

Social Network Visualization as a Contact Tracing Tool

Magnus Boman^{*}
The Swedish Institute of
Computer Science (SICS) and
Dept of Computer and
Systems Sciences, Royal
Institute of Technology
Kista, Sweden
mab@sics.se

Asim Ghaffar
Dept of Computer and
Systems Sciences
The Royal Institute of
Technology
Kista, Sweden
asim.ghaffar@gmail.com

Fredrik Liljeros
Dept of Sociology
Stockholm University
Stockholm, Sweden
liljeros@sociology.su.se

ABSTRACT

Something many pathogens have in common is the requirement for tracing their spread under harsh time constraints, posing a so-called contact tracing (or “race-to-trace”) problem. We present a tool for visualizing contact networks, an important step towards practical use by epidemiologists, which generates interactive three-dimensional (3D) network visualizations. Its general purpose visualization engine can support multiple applications and varying pathogens. The main purpose is to trace, in the case of an outbreak, contacts among individuals known to have been at the same place.

Categories and Subject Descriptors

I.6.4 [Computing Methodologies]: Simulation and Modelling—*Model Validation and Analysis*; J.3 [Computer Applications]: Life and Medical Sciences—*Medical Information Systems*

General Terms

Management

Keywords

Epidemiology, social network, contact structure, MRSA, agent visualization.

1. INTRODUCTION

Contributing to pandemic disaster preparedness is currently one of the most important challenges to research worldwide. Epidemiologists use sophisticated methods for monitoring and controlling epidemic spread, including contact

tracing. For the modeling of individual contacts, a multi-agent system (MAS) may be employed, ideally mapping its population 1:1 with the real-world population of individuals under study. Within our interdisciplinary research group S-GEM, MAS modeling and simulation has been done for smallpox (cf. [5]), and is currently underway for influenza. Something many pathogens have in common is the requirement for tracing their spread under harsh time constraints. Our earlier work was centered around methodological aspects of MAS modeling for epidemiology applications, and on the computational efficiency of model execution, both necessary requirements for successful contact tracing efforts. Here, we focus on the problem of visualizing (and not on the modeling and simulation of) contact networks, an important step towards practical use by epidemiologists.

Several software tools for contact tracing are available to the epidemiologists constituting our user group, but these tools use rudimentary and flat structure display methods, such as lists and reports. We have developed a tool named *asimplot* that generates interactive three-dimensional (3D) network visualizations. Its general purpose visualization engine can support multiple applications and varying pathogens. The purpose of developing *asimplot* is to make it easier to trace, in the case of an outbreak, contacts among individuals known to have been at the same place. For reasons of brevity, we present here only one application, for a contagious infection called methicillin-resistant *Staphylococcus aureus* (MRSA). *Staphylococcus aureus* is a bacterial species commonly found on the skin. If it gains entrance through breaches in the skin it sometimes causes skin and soft tissue infections. Since carriage of *S. aureus* is common, it is also a common cause behind skin and soft tissue infections. MRSA is a variant of *S. aureus* carrying resistance towards penicillin antibiotics and all antibiotics chemically related to penicillin (so-called beta-lactam antibiotics). These resistant organisms have spread extensively in health-care environments throughout the world. There are two primary goals for the epidemiologist trying to reduce the spread of MRSA. First, to identify individuals who are spreading the disease, and secondly to identify locations (i.e. hospital ward or clinic) in which the disease is spreading. These goals are ideally instrumentalized into producing a tool that reduces the search-time for tracking agents (nodes) in the network. In this paper, we will focus on the first goal, since the two goals pose quite different requirements on methods for their

^{*}The authors are all members of the Stockholm Group for Epidemic Modeling (S-GEM), see s-gem.se.

realization, and for measuring the rate of success.

In health-care systems, registers and databases are maintained for administrative and economic purposes, which contain information on when individuals are in contact with the health-care system as an in- or outpatient, and for how long as an inpatient. This gives a unique opportunity to map when individuals have made contact by visiting the same outpatient clinic, or by being admitted to the same hospital ward simultaneously. This type of detailed data reflecting contacts between individuals and relevant to the transmission of infectious agents does generally not exist for other parts of the community. While *asimplot* is still subject to further development and refinement, it has already been taken into practical use by epidemiologists working with real MRSA data for the Stockholm region.

Epidemic spread within a health-care system is a major issue: not only does it defy the purpose of the health-care system; it also demands a costly effort for remedy when the spread crosses a certain threshold (cf. [9]). There are a few successful surveillance tools (cf. [3, 1]), for monitoring nosocomial, i.e. hospital-acquired, infections, but they are all built for single hospital environments. Thus, they fail to diagnose an infection as nosocomial if it originated from another hospital. In countries such as Sweden, which maintains countrywide patient data, it is possible to do surveillance at a higher level. Our ambition is at least citywide contact tracing. Scientific Visualization libraries such as VTK [10] have been used in certain information visualization endeavors, and the recent availability of information visualization libraries such as InfoVis [7] allow for visualization with more ease and flexibility. However, such general libraries have limited functionality and scope regarding network visualizations. For example, InfoVis is only 2D. Being a specialized field [4], network visualization have enjoyed mature and stable software packages, such as Pajek [2], for some time. Pajek is recommended for large network analyses, but was developed for producing static outputs. Pajek also has a relatively slow learning curve. Recent packages, such as *prefuse* [8], are geared towards smaller networks. The same is true for the *Sugarscape* family of platforms originating from work on *in silico* societal structures [6], and having very simple representation of agents and their environments.

2. DATA

Patient log data is available for the health-care system in Stockholm county. This health-care system consists of several hospitals and outpatient clinics. Each hospital consists of many clinical departments, and each clinical department can have several wards. The log is maintained at the ward level. A new log entry is made if a patient moves from one ward to another, within the same clinic or otherwise. The data is complete in the sense that within the system a log entry is always made when a patient moves between wards. However, it must be recognized that a patient can infect other patients outside the health-care system.

Our users evaluated the tool on real data. Due to the sensitivity of this data, real data could not be used for testing and developing *asimplot*. Therefore, several fake data sets were constructed using the statistical parameters calculated from the real data. We think that the usage of statistically fabricated data helped us avoiding any over-fitting to the real data set, but this claim can only be verified once the

tool is utilized with several real data sets. In MRSA, one real data set is not representative of any other real data set, since the regional variations are large.

In short, the user requirement specification contained the following constraints on *asimplot*.

1. Visualize contacts between agents as a network visualization.
2. Discriminate type and level of contact.
3. Discriminate strength of contact.
4. Make nodes (and edges) moveable and selectable.
5. Discriminate agent disease type, if so required.
6. Map axes dynamically on agent properties, and expose all data fields in the MRSA data set to this mapping, along with additional derived fields.
7. Give a multiple-level view based on the three types of MRSA used in the standard classification.
8. Provide an easy way of inspecting key properties of selected nodes and edges.
9. The tool should be executable with acceptable response times, on a standard (Pentium 4, 512Mb RAM) personal computer system.
10. There should be no unreasonably slow or jerky movements for medium-sized data sets of around 500 agents.

3. DESIGN AND IMPLEMENTATION

Through analysis of the requirements, a design decision was taken to develop a tool with two layers. General purpose 3D-network visualization functionality should reside in the base layer, and the MRSA-specific features should reside in the top layer. In the case of MRSA, *asimplot* can also be adapted to any hospital standard regarding the typology of the bacterium. The application will create and set properties of all nodes and edges and then pass them on to the engine, which will plot the graph. The engine is thus left with the minimal information needed for drawing each node, such as an agent's representation in 3D, and color information. We developed a prototype in seven iterations, each involving a meeting with users, who then proceeded to test the prototype. The tool thus has two logical parts: a general-purpose network-visualization engine (NVE) and an MRSA-specific application (MRSAApp). We developed NVE with DirectX-9, using *C/C++*. The problem domain is completely represented using *C++* classes, but no effort was made to encapsulate all parts of the tool behind classes; hence, a large part is written in *C*. Rendering is the most processing-intensive part in NVE. Though not yet fully optimized, the rendering function is efficient enough to smoothly display 2000 moving agents.

Users can move the camera along all three axes, with z-axis movement providing zoom in/out functionality. Alternatively, users can rotate the generated model along X, Y and Z-axes while keeping the camera fixed. Strafing is also available, which essentially means moving the camera perpendicular to its line of vision. The NVE can combine its knowledge about the agents that it is displaying with its ability to strafe in all three axes, to make the center of the visible

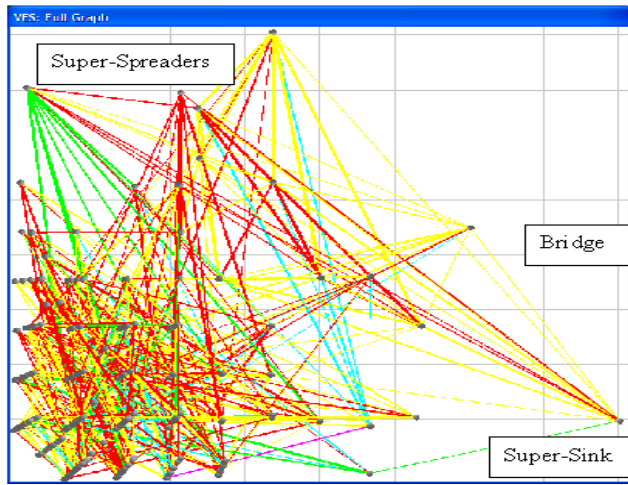


Figure 1: MRSApp plot for MRSA Dataset Stockholm 2004, with the x-axis mapped onto In Degree and the y-axis mapped onto Out Degree, at $t = 2000$ days.

agents the new center of the screen. Coloring the agents allows for the users to discriminate between the agents based on some property chosen at the application level. Edges can have different widths and colors. These two properties can be used to show various information associated with the edges, such as strength, category, etc. The NVE lets the user select any agent with a mouse-click. Selecting an object in 3D space is called picking and we have implemented this in the NVE. After loading an application, the NVE calls an application-specific normalization function which ensures that the final plot is visible. In the NVE output, an edge is the line between any two nodes and represents a contact in the MRSApp.

Figure 1 shows a plot in which the x-axis is mapped to In-degree and y-axis is mapped to Out-Degree. These two values are useful in contact tracing when it is important to guess who infected whom in a given relation. At the top of Figure 1, there are nodes representing individuals who have infected at least ten others. We call such individuals super-spreaders and this figure reveals some of their interesting properties. For example, the one with the maximum Out-degree has only four red links; this means that the individual was diagnosed after he or she possibly had infected most of his or her infected contacts. One other super-spreader only has red edges, indicating that the individual was already diagnosed when he or she infected others. There is also one super-sink: the node on the extreme right. Moreover, there are several bridge nodes, representing individuals that get infected within the health-care system, and then infect others there. One of these has been labeled, and that node served as a sink for a half-dozen nodes, and then as a possible source to another half-dozen.

4. CONCLUSIONS

We have presented asimplot, a network visualization tool for nosocomial epidemic spreads. At its base layer, the tool has a general-purpose network visualization engine. Multiple applications can be built on top, and as proof-of-concept we have developed an MRSA-specific contact-tracing ap-

plication. We have developed the application using user-centered development methodology, involving a small user group of active epidemiologists. An important goal was that the tool should reduce the time needed to track source nodes. This includes the time taken to generate the graph plus the time taken to comprehend a plot. Tests using real data indicate that asimplot actually does so. However, further tests and user evaluation is required. It is encouraging that asimplot seems useful in attacking the important race-to-trace problem: the task of improving the efficiency of control activities.

5. ACKNOWLEDGMENTS

The authors would like to thank the Swedish Emergency Management Agency for partial economic support. In addition, Liljeros acknowledges support from the European research NEST/PATHFINDER project DYSONET 012911.

6. ADDITIONAL AUTHORS

Additional authors: Mikael Stenhem (Department of Medical Epidemiology and Biostatistics, Karolinska Institutet, email: mikael.stenhem@smi.ki.se).

7. REFERENCES

- [1] A. Arantes, E. Carvalho, E. Medeiros, C. Farhat, and O. Mantese. Use of statistical process control charts in the epidemiological surveillance of nosocomial infections. *Rev. Saúde Pública*, 37(6):768–774, 1993.
- [2] V. Batagelj and A. Mrvar. Pajek—program for large network analysis. *Connections*, 21(2):47–57, 1998.
- [3] S. Bouam, E. Girou, C. Brun-Buisson, and E. Lepage. Development of a web-based clinical information system for surveillance of multiresistant organisms and nosocomial infections. In *Proc AMIA Symp*, pages 696–700, 1999.
- [4] U. Brandes, P. Kenis, J. Raab, V. Schneider, and D. Wagner. Explorations into the visualization of policy networks. *Theoretical Politics*, 11:75–106, 1999.
- [5] L. Brouwers. Micropox—a large-scale and spatially explicit micro-simulation model for smallpox planning. In *Proc Intl Conf Health Sciences Simulation*, pages 84–89, January 2005.
- [6] J. M. Epstein and R. L. Axtell. *Growing Artificial Societies—Social Science From the Bottom Up*. MIT Press, 1996. Brookings Institution.
- [7] J.-D. Fekete. The InfoVis toolkit. In *Proc IEEE Symp. Information Visualization*, pages 167–174. IEEE Press, 2004.
- [8] J. Heer, S. K. Card, and J. A. Landay. prefuse—a toolkit for interactive information visualization. In *Proc CHI*, pages 421–430. SIGCHI, 2005.
- [9] F. Liljeros, P. Holme, and J. Giesecke. The contact network of patients in a regional healthcare system. In *www.arxiv.org/abs/q-bio.OT/0505020*, May 2005.
- [10] W. J. Schroeder, K. M. Martin, and W. E. Lorensen. The design and implementation of an object-oriented toolkit for 3d graphics and visualization. In *Proc IEEE Visualization*, pages 93–100. IEEE Computer Society Press, 1996.