Creative Application to Remedy Epidemics

R. Kasprzyk, A. Najgebauer & D. Pierzchała
Military University of Technology, Cybernetics Faculty, Poland

Abstract

This paper focuses its attention on a project named CARE – Creative Application to Remedy Epidemics. CARE is a subsystem of monitoring, early warning and forecasting system SARNA, which was built at the Military University of Technology and was put into practice in the Government Safety Centre. CARE is a creative software solution that takes advantage of pioneering sociological theories, graph and network theory and the state-of-the-art in software technologies. Its very purpose, of particularly high importance nowadays, is to counter infectious diseases, such as a flu.

The paper deals with research of Complex Networks displaying the so called, *Scale Free* and *Small World* features, which make them accurate models of many networks – Social Networks in particular. These features on one hand appear to boost efficiency in communication networks and on the other speed up the spreading of many diseases.

Keywords: social networks, complex networks, epidemic modeling and simulation, vaccination strategies.

1 Introduction

Among many disasters on a worldwide scale there is one that concerns almost every country of the world. This is the influenza virus (flu) and the new strain, which is called *swine influenza* (A H1N1). Last season's flu has infected so many people that the problem of lack of hospital beds appeared. Due to the high risk of infection, according to legal acts, the Government Safety Centre (Polish acronym RCB) had to carry out tasks of daily monitoring of influenza and influenza-like illness in the country. Due to this reason and at the request of the RCB, at the Military University of Technology, a dedicated early warning and

doi:10.2495/RISK100451

decision support system named SARNA was built and put into practice. The SARNA is a multi-level system for reporting patient data and resources of hospitals for cases of influenza and influenza-like illness.

The most innovative module (at present in a testing phase) of the system is CARE. It utilizes state-of-the-art software technology and includes advanced forecasting models and pioneering social network theories. It has a very practical purpose of growing importance and demand: to counter infectious diseases, particularly any kind of flu. The system allows one to model almost any kind of disease based on epidemiological knowledge and to generate social networks using Complex Network theory. The very significant functional goals are: building special polls running on mobile devices to discover network topology, identifying "super-spreaders" and come up with a vaccination or isolation strategy, simulating and visualizing how the epidemic will spread in a given population and assessing the excepted outcomes of different epidemic scenarios. Interactive techniques for information visualization demystify data and reveal otherwise hidden patterns by leveraging human visual capabilities to make sense of abstract information.

CARE has enormous practical potential in regions where there are not enough medicines or time to treat those at risk. It could be widely applied not only for a flu virus but also for many others and can help to destroy those viruses. The proposed technology creates new possibilities for crisis management centers, epidemiology centers and governments to estimate and to management epidemic risks.

2 Definition and notation

Complex Networks are commonly modeled with either simple or directed graphs where the set of nodes (vertices) represents objects under investigation, and joins (arcs, edges) between two such nodes exist if the corresponding objects are related due to some interesting relationship. Formally, a graph is a vector G=<V,E,P> where: V is a set of vertices, E *is* a set of edges, and P is an incidence relationship, i.e. $P \subset V \times E \times V$. The degree k_i of a vertex v_i is the number of edges originating from or ending in vertex v_i. The shortest path d_{ij} from v_i to v_j is the shortest sequence of alternating vertices and edges, starting in vertex v_i and ending in vertex v_j. The length of a path is defined as the number of links in it. Networks very often are represented in practice by a matrix called the adjacency matrix **A**, which in the simplest case is an n x n symmetric matrix, where n is the number of vertices in the network, $n = |V|$. Element of adjacency matrix A_{ij}=1, if there is an edge between vertices i and j, and 0 otherwise.

In some cases the use of graph does not provide a complete description of the real-world systems under investigation. For instance, if network are represented as a simple graph, we only know whether systems are connected (data are exchanged between them), but we cannot model kind/strength of that connection. For now, however, we will use only formal graph definition.

3 Epidemic modeling in social networks

The network of possible contacts between individuals describes which individuals can infect which. We explore epidemic spreading in Social Networks modeled by Complex Networks. One of the most known mathematical models of Social Networks generators was a random graph [1]. Assuming equally probable and independent random connections between any two vertices in initially not connected graph, they derived a model with a highly unrealistic social network topology. Apparently, Complex Networks have *Scale Free* [3] and *Small World* [2] features, what make them accurate models of Social Networks. What are those features? A *Small World* network is a type of graph in which most nodes are not neighbors of one another, but most of them can be reached from any other with a small number of steps. The *Scale Free* feature pertains to a network in which most of people have relatively small amount of contacts, but there are some individuals that have huge amount of contacts. These individuals are called "*super-spreaders*", because they can spread diseases very fast. If such individual gets infected and in turn infects a portion (or perhaps all) of his numerous neighbors, that causes a sudden increase in the count of sick people. The application uses a few centrality measures [11] that help in finding the critical elements (e.g. degree centrality, radius centrality, closeness centrality or betweenness centrality) and finally suggests who should be immunized [14].

It is important to notice that we can determine the dynamic of epidemics if we know the network of possible contacts between people. Thus, the knowledge of such network topology makes it possible to simulate spreading of contagious diseases and to counteract them effectively. The system is a novel attempt in countering spreading of diseases, the first one to incorporate the knowledge stated above. Today there is no similar solution, and in most cases epidemiologists still choose people to vaccinate at random or decide to vaccinate the whole population if they have enough vaccines. We argue that these current methods are improper. Unfortunately, the most frequent situation is that we do not have enough vaccines to treat the whole population. And random immunization is almost useless, because it gives a very small chance of separating a social network into independent components – it is characteristic for *Scale Free* networks that they remain connected even after up to 80% of their nodes are removed (in our case: immunized or isolated) [10]. This suggests a simple solution: to immunize the "*super-spreaders*" first, which will slow or stop the spread. Unfortunately, this solution is very often impossible because in most cases the knowledge of network topology is uncertain and incomplete.

Our system proposes a number of solutions of gathering and using the knowledge of social networks features. First of all, it enables to create specific questionnaires (polls) which help in discover networks topologies. For example they may include a question that requires naming at least one acquaintance/ friend/ partner/ colleague etc. Individuals who are frequently named could potentially be "*super-spreaders*" [13]. It also gives a tool to generate synthetic social networks with the same statistical properties as real networks [12].

4 Social Network generators

In 1960, Erdös and Rényi [1] described their investigations of random graphs. Assuming equally probable and independent random connections made between any pair of vertices, they derived a model that suffered unrealistic topology. Because of that their model was not very useful for modeling real life social networks; nevertheless they proved a number of interesting results about random graphs. There are a few models of random graphs, some of them we have implemented.

Identifying and measuring properties of Social Networks is the first step towards understanding their topology, structure and dynamics. The next step is to develop a mathematical model, which typically takes a form of an algorithm for generating networks with the same statistical properties. Apparently, networks derived from real data (most often spontaneously growing) have "*six degrees of separation*", power law degree distributions, hubs occurring, tendency to form clusters and many other interesting features. Two very interesting models capture these feature, have been introduced recently.

First, Watts and Strogatz in 1998 [2] deal with mentioned features by a strategy that seems perfectly obvious once someone else has thought of it. They interpolate between two known models. They begin with a regular lattice, such as a ring, and then introduce randomness by 'rewiring' some of the edges. If all edges are rewired a random graph appears. The process of rewiring affects not only the average path's length but also the clustering coefficient. Both of them decrease as probability of rewiring increases. The striking features of this procedure is that for relatively wide range of rewiring probabilities the average path length is already low while clustering coefficient remains high. It is called *Small World* model, or more precisely: Beta-model of *Small World* network. Next to Beta-model there exists also Alfa-model of *Small World* network. That model tries to capture the actual way the social connections are formed. What is surprising is not that real Social Networks are *Small World* but that people are able to find the shortest path between each other so easily. Kleinberg [15] explained it using his own model of *Small World* networks.

Second, Barabási and Albert in 1999 [3] introduced their model of networks as a result of two main assumption: constant growth and preferential attachment. They expressed the degree sequence – the count of vertices with the same degree (number of adjacent edges) for all degree values found in the network. They show why the distribution of degrees' values is described by a power function. The process of network generation is quite simple. The network grows gradually, and when a new node is added, it creates links to the existing nodes with probability proportional to their current connectivity. This way, highly connected individuals receive more new links than not so connected ones, and also, 'old' nodes are more connected than 'young' ones. It is called *Scale Free* model. The process of *Scale Free* networks generation has many extensions and modifications [12].

5 Centrality measures

Turning to the analysis of network, we start by introducing centrality measures, which are the most fundament and frequency used measures of network structure. The central vertices in Complex Networks are of particular interest because they might play the role of organization hubs. Centrality measures address the question “Who (what) is the most important or central person (node) in given network?” No single measure of center is suited for all application.

We considered five most important centrality measures [12–15]: degree (gives the highest score of influence to the vertex with the largest number of direct neighbors), radius (chooses the vertex with the smallest value of shortest longest path to any other vertex), closeness (focuses on the idea of communications between different vertices and the vertex which is on average ‘closer’ to all vertices gets the highest score), betweenness (it can be defined as the percent of shortest paths connecting any two vertices that pass through the considered vertex) and eigenvector (acknowledges that not all connections are equal so connections to vertices which are themselves influential will grant a vertex more influence than connections to less important vertices). Normalization into the range [0, 1] is used here to make the centrality of different vertices comparable, and also independent of the size of the network.

5.1 Degree centrality

The simplest of centrality measures is degree centrality, also called simply degree. The degree centrality measure gives the highest score of influence to the vertex with the largest number of direct neighbors. This agrees with the intuitive way to estimate someone’s influence from the size of his immediate environment: $k_i = \sum_{j=1}^{n} A_{ij}$. The degree centrality is traditionally defined analogically to the degree of a vertex, normalized with the maximum number of neighbors that this vertex could have. Thus, in a network of n vertices, the degree centrality of vertex v_i, is defined as:

$$center_i^{Degree} = \frac{k_i}{n-1} \tag{1}$$

5.2 Radius centrality

If we need to find influential nodes in an area modeled by a network it is quite natural to use the radius centrality measures, which chooses the vertex using pessimist’s criterion. The vertex with the smallest value of shortest longest path is the most central node [12]. So if we need to find the most influential node for the most remote nodes it is quite natural and easy to use this measure. The radius centrality of vertex v_i, can be defined as:

$$center_i^{Radius} = \frac{1}{max_{j \in V} d_{ij}} \tag{2}$$

5.3 Closeness centrality

This notion of centrality focuses on the idea of communication between different vertices. The vertex which is ‘closer’ to all vertices gets the highest score. In effect, this measure indicates which one of two vertices needs fewer steps in order to communicate with some other vertex [13]. Because this measures is defined as ‘closeness’, the inverse of the mean distance from a vertex to all others is used.

$$center_i^{Closness} = \left[\frac{\sum_{j=1}^{n} d_{ij}}{n-1}\right]^{-1} = \frac{n-1}{\sum_{j=1}^{n} d_{ij}} \tag{3}$$

5.4 Betweenness (load) centrality

This measure assumes that the greater number of paths in which a vertex participates, the higher the importance of this vertex is for the network. Betweenness centrality refines the concept of communications, introduced in closeness centrality.

Informally, betweenness centrality of a vertex can be defined as the percent of shortest paths connecting any two vertices that pass through that vertex [14]. If $p_{lk}(i)$ is the set of all shortest paths between vertices v_l and v_k passing through vertex v_i and p_{lk} is the set of all shortest paths between vertices v_l and v_k then:

$$center_i^{Betweenness} = \frac{2\sum_{l<k}\frac{p_{lk}(i)}{p_{lk}}}{(n-2)(n-1)} \tag{4}$$

This definition of centrality explores the ability of a vertex to be ‘irreplaceable’ in the communication of two random vertices. It is of particular interest in the study of network immunization, because at any given time the removal of the vertex with the highest betweenness seems to cause maximum damage to the network in terms of its connectivity and mean distance.

Its main disadvantage is that the use of summation operator in practice requires global information about the network, in order to compute the betweenness of a single vertex, and that is simply not possible in many contexts. For the same reason it is expensive (time-wise) to compute the score of a vertex, although this disadvantage is recently of lesser significance.

5.5 Eigenvector centrality

Where degree centrality gives a simple count of the number of connection that a vertex has, eigenvector centrality acknowledges that not all connections are equal [15]. In general, connections to vertices which are themselves influential will grant a vertex more influence than connections to less important vertex. If we denote the centrality of vertex v_i by e_i, then we can allow for this effect by making e_i proportional to the average of the centralities of the v_i’s network neighbors.

$$e_i = \frac{1}{\lambda}\sum_{j=1}^{n} A_{ij} e_j \Rightarrow \vec{e} = \frac{1}{\lambda} A\vec{e} \Rightarrow A\vec{e} = \lambda\,\vec{e} \tag{5}$$

So we have $A\vec{e} - \lambda I\,\vec{e} = 0$ and the λ value we can calculate using $\det(A - \lambda I) = 0$. Hence, we see that $\vec{e}$ is an eigenvector and λ – an eigenvalue of the adjacency matrix. Assuming that we wish the centralities to be non-negative, it can be shown that λ must be the largest eigenvalue of the adjacency matrix and $\vec{e}$ the corresponding eigenvector.

5.6 Computational complexity

In this section we estimate computational complexity of centrality measures evaluation.

To compute degree centrality for the i-th node we must compute k_i in time $O(n)$, hence evaluation of the measure (1) for single node takes time $O(n)$ for weighted (WG) and unweighted graph (UG). For all n nodes it requires time $O(n^2)$.

To compute radius centrality (2) for the i-th node we must first compute shortest path tree in the i-th root to compute d_{ij} for all $j \in V$. For UG it takes time $O(n+m)$ using BFS algorithm and for WG – $O(n \log n + m)$ when we use Fibonacci's heap to implement priority queue in Dijkstra's algorithm. Next, we must find maximal value of d_{ij} – it takes time $O(n)$. Hence, evaluation of the measure for all n nodes in this case is $O(2n^2 + nm)$ for UG and $O(n^2 \log n + nm + n^2)$ for WG.

To compute closeness centrality (3) for the i-th node we use time $O(n^2 + nm)$ for UG and $O(n^2 \log n + nm)$ for WG (from the computation complexity point of view operation "max" is replaced by operation "sum", the rest is the same like in evaluation of radius centrality).

Fast algorithm for evaluation of betweenness centrality measure has been proposed by Brandes [16] (faster than in [17]) and for single node it requires time $O(m)$ for UG and $O(n \log n + m)$ for WG. Hence, evaluation of the measure for all n nodes is equal $O(nm)$ for UG and $O(n^2 \log n + nm)$ for WG.

Computation of eigenvector centrality for single node is estimated as $O(n^3)$, hence for n vertices this estimation is $O(n^4)$ (we can find better estimations).

6 Connection efficiency

To evaluate how well a network is connected before and after the removal of a set of nodes we use the global connection efficiency (GCE) [16]. We assume that the connection efficiency between vertex v_l and v_j is inversely proportional to the shortest distance:

$$connection_{ij}^{efficiency} = \frac{1}{d_{ij}} \tag{6}$$

When there is no path in the graph between vertex v_l and v_j we have d_{ij} = *infinity* and consequently connection efficiency is equal zero. The global connection efficiently is defined as the average of the connection efficiency over all pairs of nodes.

$$GCE = \frac{2}{n(n-1)} \sum_{i<j} \frac{1}{d_{ij}} \quad (7)$$

Unlike the average path length, the global connection efficiency is a well-defined quantity also in the case of non-connected graphs.

7 Vaccination strategies

As we mentioned earlier, random immunization of social networks is almost useless because *Scale Free* networks remain connected even after up to 80% of their all nodes removed (immunized or isolated). This would mean that under random vaccination, almost the whole population must be vaccinated to prevent the disease's spread. However, a clever attack (targeted vaccination) aimed at "*super-spreader*" will disintegrate the network rapidly. So, if we know social network topology we can use centrality measures to identify most important nodes and then vaccinate only those with the highest score to stop the disease.

It might be harder to come up with a clever strategy when we do not know the topology of social network. The question here is: how to identify and/or find the "*super-spreaders*" if we are not able to calculate values of centrality measures? We believe that it can be accomplished with a simple modification of random vaccination based on a new concept introduced in [13, 14] with few modification. According to our computer simulation results the new vaccination strategy it is much more effective, also in the case when our knowledge of the network topology is uncertain and incomplete.

The idea is to randomly choose, say, 20% of the individuals and ask them to fill out our special questionnaires. One of the most important question in all forms for any disease is to name at least one acquaintance/ friend/ partner/ colleague etc., and then vaccinate those identified individuals (vaccinate the neighbors). Potential "*super-spreaders*" have such a large number of contacts that they are very likely to be named at least once. On the other hand, the "*super-spreaders*" are so few in number that the random sample of individuals is unlikely to include many of them. Using this vaccination strategy, a disease can be stopped by vaccinating less than 20% of individuals. If a larger sample is polled, or those named twice are vaccinated the total number of vaccinations required can be even lower. This basic method can be modified in many ways and be adapted to a specific disease or virus.

Figure 1 shows the percentages of population that must be vaccinated using three vaccination strategies to reduce GCE by a factor of ten (we assumed that it would prevent the disease's further spread).

8 System overview

Our application was implemented using *Microsoft* .NET 3.5 platform with SP1 [18]. Originally, the simulator was developed as a desktop application however our experiences with Web applications motivated a complete modification of architecture. Now we use SOA (*Services Oriented Architecture*) with *Web Services*. The web user interface uses AJAX (*Asynchronous JavaScript and* XML) technology takes advantage of *Microsoft Silverlight* 2.0 solution [18].

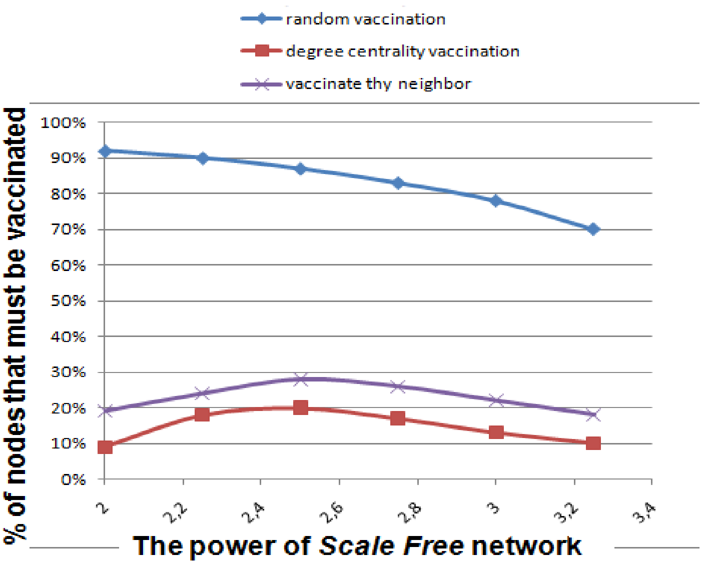


Figure 1: The effect of three vaccination strategies for different *Scale Free* networks (by the power of their node degree distribution).

The information is stored in MS SQL Server database which is divided into three parts: questionnaires and validated data from questionnaires; models of diseases and networks; and reports with simulation results.

The system consists of a server and different types of clients. The server side contains a few components such as the database, web services, and the simulation engine. At the moment there are two types of clients. First, a web client accessible through web browser, which contains diseases' models editor, social network generators algorithms, interactive simulation with visualization subsystem and geo-contextual data relation based on MS Virtual Earth. Second, there is a mobile platform client that carries the poll application. Other developers could also access web services and design their completely new interface and/or create custom analyses.

8.1 CARE functionality

The application allows users:

a) Model any kind of disease based on epidemiological knowledge;

b) Model and generate social networks using Complex Network theory;

c) Build special polls running on mobile devices and designed discover network topology;

d) Identify "*super-spreaders*" and come up with the most efficient vaccination strategy;

e) Simulate and visualize how the epidemic will spread in a given population;

f) And assess the excepted outcomes of different vaccination strategies.

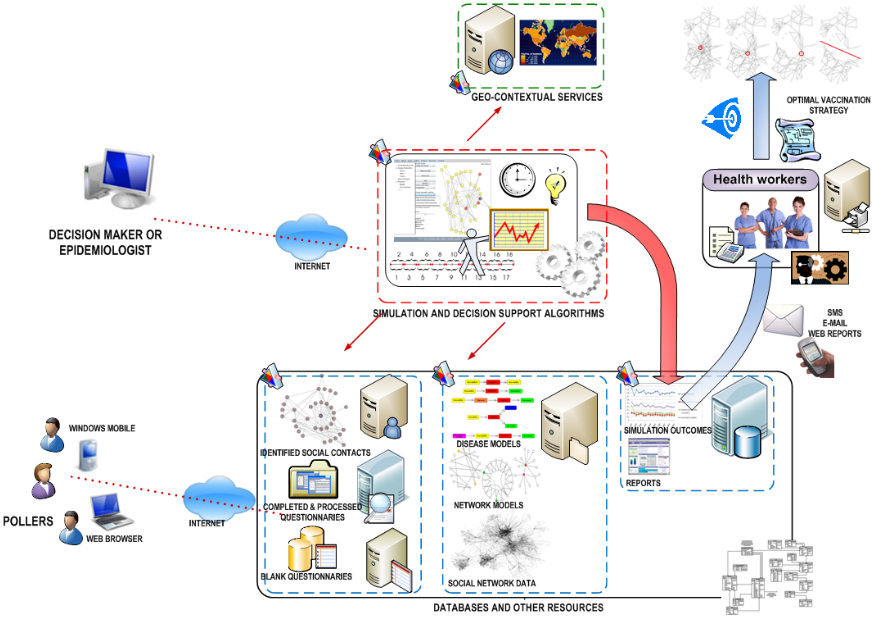


Figure 2: CARE architecture overview.

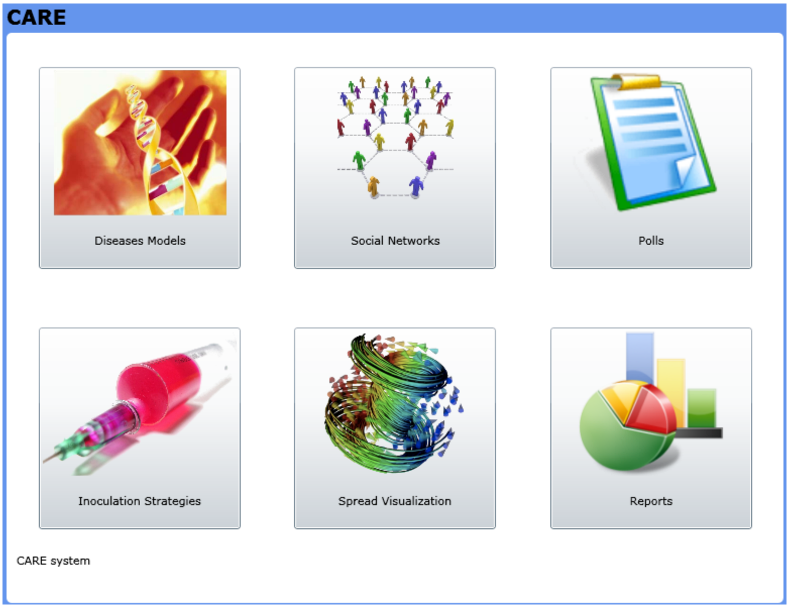
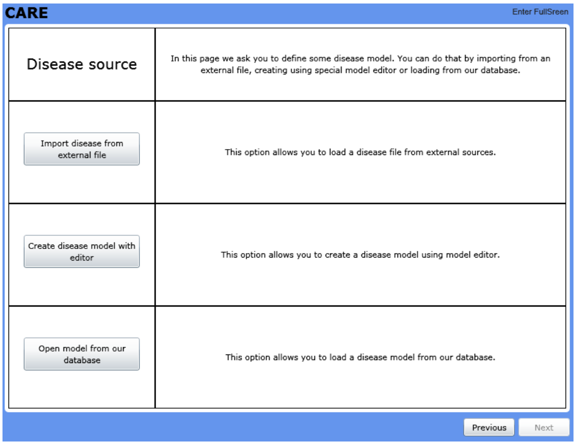


Figure 3: CARE main screen.



CARE

Enter FullSreen

Disease source	In this page we ask you to define some disease model. You can do that by importing from an external file, creating using special model editor or loading from our database.
Import disease from external file	This option allows you to load a disease file from external sources.
Create disease model with editor	This option allows you to create a disease model using model editor.
Open model from our database	This option allows you to load a disease model from our database.

Figure 4: The page with a disease source choice.

8.1.1 Disease source

The next step is to choose a disease model. The user has three possibilities: he can import it from external sources, create using the model editor or open a model from the database. Options no. 1 and no. 3 mean simple load *graph*ML file but the second option is a very interesting feature of the system.

It is a good moment to explain how the system defines a disease model. Generally it uses a State Machine approach. It means that every disease consists of a few states (e.g. susceptible, infected, carrier, immunized, dead etc.) which can be assigned to each individual and the system allows state to change as a result of social interactions (contacts). So underling network topology is crucial problem in our simulator. For simulation we need at least two states S (*Susceptible*) and I (*Infected*). For realistic scenarios we make possible to define the model of disease with many more states and transitions between them. In the classical theory of infectious diseases, the most extensively studies epidemic models are the SIR (*Susceptible-Infected-Removed*) and SIS (*Susceptible-Infected-Susceptible*) model.

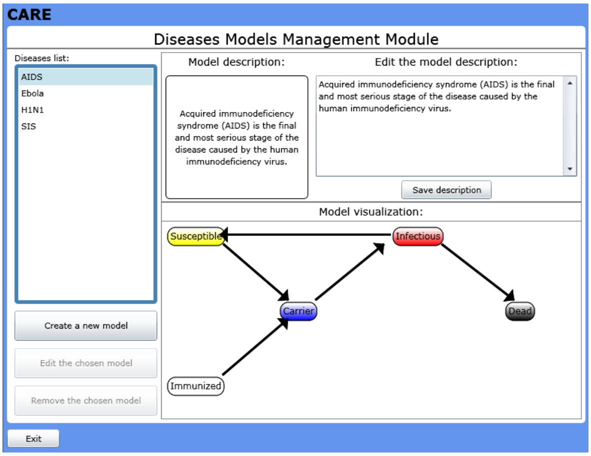
We allow one to build the models of diseases with any state in the editor we have proposed. There is also some transition probability, minimal/maximal time that individual spend in state and maximal number of neighbor that can be infected in simulation time step.

Figure 5 presents AIDS model created using the model editor.

As we can see, the model editor also allows the user to export the created model to the database for a later use.

8.1.2 Network source

Now let’s have a look at the Silverlight client. The first thing a potential user is asked to do is to load or generate a network. He can do this by importing an appropriate file or by generating new network using our predefined generators.



Acquired immunodeficiency syndrome (AIDS) is the final and most serious stage of the disease caused by the human immunodeficiency virus.

Acquired immunodeficiency syndrome (AIDS) is the final and most serious stage of the disease caused by the human immunodeficiency virus.

Figure 5: AIDS model created with the model editor.

The system uses *graph*ML network format to keep networks in the database and external sources (e.g. epidemiologists' sources). It is a very popular xml format that is used by many applications working with networks.

Another interesting option is to use generators to generate an artificial network but with the same statistical properties as real social networks.

The algorithms generally make networks that are random graphs, *Small World* networks, *Scale Free* networks or their modifications. They all were defined in studies of authors like Watts and Strogatz (who worked on the *Small World* model), Kleinberg (who explained using his own model why people are able to find the shortest paths between each other in *Small World* model very easily) or Barabasi and Albert (who introduced their model of networks as a result of two main assumptions: constant growth and preferential attachment, they worked on *Scale Free* model).

8.1.3 Poll application

The first step in fighting against a disease is to get information about social network subject to that disease. The CARE software helps sharing the special poll application dedicated for mobile platforms.

The application could be used for instance by missionaries in Africa to get information about people. Designed with the purpose of decreasing costs of operation, the poll application could be launched on PDA or simple mobile phones. Since in some remote places (say, in Africa) internet access could be a problem, the application stores data in a local database (deployed together with the application) and when the access is possible, the local data is synchronized with the main database stored on the server.

Figure 6 presents how the poll application looks like at the moment.

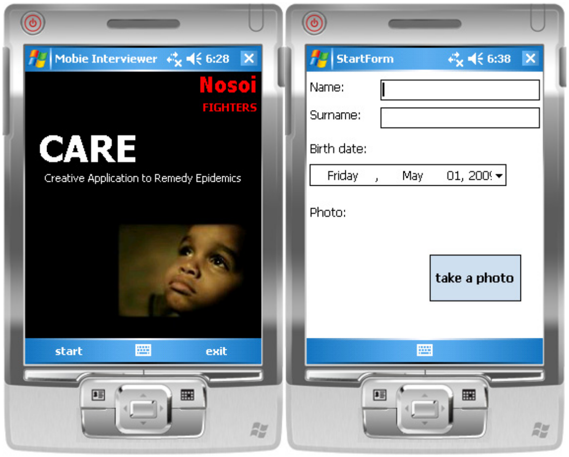
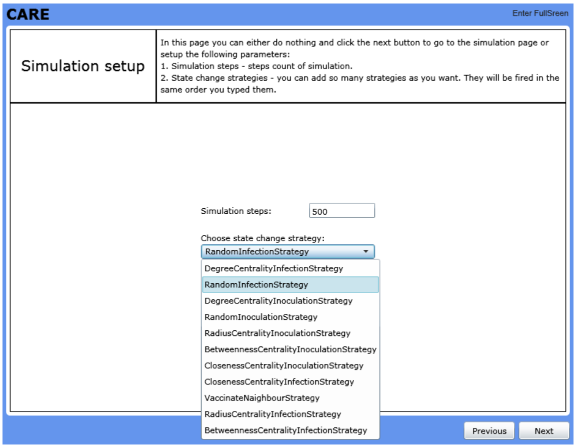


Figure 6: The poll application user interface.



In this page you can either do nothing and click the next button to go to the simulation page or setup the following parameters:
1. Simulation steps - steps count of simulation.
2. State change strategies - you can add so many strategies as you want. They will be fired in the same order you typed them.

Simulation steps: 500

Choose state change strategy: RandomInfectionStrategy

DegreeCentralityInfectionStrategy
RandomInfectionStrategy
DegreeCentralityInoculationStrategy
RandomInoculationStrategy
RadiusCentralityInoculationStrategy
BetweennessCentralityInoculationStrategy
ClosenessCentralityInoculationStrategy
ClosenessCentralityInfectionStrategy
VaccinateNaighbourStrategy
RadiusCentralityInfectionStrategy
BetweennessCentralityInfectionStrategy
Previous
Next

Figure 7: In this page the user sets up the simulation.

8.1.4 Simulation

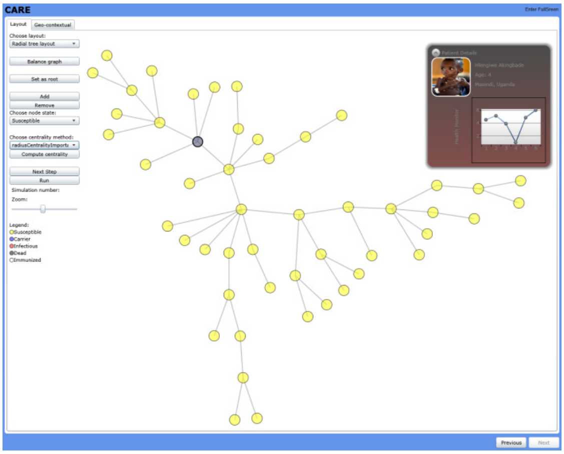
The next page is the simulation setup. In here the user can either accept the defaults or modify several parameters: number of simulation steps and initial conditions for the simulation – which include the number and pattern (randomly, by a chosen centrality measure) of infected individuals and implemented vaccination strategy (random, vaccinate thy neighbor, by chosen centrality measure).

From the user's point of view this page is perhaps the most interesting step in the web client. It is an evidence for statement that interactive information visualization is important and how visual representation of information can be used to demystify data and reveal otherwise hidden patterns by leveraging human visual capabilities to make sense of abstract information [8].

Generally the system proposes two ways of information visualization. The first way is called "layout" and is presented below.

Here the user can manipulate the network, in particular he can:

- change a position of each node;
- choose a layout (which means a choice of some graph balance algorithm);
- set some node as the root of layout (it's a feature suitable for tree based layouts like radial tree layout shown on the picture above);
- remove nodes (equivalent to e.g. isolating some individual);
- zoom in and out for better view;
- start the simulation (the most important option here).



Choose layout: Radial tree layout

Choose node state: Susceptible

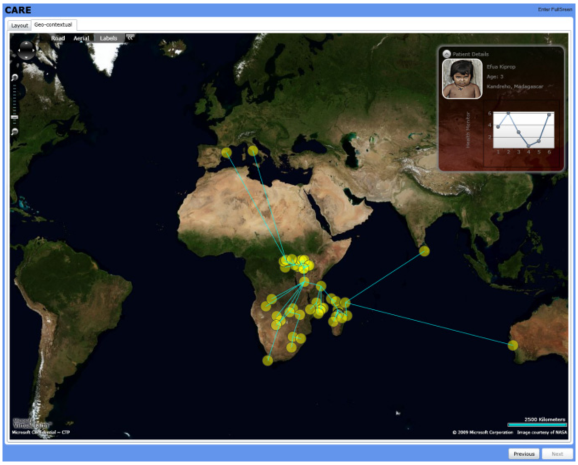
Choose centrality method: radiusCentralityImport

Simulation number:

Zoom:

Hlengiwe Akingbade
Age: 4
Masindi, Uganda

Figure 8: The "layout" view of the simulation page.



Efua Kiprop
Age: 3
Kandreho, Madagascar

Figure 9: The "geo-contextual" view of the simulation page.

In the upper right corner of the picture we can see the patient’s details panel, which presents information about the selected node (individual).

An alternative way to visualize information, called “geo-contextual”, makes it easier to become well oriented in locations of each individual. It could be very useful for epidemiologists which now can see all the nodes presented on the map and use that to make better decisions.

8.1.5 Reports

The last step is to analyze a report chart, which is created on the basis of the simulation. The x-axis represents simulation steps and the y-axis represents the number of individuals in each state in the current step.

The user can estimate the results and the effectiveness of the chosen strategy. In the future there will be also a possibility of getting the list of individuals that the system suggests to vaccinate.

9 Summary

Based on the defined centrality measures, we show how to discover the critical elements of any network. The identification and then vaccination of such critical individuals in a given network should be the first concern in order to reduce the consequence of an epidemic.

We believe that the CARE software has enormous practical potential in such regions as Africa, where there are not enough medicines to fight against dangerous diseases. It can be used in many various matters: automatic notification of hospitals, rescue services, etc. about emergency situation e.g. risk of epidemic (with providing coordinates and patient information). It could be also used by crisis management centers, epidemiology centers in the whole world to fight against not only terminal infectious diseases, but also any kind of flu etc.

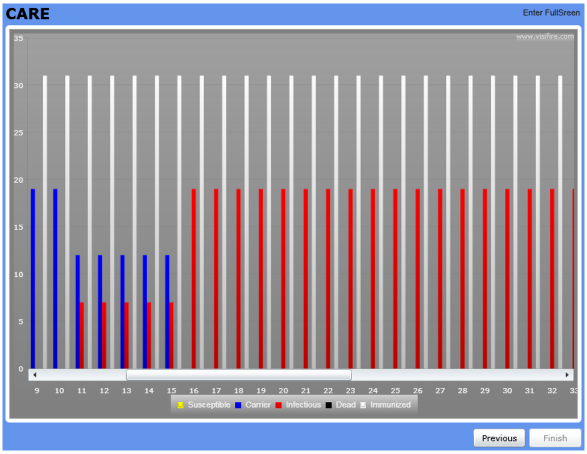


Figure 10: The page with a report chart.

Acknowledgements

This work was partially supported by the research projects: PBZ-MNiSW-DBO-01/I/2007 and GD-WAT-WCY/2010 titled "Service-oriented information exchange bus for simulation systems supporting decision in conflict and crisis situation".

References

[1] Erdös, P. & Rényi, A.: *On random graphs*, Publ. Math Debrecen 6 (1959), 290-297.

[2] Watts, D.J. & Strogatz, S.H.: *Collective dynamics of "small-world" networks*, Nature, 393:440-442, 1998.

[3] Barabási, A.L. & Albert R.: *Emergency of Scaling in Random Networks*, Science, 286:509-512, 1999.

[4] Kasprzyk, R.: *Complex networks in countering terrorism*, VII International Workshop for Candidates for a Doctor's Degree OWD, Wisła 22-25.10.2005.

[5] Kasprzyk, R.: *Fault and attack resistance of Complex Networks*, Military CIS Conference 2008, Kraków 22-24.09.2008.

[6] Kasprzyk, R.: *Modelling the evolution of Complex Networks with Small World and Scale Free features*, X International Workshop for Candidates for a Doctor's Degree OWD, Wisła 18-21.10.2008.

[7] Kasprzyk, R.: *Fault and attack resistance of Complex Networks*, X International Workshop for Candidates for a Doctor's Degree OWD, Wisła, 18-21.10.2008.

[8] Brandes, U. & Kenis, P. & Raab, J.: *Explanation Through Network Visualization*, Methodology 2006, Vol. 2(1): 16-23.

[9] Pastor-Satorras, R. & Vespignani, A.: *Epidemic Spreading in Scale-Free Networks*, PRL Volume 86, Number 14 p. 3200 (2 April 2001).

[10] Crucitti, P. & Latora, V. & Marchiori, M. & Rapisarda, A.: *Error and attack tolerance of complex networks*, Physica A 340 (2004), 388-394.

[11] Wuchty, S. & Stadler P.F.: *Centers of complex networks*, Journal of Theoretical Biology 222 (2003), 45-53.

[12] Barabási, A.L. & Albert R.: *Topology of Evolving Networks: Local Events and Universality*, PRL Volume 85, Number 24 p.5234 (11 December 2000).

[13] Cohen, R. & Havlin, S. & ben-Avraham D.: *Efficient Immunization Strategies for Computer Networks and Population*, PRL Volume 91 Number 24 p.247901-1 (12 December 2003).

[14] Madar, N. & Kalisky, T. & Cohen, R. & ben-Avraham, D. & Havlin, S.: *Immunization and epidemic dynamics in complex networks*, Eur. Phys. J. B 38, 269-276 (2004).

[15] Kleinberg, J.M: *Navigation in small world*, Nature 406, 845 (24 August 2000)

[16] Brandes, U.: *A Faster Algorithm for Betweenness Centrality*, Journal of Mathematical Sociology, 25 (2001), 163-177.

[17] Freeman, L.: *A set of Measures of Centrality Based on Betweenness*, Sociometry, vol.40, pp. 35-41, 1977.

[18] http://www.microsoft.com/NET/

[19] http://silverlight.net/