Assignment Report

by Daniel Nüst (daniel.nuest@uni-muenster.de)
for the seminar *R+OSGeo* held by E. Pebesma, T. Hengl and V. Olaya

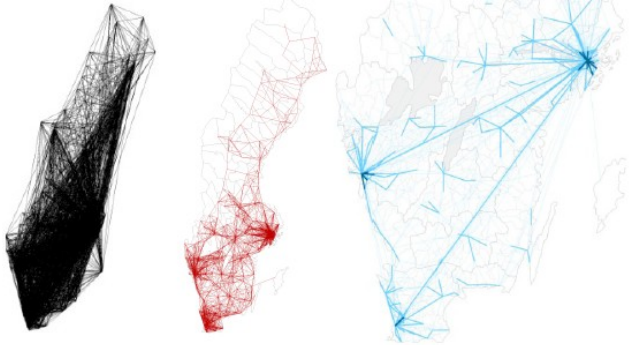
Summer semester 2009

Effective 07/10/09

1. Introduction

Flow maps are well-known within the cartographic community to represent movement of material or abstract things (migration of people, traded goods). The encoded information comprises strength, direction and course of the flow between locations. These locations are often a pair of coordinates. Albeit their powerful ability to remove visual clutter from these often very complicated patterns, there are no advanced automatic algorithms for flow maps found in professional GIS software. The most „beautiful“ examples remain hand-crafted, like the mother of all flow maps by Minard[1] (e.g. Minard's map of French wine exports for 1864[2]). The possible tools are:

- *Manual creation of straight lines between source and target of the flow in any capable GIS* – see examples below (own figures created with MapInfo). The leftmost map shows the clutter that a large dataset with a lot of different sources and targets, in this case commuters between Swedish communes, can create. The two maps to the right show filtered flows ("at least n-many people") and a visualisation of flow strength by colour and line thickness.



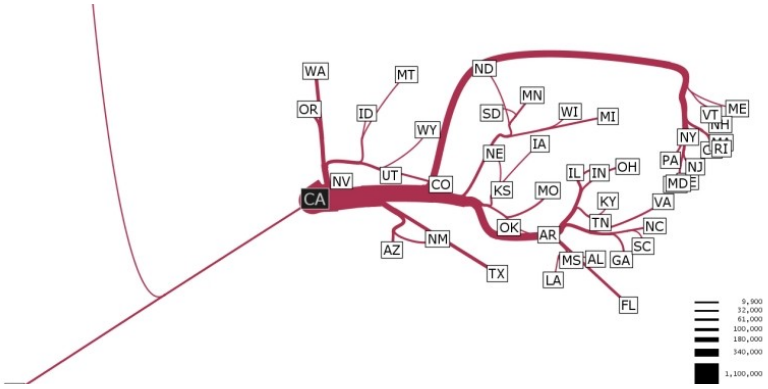
- *Tobler's Flow Mapper*[3] and *Flow Data Model Tools (FDMT) for ArcGIS*[4] can be used to

1 http://en.wikipedia.org/wiki/Charles_Joseph_Minard
2 http://en.wikipedia.org/wiki/File:Minard's_map_of_French_wine_exports_for_1864.jpg
3 http://www.csiss.org/clearinghouse/FlowMapper/
4 http://dynamicgeography.ou.edu/flow/

create lines or arrow shaped visualisations of flow. Waldo Tobler developed an interactive standalone software to create flow maps which was ported to VBA macros for ArcGIS by Alan Glennon. But both projects are currently not under active development and their products are still quite simple.

- *Flowmap* 7.3[5] is a standalone software that offers extensive analysis features for flow maps, for example using a road network as a basis for travel cost calculation.
- *Flow Maps – Automatic Generation and Visualization in GIS* by Birgit Pieke and Antonio Krüger [2] is a diploma thesis written at ifgi. It presents an algorithm for automatic flow map layout that reduces intersections of edges. The algorithm is implemented as an ArcGIS extension but could not be examined during this project.

These tools lack an intelligent routing for the lines in the sense of a good (understandable) visualisation as soon as many connections exist and/or are not available within open source software. Also a hierarchical structuring of the flow that can simplify the flow patterns is not utilized. They are also visually not really appealing. These are exactly the issues that are addressed by the tool *Flow Map Layout* (later referred to as FML) by Phan et al. [1]. Their technique uses „graph layout algorithms that minimize edge crossings and distort node positions“ to create a good flow map while maintaining relative positions. Example screenshot and figure title from the paper's website:



„Figure 1: A flow map of migration from California from 1995-2000, generated automatically by our system using edge routing but no layout adjustment.“

The freely available version of FML is limited to only one layer, meaning only one source (or sink) position for a group of target positions (or sources).

2. The Project

Because the code of Flow Map Layout is written in Java and available under a BSD license, which is compatible with GPL, I decided to implement an original contribution to the R+OSGeo community during my assignment. By using the algorithm implemented in Flow Map I create some first flow mapping features within the SEXTANTE[6] algorithm library.

The code is published publically on the project's website[7] as a zip file and in a public repository (see section Code and Libraries). I hope the licensing, documentation and availability of the project encourage others to engage themselves in developing flow tools for SEXTANTE, developing my algorithms further or trying out open source software in general.

A comment has to be made about working with gvSIG and SEXTANTE. A guide is given in

5 http://flowmap.geo.uu.nl/
6 www.sextantegis.com
7 http://ifgiweb.uni-muenster.de/~d_nues01/flowtools/

SEXTANTE Programming Guide [3] about setting up eclipse to work with SEXTANTE and gvSIG. Sadly, the guide was partly incomplete and quite a few (partly undocumented) errors had to be solved. The building process across many projects was not trivial to apply. At two points, extensions stopped working and time was lost in getting the components running again. The result is that close to 12 hours of working time were lost in work not including the actual project. The combination of gvSIG and SEXTANTE is a powerful tool and it's great to have all the code to see the inner workings – but it is very challenging as well.

2.1 Algorithms

First, a direct linear version is implemented (see first possible tool in the introduction). It simply creates lines between two points and attaches the given weight to it. These lines are stored in an ouput shape file. The weight can be used to visualize the strength of the flow along that line and thereby even filter out selected flows. This visualization is out of the scope of this project, but the attribute can be used for cartography features of any GIS or mapping application.

Second, the Flow Map Layout library is used to create a more sophisticated routing of the lines which is then transferred to bezier-style lines in a a shape file, also attributes with a weight. Naturally, the node adjustment feature of FML is deactivated for that to keep actual positions during the process. Because of the limitations of shape files regarding Bezier curves I implemented a sampling method that creates many short lines to imitate the curve using a `FlatteningPathIterator`[8]. The properties of the curve sampling can be set via two numeric variables: maximum sampling flattness (the maximum allowable distance between the control points and the flattened curve) and maximum point limit (the maximum number of recursive subdivisions allowed for any curved segment). The sampling itself can even be deactivated by unchecking the respective box in the algorithm menu.

2.2 Data Format

FML uses a simple text file as the input, which contains coordinates for one source (stored along with coordinates) and many destinations (stored with coordinates and value of the flow). This approach could be supported by providing tabular input to the given algorithms, but instead and a more GIS adapted one is taken: My SEXTANTE algorithm takes two shape files as inputs. The first file contains only the source point. The second file contains destination points with a flow size respectively a weight as an attribute.
The weight is assumed to be sensible already because it's processing lies out of the scope of this project. Preprocessing by the user, like standardization, might be required for a successful result. For simplicity the data is assumed to be in geographical coordinates (latitude and longitude) so that projected input points might require a conversion beforehand.
Multiple flows can be created using the layering option of any GIS utilizing SEXTANTE.

2.3 Problems

The goals of the project are not reached completely. The first, simple algorithms could easily be implemented. The second algorithm however is not successfully executed.

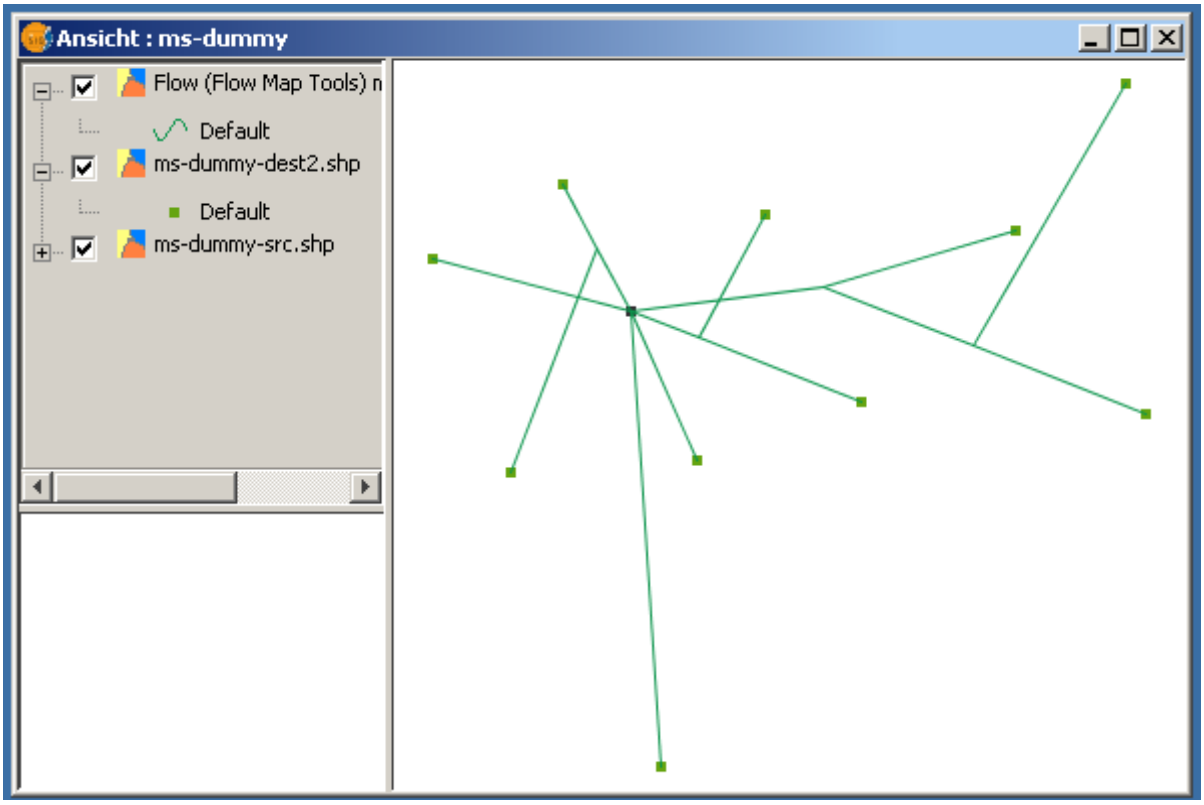
The problem lies in the calculation of the cubic curves. Errorenous values (i.e. NaN, not a number) are set for the control points of the Java class CubicCurve2D. These errors could be resolved with fewer destination points (see screenshot below). The root problem might even be a rounding error problem, because the used dummy points lie relatively close to each other.

But even if the curves are calculated, then the lines take long ways around the points. This again

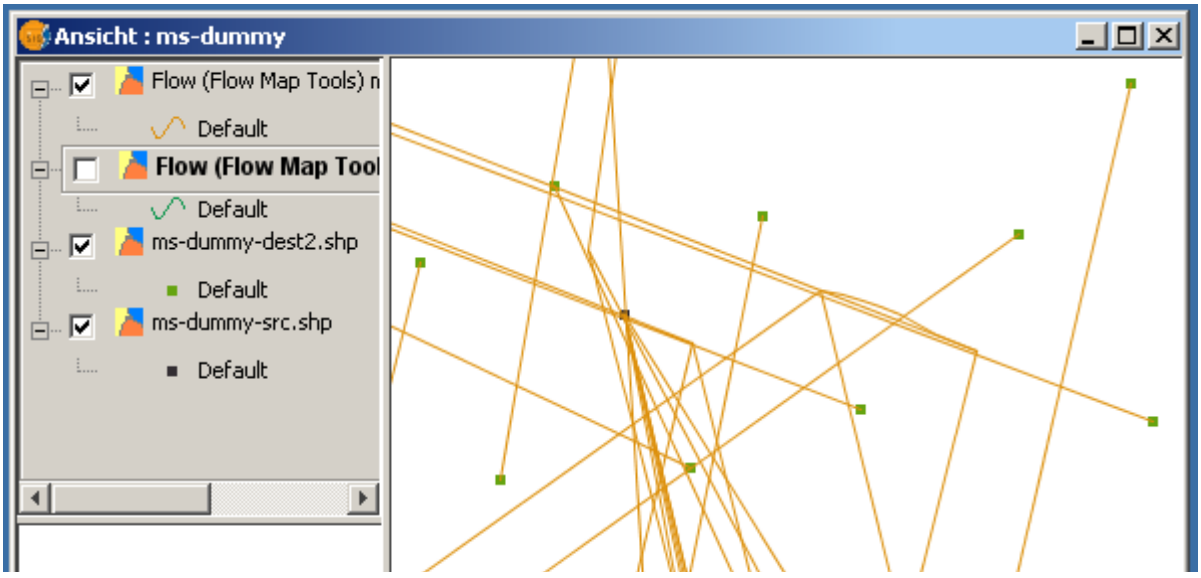
8 http://www.j2ee.me/javase/6/docs/api/java/awt/geom/FlatteningPathIterator.html

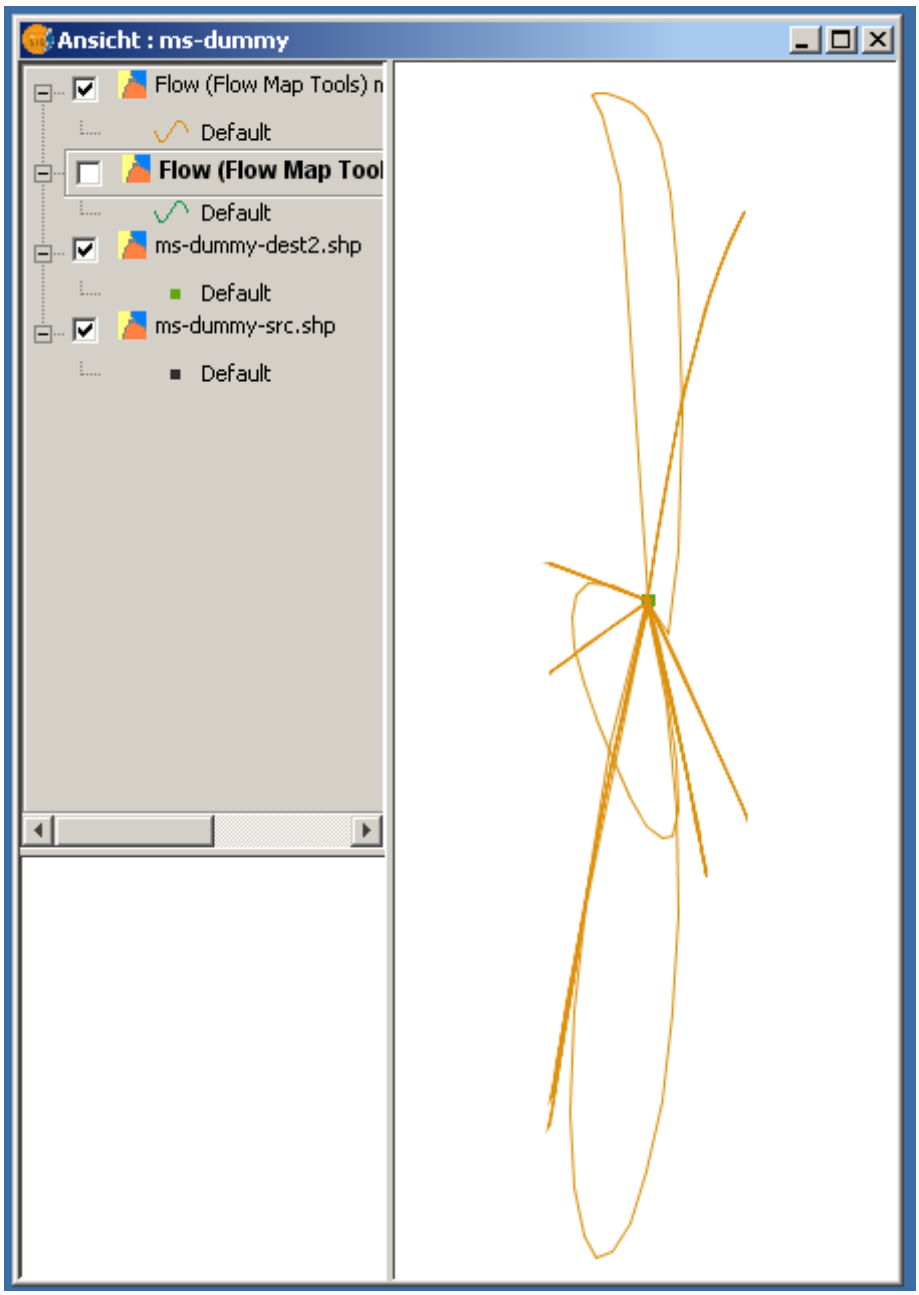
might be caused by the high proximity of the input points and requires deeper investigation of the layout algorithm itself.

Screenshot of flow map layout algorithm without sampling the curves:



Result with maximum sampling flatness 1.0E-5 and point limit 4, one close up screenshot and one showing the whole line layer:





2.4 Future Work

In this sections some targets for future work, other than fixing the problems mentioned before, are

- creating additional flow algorithms in flow tools, not limited to visualization of flows.
- accepting tabular data as input (e.g. in the original FML format).
- testing further with larger and with actual (!) datasets.
- adding translations.
- adding maven support (i.e. a pom.xml document).
- integrate processing (classification etc.) for the flow variable.

3. Code and Libraries

The libraries used by FML do not bring any problems as far as I can tell after some investigations. Due to that I decided to copy the source files of FML and the needed libraries directly into my project folder instead of using a .jar file. The libraries and packages are:

- epsgraphics.jar is licensed under GPLv2[9].
- j3dcore.jar, vecmath.jar are licensed under GPLv2 with the "classpath exception"[10].
- prefuse.jar[11] and prefusex.jar are licensed under BSD license[12].

The newly implemented algorithms can be found in the package `de.ifgi.sextante.flowTools`. Adaptation of classes from FML where minor changes were tested are in the subordinated package `flowmap`.

The code is available as an Eclipse[13] Java-project in a Subversion[14] repository under this address: `http://svn.xp-dev.com/svn/FlowTools/`.

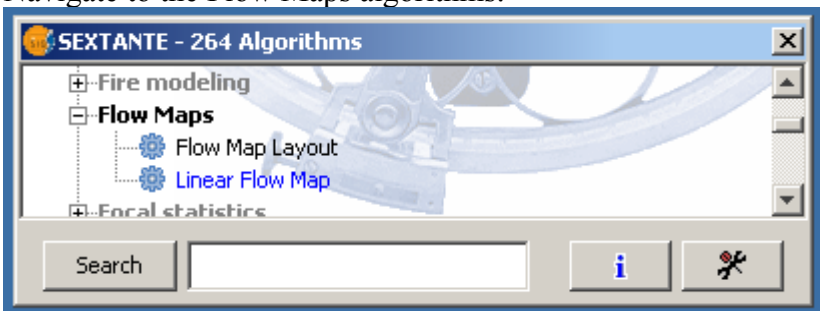
The installation procedure is as follows:

- Follow the instructions in [3] in the section *Setting up Eclipse to work with SEXTANTE and gvSIG*.
- Import the flowTools project folder to your workspace (checkout via Subversion or import zip file in Eclipse).
- Adjust the following properties in the ant build-file to your needs: `extensions-dir`, `gvSIG-lib-dir`.
- Run the build file, start gvSIG and try out the algorithms with the provided dummy data (see Example Workflow).

4. Example Workflow

This workflow is based on dummy data that can be found in the data directory in the project folder (`ms-dummy-dest2.shp` for destinations and `ms-dummy-src.shp` for source of the flow). Due to the encountered problems with the FML variant only the linear flow map algorithm is presented.

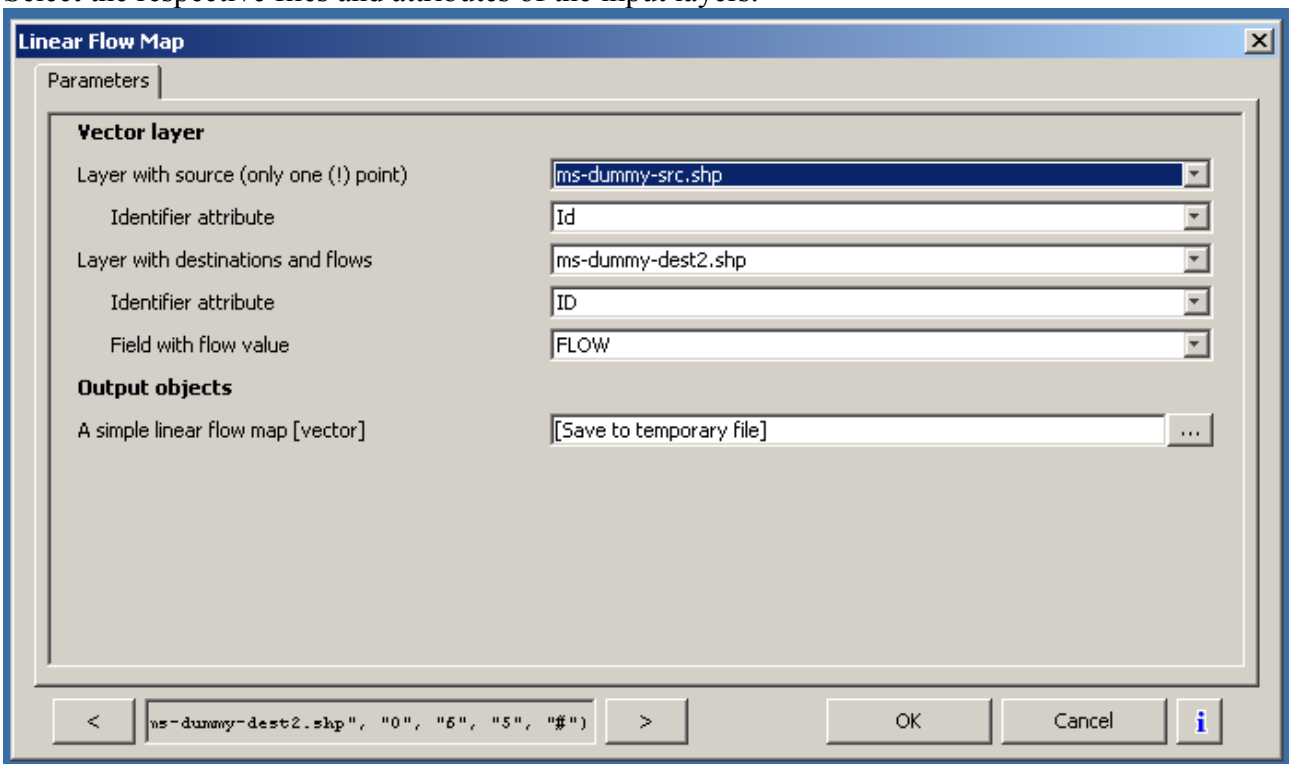
- Start gvSIG and add the shape files `ms-dummy-dest2.shp` and `ms-dummy-src.shp` as layers to your project.
- Navigate to the Flow Maps algorithms.



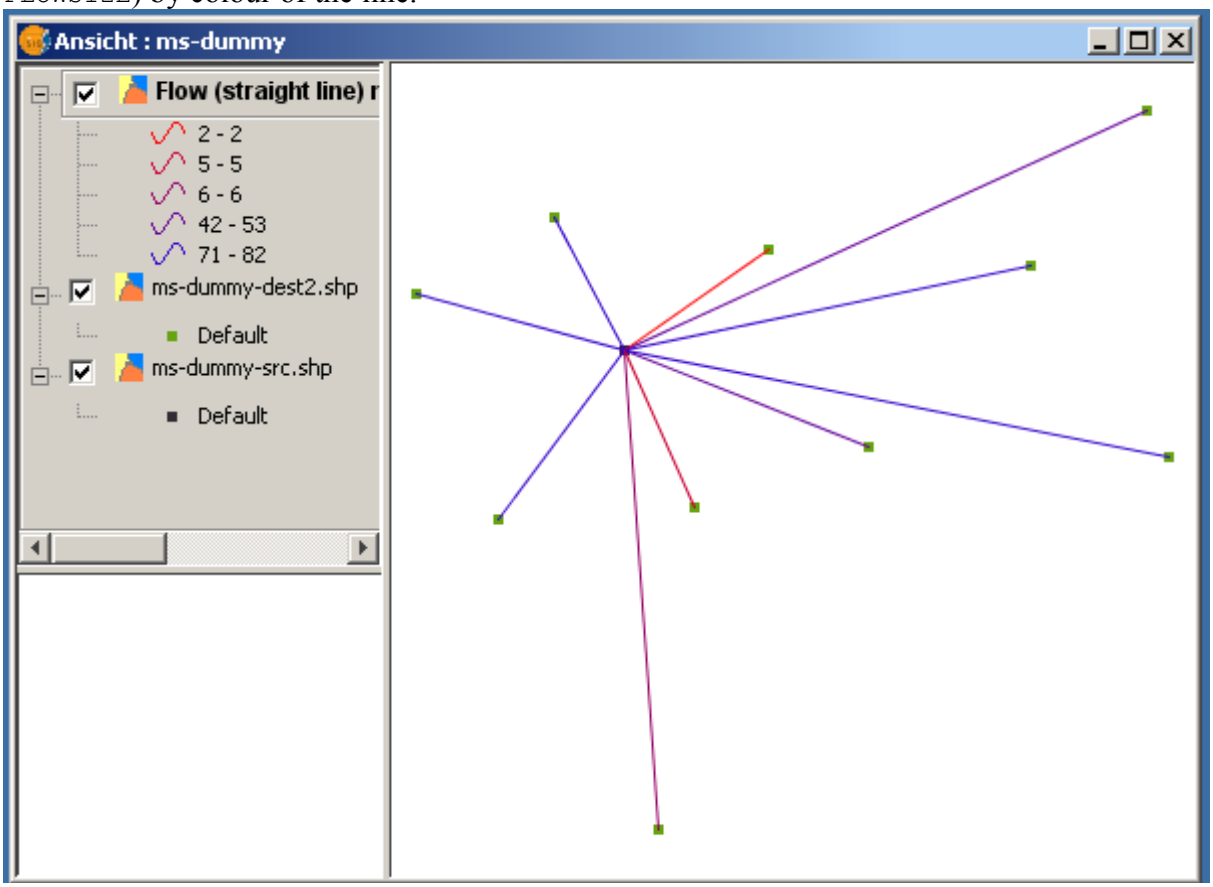
- Select "Linear Flow Map".

9 http://www.jibble.org/licenses/gnu-license.php
10 https://java3d.dev.java.net/#Licenses
11 http://prefuse.org
12 http://prefuse.org/license-prefuse.txt
13 http://www.eclipse.org
14 http://subversion.tigris.org

- Select the respective files and attributes of the input layers.



- Click “Ok”. The following screen shot shows a visualization of the flow strength (field `FLOWSIZE`) by colour of the line.



5. References and Literature

[1] Phan D, Xiao L, Yeh R, Hanrahan P,Winograd T. Flow map layout. *Proceedings of the 2005 IEEE Symposium on Information Visualization,* 2005 (Mineapolis, MN), IEEE Computer Society, Washingon, DC, USA, 2005; 219–224.
Homepage: http://graphics.stanford.edu/papers/flow_map_layout/

[2] Pieke B, Krüger A. Flow Maps – Automatic Generation and Visualization in GIS. *Proceedings of GI-Days 2007*, IfGIprints 30, Münster; 261-265.
Download: http://www.gi-tage.de/archive/2007/downloads/acceptedPapers/pieke.pdf

[3] Olaya V. *SEXTANTE Programming Guide*, Version 1.0 Rev. May 19, 2009
Download: https://svn.forge.osor.eu/svn/sextante/docs/LaTeX/en/ProgrammingSextante/ProgrammingGuide.pdf

6. License

15 http://creativecommons.org/licenses/by-nc-sa/3.0/