	black
6	5	3.514	3.514	0.2	1.6	black
6	6	3.514	3.514	0.2	1.6	black
6	7	3.514	3.514	0.2	1.6	black
8	1	3.649	3.649	0.2	1.6	black

Im Anschluss auf das .son-Eingabeformat wird auf das .net-Format übergegangen.

2.1.3 .net

Der .net-Parser wurde entwickelt um Textfiles, welche in Pajek¹ erstellt wurden, zu übersetzen. Es handelt sich dabei um eine Kanten/Knotenliste. Innerhalb eines Files müssen sowohl die Kanten als auch die Knoten definiert werden.

Das .net Format wurde umgesetzt, damit das „Zeit-Intervall-Format“ in beiden Programmen verwendet werden kann. Es ist anzumerken, dass das .net-Format keine Flexibilität im Zeitverlauf bietet. Allerdings ist es relativ einfach und es enthält einige Möglichkeiten zur Kontrolle der Attribute. Zurzeit können nur wenige dieser Funktionen in SoNIA umgesetzt werden. Aus diesem Grund wird auch das .son-Format empfohlen.

Das .net-File wird in zwei oder drei Abschnitte unterteilt. Der erste Abschnitt beginnt mit „*Vertices n“, welcher die Knoten beschreibt. Der zweite und dritte Abschnitt startet mit „*Edges“ für ungerichtete und „*Arcs“ für gerichtete Kanten.

Der Header für den ersten Abschnitt enthält die Anzahl der Knoten im entsprechenden File und im Anschluss werden zeilenweise die Knoten aufgelistet. Jede Zeile hat dabei mit einer Integer-ID zu beginnen. Werden für die Bezeichnung „Strings“ eingesetzt, so sind diese unter Anführungszeichen zu setzen. Die Zeitinformation wird in einer eckigen Klammer definiert. Innerhalb der Klammer werden die folgenden Zeichen eingesetzt:

„-“, Teilung von niedrigen und höheren Zeitintervallen
„“ Teilung der einzelnen Intervalle
„*“ unendliches Zeitintervall

Im Anschluss an die Erläuterung zum File-Aufbau wird ein Beispiel gezeigt:

```
*Vertices 3
1 "a" c blue [5-10,12-14]
2 "b" c red [1-3,7]
3 "e" c green [4-*]
*Edges
1 2 1 c gray [7]
1 3 1 c red [6-8]
```

2.2 Beispiel

Nachdem die dahinterstehende Theorie behandelt wurde, wird im Anschluss ein Beispiel mit den dazugehörigen Schritten gezeigt.

¹ Bei Pajek handelt es sich um eine auf Windows-basierende Freeware, welche unter <http://vlado.fmf.uni-lj.si/pub/networks/pajek/> zum Download bereitgestellt wird.

2.2.1 Inputfile im .son-Format

NodeId	Label	StartTime	EndTime	NodeSize	NodeShape
	ColorName	BorderWidth	BorderColor		
1	129473	0.0	42.0	5.0	square black 1.5
2	129047	0.0	42.0	5.0	square black 1.5
3	132996	0.0	42.0	5.0	square black 1.5
4	145242	0.0	42.0	5.0	square black 1.5
5	127535	0.0	42.0	5.0	square black 1.5
6	127319	0.0	42.0	5.0	square black 1.5
7	129801	0.0	42.0	5.0	square black 1.5
8	104456	0.0	42.0	5.0	square black 1.5
FromId	ToId	StartTime	EndTime	ArcWeight	ArcWidth
	ColorName				
2	1	0.135	0.135	0.2	1.6 black
4	2	0.135	0.135	0.2	1.6 black
2	4	0.135	0.135	0.2	1.6 black
4	3	0.135	0.135	0.2	1.6 black
2	6	0.135	0.135	0.2	1.6 black
6	5	0.135	0.135	0.2	1.6 black
8	7	0.135	0.135	0.2	1.6 black
4	8	0.135	0.135	0.2	1.6 black
2	8	0.135	0.135	0.2	1.6 black
4	1	0.135	0.135	0.2	1.6 black
2	1	0.135	0.135	0.2	1.6 black
4	2	0.135	0.135	0.2	1.6 black
6	5	3.514	3.514	0.2	1.6 black
6	6	3.514	3.514	0.2	1.6 black
6	7	3.514	3.514	0.2	1.6 black
8	1	3.649	3.649	0.2	1.6 black

2.2.2 Laden des Files und Erstellen eines neuen Layouts

Das Laden der Files erfolgt durch den Befehl „Load Files“. Entspricht das File nicht den Anforderungen, so wird eine Fehlermeldung ausgegeben. Nachdem das File geladen wurde, kann das Layout erstellt werden. Dies erfolgt über den Befehl „Create Layout“. Die beiden Befehle sind der unten gezeigten Abbildung durch einen Kreis gekennzeichnet.

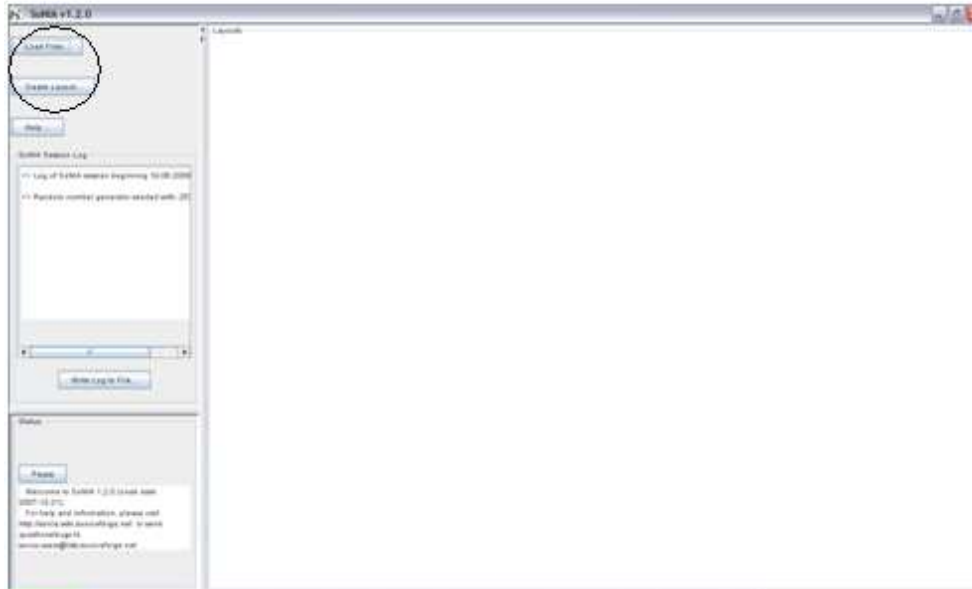


Abbildung 1: Startmaske

Wird ein neues Layout erstellt, so öffnet sich die kommende Maske. Hierbei ist unter anderem die Start- und Endzeit und die Dauer anzugeben. Des Weiteren kann der Animationstyp und der Layouttyp definiert werden.

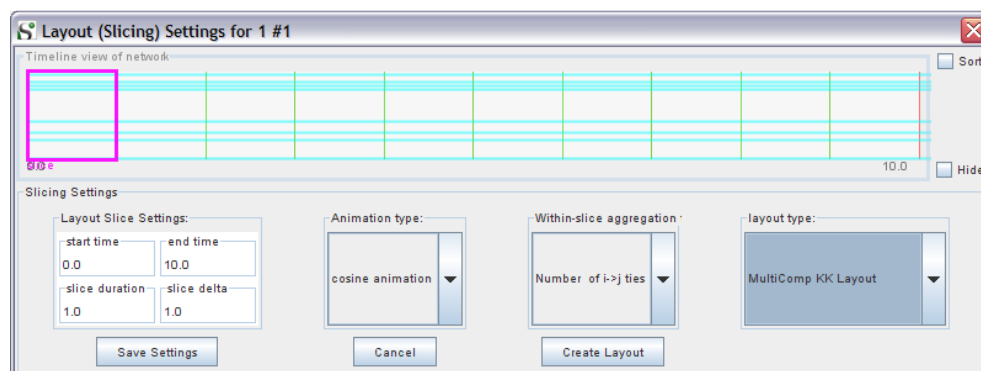


Abbildung 2: Layout erstellen

Hinweis: Für das Erstellen des Layouts ist es wichtig, dass eine Endzeit angegeben wird.

Nachdem die das Layout definiert wurde, öffnet sich die kommende Maske:

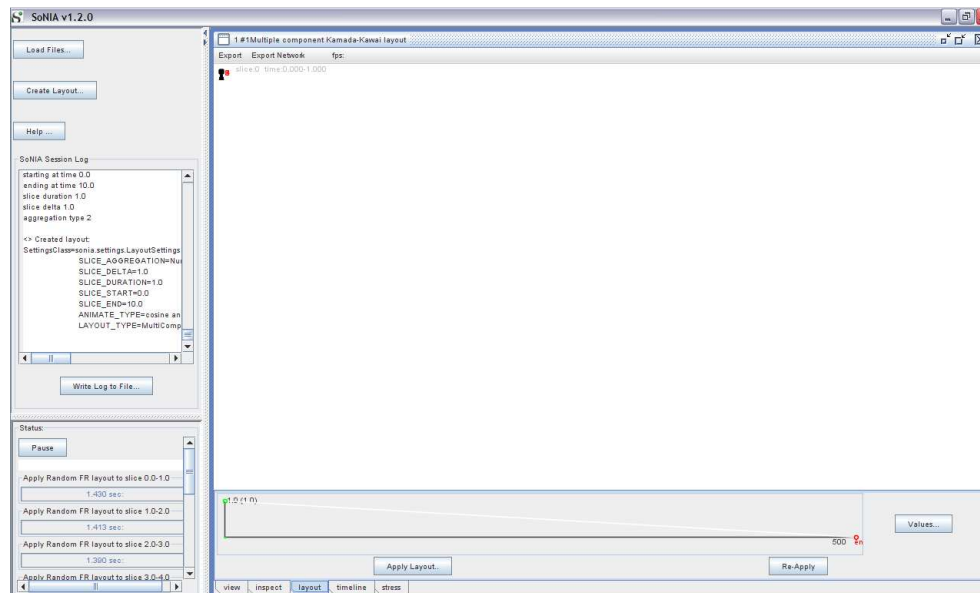


Abbildung 3: Startmakse

2.2.3 Graphikeinstellungen

Die Einstellungen für die Graphiken werden im Tab „View“ vorgenommen. Klickt man innerhalb dieses Tabs auf „View Options“, so öffnet sich die unten gezeigte Abbildung:

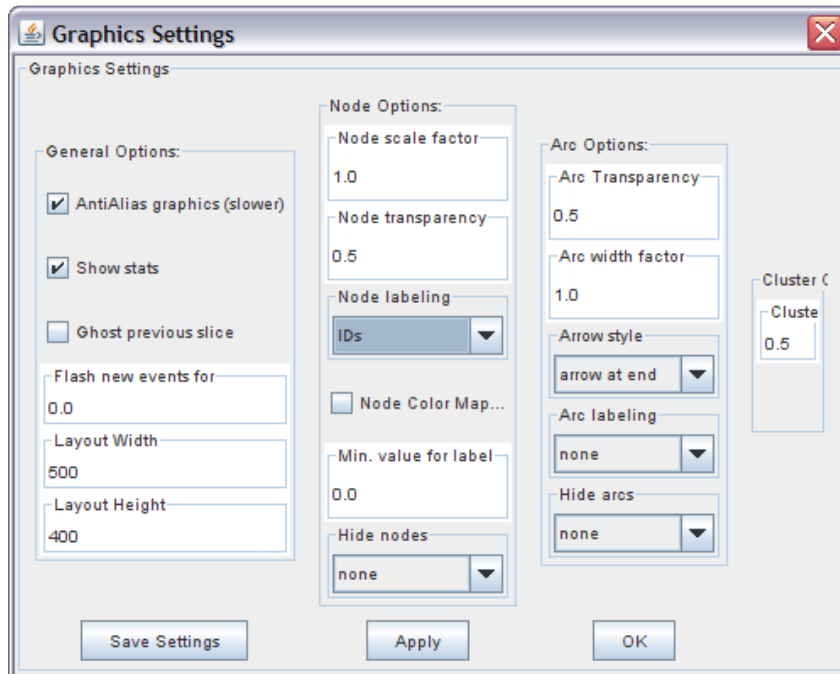


Abbildung 4: Graphikeinstellungen

Innerhalb dieser Maske können sämtliche graphische Einstellungen getätigt werden. So kann hier unter anderem die Größe des Fensters und die Bezeichnung der Knoten und Kanten definiert werden. Auch kann festgelegt werden, wie die Pfeile auszusehen haben oder ob die Knoten und Kanten versteckt werden sollen.

Layout

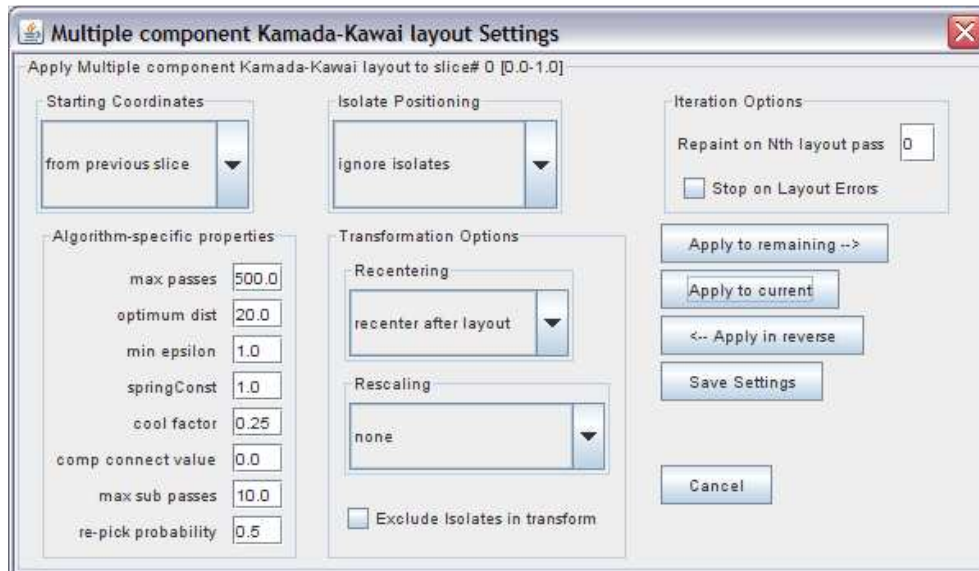


Abbildung 5: Layouteinstellungen

Innerhalb des Tab „Layout“ sind die Einstellungen zum Layout vorzunehmen. So sind hier unter anderem die Startkoordinaten, die Skalierung und die Positionierung festzulegen. Nachdem die gewünschten Parameter festgelegt wurden, können diese mit „Apply to current“ der aktuellen Zeitspanne übergeben werden.

Zeitspanne

Innerhalb des Tabs „Timeline“ werden die einzelnen Slides angezeigt. Soll ein Slide geändert werden, so ist dieser auszuwählen und zum Tab „Layout“ zu wechseln. Nachdem alle Slides ein definiertes Layout besitzen, kann die Datei exportiert werden. Dies kann zum einen zweidimensional bzw. dreidimensional erfolgen.

Export Matrix

```

10 Matricies from layout 1 #1
8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0

```

8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0

8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0

8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0

8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0

8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0

```
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0
```

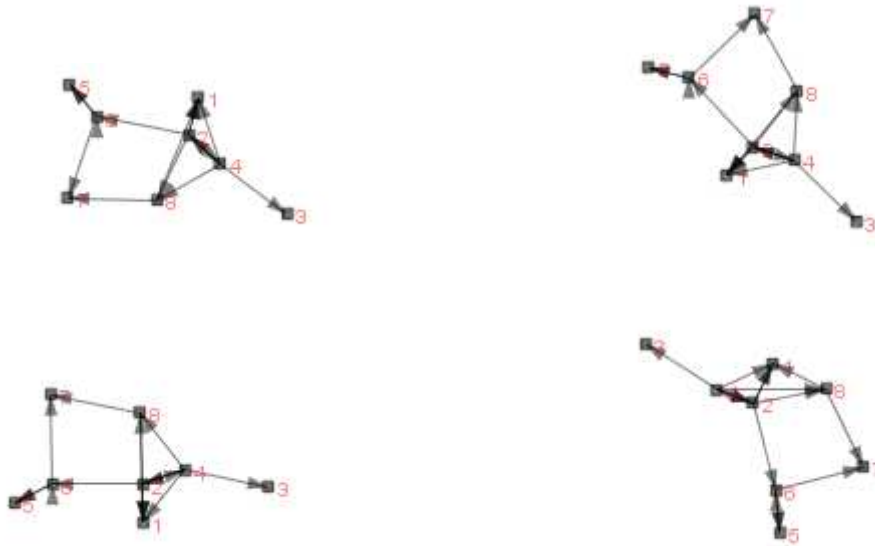
```
8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0
```

```
8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0
```

```
8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0
```

```
8 x 8 matrix
0 0 0 0 0 0 0 0
2 0 0 1 0 1 0 1
0 0 0 0 0 0 0 0
1 2 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0
```

2.2.4 Screenshots



3 Zusammenfassung

SoNIA ist ein auf Java basierendes Visualisierungstool für animierte Netzwerke. Es wurde an der Stanford-University entwickelt. Das empfohlene Inputformat ist .son. Es können auch Files im Format .net eingelesen werden, allerdings sind hier nur wenige Funktionalitäten umgesetzt. Nachdem die unterschiedlichen Betrachtungsweisen des Netzwerkes festgelegt wurden, können diese exportiert werden. Die Netzwerke können in Dateien (Matrix) ausgegeben werden, die Darstellung kann im Bildformat gespeichert und ein kleiner „Film“ erstellt werden.

4 Literaturverzeichnis

1. Sonia - Social Network Image Animator
<http://www.stanford.edu/group/sonia/>; Stand: 18. Juni 2008
2. SourceForge.net: sonia
<http://sonia.wiki.sourceforge.net/>; Stand: 18. Juni 2008