



Are matrix-based or node-link graphs more readable when representing causal relationships for social and health data?

Kristina Lazarova

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March, 2017

Abstract

We show how to produce a level 4 project report using latex and pdflatex using the style file l4proj.cls

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

1	Literature Review	1
1.1	Data Visualisation	1
1.2	Causal Relationships	2
1.3	Node-Link Graphs	3
1.4	Matrix-based Graphs	4
2	Introduction	5
3	Implementation details	7
3.1	Software tools and technologies	7
3.2	Development Process	8
3.2.1	User Stories	8
3.2.2	Wire-frames	8
3.2.3	System Design	9
3.2.4	Implementation	10
3.3	Challenges	11
3.4	Software reliability testing	11
4	Evaluation	12
4.1	Design	12
4.2	Stimuli	12
4.3	Pilot	13
4.4	Participants	14
4.5	Procedure	15

4.6	Results	15
4.7	Discussion	15
Appendices		16
A	Database Schema	17
B	Running the Programs	18
C	Generating Random Graphs	19

Chapter 1

Literature Review

1.1 Data Visualisation

Data visualisation has been described as a technique that makes use of computer-supported, interactive illustrations to deepen human's understanding of a dataset [4]. Information visualisation is necessary when dealing with increasingly large and complex data.

Information visualisation systems are most helpful when a set of data is being explored [10]. Usually this occurs not when someone is looking for a specific answer to a question, but when a deeper understanding of that data set is sought. It was suggested that the value of information visualisation is not in understanding a specific question, it is about developing and deepening one's insights of a set of data [10]. Data visualisation is able to facilitate this process because there are a number of cognitive benefits associated with it. A large visual cue that illustrates data becomes a single point of reference for human cognitive processes. Visuals become external cognition helpers in facilitating human memory by providing a bigger working set for analysing data. Furthermore, visual architecture and design applied at company and department levels have been reported to be successful due to the low cognitive burden for visualization reading [22].

//examples, why is visualisation good and areas in which it helps

Data visualisation has been used in a number of fields. For example, it can be successfully used in the Bank of America Chicago Marathon, which is one of the biggest marathons in the world [18]. Large amounts of live data are gathered to establish some of the main principles for event management information visualization. Some of the main benefits from these practices are keeping track of the progress of the participants, communicating information between the agencies organizing the marathon efficiently, and improve medical preparedness. Moreover, astronomical researchers have been suggested to benefit from data visualization techniques [13]. One such method is linked views in which a user is able to select, highlight and include or exclude points from display and analysis from various data visualisation dimensions. For example, a valuable visualisation for astrologists can be interactive exploration of relationships between data points in statistical graphs and locations in live 3D space. Another instance of beneficial data visualisation is "Geographic Information System" tools which are used in demographics and geography. For example, Engage3D provides the functionality of exploring layers of maps in a linked-view systems.

Inventions in this field have been continuously made in order to improve the representations. For example, a heatmap represents a graphical illustration that evaluates conical distribution values around data points based on a respective data value that has associated respective data point [5]. Various implementations of a heatmap were developed and discussed in order to enable easy future work with heatmaps suited for a particularly needed implementation.

There have been data visualisation issues connected with Big Data [14]. First, visual noise is the phenomenon of having so much data points on the screen that the user cannot see them as separate points which leads to visibility loss. Second, large images are dependent on the device resolution abilities. However, even if add together many devices for partial data visualisation to represent a more detailed illustration or a larger amount of data, human perception will eventually reach its limit. Therefore, after a certain point people will not be able to understand the representations or their analysis. A third problem is Information Loss which is caused by the previous issues because of data filtration and aggregation. These techniques are likely to hide information and mislead to people analysing the data. Another problem is High Performance Requirements especially in dynamic visualization. In addition, the visualisation might be accompanied by high rate of image change and lead to people being unable to react to the number of data changes.

A Big Data visualization method is suggested to be TreeMap [14] which is a hierarchical way of representing data by rectangles. Some advantages of this method are that the hierarchical matching clearly shows data relations and extreme outliers are visible. The disadvantages related to this method are that the data must be hierarchical which makes it unsuitable for examining time patterns, for example.

//readability

A well known issue with data visualisation is that sometimes it is challenging for people to understand it. A study aimed to examine familiarity of museum visitors with different visualisation techniques [3]. They included charts, maps, graphs, and networks to reveal how familiar people are with them. It was found that even though most participants were interested in science and art they have difficulties naming and reading the visualizations. It was concluded that people are interested in visualisation techniques, but have significant difficulties in naming and understanding them.

In order to solve this issue, another research area has focused on understanding how information visualization novices think and the approaches that can facilitate their learning. A study used sales data to find the barriers for novices when reading iterative visualisation construction and the way they think about visualisation specifications [15]. They found that the biggest barriers were interpreting questions into data factors, visual mappings, and understanding the visual representations. It was found that there is a need for instruments that suggest possible visualizations, facilitate help with learning, and are integrated with tool support for the whole analytic process. Furthermore, recent research acknowledged that individual differences between people will have influence on the way they interpret graphical representations [28]. They suggested that visualization performance can be improved by personalising visuals according to one's needs, abilities and preferences.

Very little is known about the way users understand and read data visualisations and how they interact with different layouts [9]. It is interesting to explore if there are layouts which provoke better task performance and better response time. Etemadpour and colleagues [9] conducted a research to find out more about performance in similarity based layouts that are generated by multidimensional projections. The results suggested that projection performance is task-dependent and depends on the nature of the data. Therefore, they concluded that the same data layout can have different performance on different tasks and that performance will also be influenced by the characteristics of the data.

1.2 Causal Relationships

Causal relationships are of great interest for scientist who examine influence of different factors on each other. Particularly, research regarding health, social and behavioural sciences aim to investigate questions about causal rather than associative relationships. With the help of statistical analysis associations among different factors can be inferred. Associations are relationships that can be observed in joint distributions of factors such as regressions and correlations. Causal analysis, in addition, is the practice that aims to infer probabilities under factors that are

changing [26]. These could be, for example, changes influenced by factors such as drinking, childhood issues or applied treatments. Causal relationships cannot be defined from the distribution alone - for example, effect, confounding, disturbance.

A study looking to identify factors influencing blog design used the Decision Making Trial and Evaluation Laboratory method (DALMATEL) which is used to illustrate the relationships between factors and allows causal relationships to be shown [19]. Some of the causal relationships they found were that color arrangement directly impacts simplicity of layout, colour arrangement directly impacts font arrangement, and color arrangement impacts itself.

ReView is a tool for finding causal relationships in anomalies in network traffic. It has been suggested to facilitate better understanding of network representations [33]. One of its features is minimizing the detailed information while showing the causal relationship. ReView can also quickly navigate the user through networks with a large number of requests and levels of abstractions.

Causal relationships are thought to be perceived directly if they are accompanied by animation [31]. The researchers followed Michotte's perception of causality principle which illustrates a causal event with a billiard ball hitting another ball and causing its motion. Ware and colleagues [31] introduced a visual causal vector method that shows a causal relationship between representations using animation. They found that the perception of causality depends on the simultaneous occurrence the visual causal vector and the change in the graphical representations.

1.3 Node-Link Graphs

Node-link graphs consist of nodes which are connected by edges. Large amount of work has been dedicated to visualizing those structures in the most readable way. However, the larger the network that is being illustrated, the higher the possibility that the graph becomes dense and unreadable. This is especially true when the direction of the edges is important [8]. When representing causal relationships with node-link graphs if node n is causing node m , then the edge between those two nodes will point towards node m . In order to avoid the complexity of the large network some researchers have tried to add interaction with the graph [11], while others' intention was to create visuals that do not change the structure of the network and does not require interaction [8]. An innovative interaction technique was designed by Abello et al [1] where a user is able to navigate through a hierarchically clustered base of the graph. In the non interactive technique researchers decreased the complexity of large directed graphs by replacing single edges with edges which connect to groups of nodes [8].

Data driven journalism is also concerned with difficulties in showing directed relationships in large datasets [25].

Nodes have also been used to represent "visual programming language" or workflow and processing functions [30]. The system translates the user input into "a sequence of data language", which will then be transformed into "service commands" that will be executed. The dependencies between the nodes and data flows are illustrated as interconnections between the vertices on the graphical user interface.

A collider "C" is well represented by an "inverted fork", $A \rightarrow C \leftarrow B$, in which the arrow shows a direct link from the tail factors to the head factor [16]. Stratifying on the collider is likely to induce bias and influence the the association between A and B. In addition, a similar pattern found in directed acyclic graphs, called a confounder, can be illustrated by a "causal fork" - $A \leftarrow C \rightarrow B$. Greenland [16] suggested that any change in the A-B link upon C-stratification will induce bias. Statistical adjustment for collider bias is suggested to lead to as much bias as not making adjustments for a confounder [20].

Directed acyclic graphs (DAGs) are increasingly used in representing causal relationships in modern epidemiology [29]. The factors in DAGs are connected by arrows which can never create a closed loop [17]. A path is defined by a sequence of arrows between the factors of interest. The DAG method is valuable as it is found to show underlying relations explicitly and easily identifies sources of confounding. Confounder is known to be a variable that is related with the exposure and the outcome but does not exist in the causal path between them. For example, chronic kidney disease and mortality are often caused by age. Confounding occurs when reading the causal relationships between chronic kidney disease and the outcome mortality.

1.4 Matrix-based Graphs

An Adjacency matrix is frequently used to represent a network [24]. If a network consists of n nodes, the matrix will consist of an $n \times n$ grid of cells. This is considered to be an unambiguous way of representing data. However, some of its disadvantages are that the area increases quadratically and as large networks are sparse there will be mainly empty space on the matrix. Matrices can be used to represent both directed and undirected graphs.

A new technique called Compressed Adjacency Matrices was introduced in 2012 for visualising gene regulatory networks [7]. As those directed networks have specific structural traits, standard representations such as adjacency matrices, and node-link diagrams are unable to depict all traits. Compressed Adjacency Matrices cut and rearrange adjacency matrix so that no space is wasted in case of sparse network. There are specific structures which represent sub networks. This is how scientists came up with a new data structure in order to fit the characteristics of the data they analyse.

Furthermore, PathwayMatrix is another visualisation tool that represents specific relations between proteins in a pathway [6]. The implementation of the tool consists of adjacent matrices that interact with each other. Additional features were added to facilitate the data analysis. This visualisation software received positive feedback in the specific area of representing relations in proteins pathways. Consequently, there is no one best representations technique. Depending on the dataset specifications, the complexity and size of the data there might be many or only a few sufficient ways to visualise it.

A performance comparison between square and triangular matrices has been conducted to measure performance speed and accuracy [23]. It was found that performance is influenced by the matrix juxtaposition type which leads to the creation of a new matrix visualisation called TileMatrix. It represents a large amount of matrices and is effective at analysing networks that are multi-faceted and time-varying. With TileMatrix it is easy to see differences in matrices across time and facets. However, triangular matrices can work only in non-directed networks.

Chapter 2

Introduction

Brain connectivity visualisations are in the form of weighted graphs which are node-link graphs in which each edge is given a numerical weight. Alper and colleagues [2] compared augmented adjacency matrix with node-link visualization by conducting a controlled experiment. They found that matrix-based graphs outperform node-linked graphs. It was concluded that for weight graphs, node-link representations are less readable and more error-prone when compared to matrices. On the other side, node-link graphs are adjustable to a specific spacial representation which might also be insightful when reading the graph. Overall, matrix-based graphs had higher accuracy.

Moreover, another paper suggested that adjacency matrices are superior to node-link graphs in representing dense graphs because they are more compact and easier to look at [27]. However, they also stated that node-link graphs are better for path finding as a path can easily be followed if the arrows are not too tangled.

Adjacency matrices and node-link diagrams have been compared in another study [21] to examine which is more suitable graphical representation for the general public. This research was provoked by the fact that matrices are mainly use in engineering area, while node-link graphs are generally a more popular way of visualisation. The research questions being examined were related to the attributes of the connectivity model influencing readability and which representation is more suitable for particular tasks. In addition, they filled the gap in previous research that did not take into consideration participants' familiarity with the data sets. It was found that error rates and response time are highly influenced by size and density. They confirmed Ghoniem et al () findings that matrix-based graphs outperform node-link graphs for dense and large graphs, and node-link graphs are more readable for small and sparse graphs. The only exception found is finding the path between two nodes in a graph. Furthermore, experience with the data set showed to has effect on performance.

The main advantages of a matrix-based graph are that it has no overlapping and can be ordered. Therefore, it was predicted that tasks involving link finding and node finding will be better performed in matrix-based graphs rather than node-link graphs. Furthermore, node-link and matrix-based graphs were compared [12]. They also predicted that counting nodes should be equally difficult on both, while counting links and finding the most connected node should be easier in matrix-based representations. Node-link graphs should be better when building a path between two nodes. The reason for this is that matrix based graphs have the nodes represented twice which introduces extra complexity. Ghoniem and colleagues [12] also hypothesised that node-link graphs will be easier to work with when dealing with graphs with a small amount of graphs. In their experiment they had three sizes of graphs - 20, 50, and 100 nodes. They tested 36 participants and measured their performance on various tasks such as finding paths, neighbours, nodes and links between nodes. They found that performance in node-link graphs decreases as the size of the graph increases. This pattern was confirmed for all of the tasks except for finding path, where node-link graphs regardless of the size and density showed better results.

The present research aims to compare readability in different layouts in node-link and matrix-based graphs

representing causal relationships. Following the literature review it is intriguing to explore which representations will show better performance and whether this will be modulated by factors such as layout, question domain, size and task.

Chapter 3

Implementation details

3.1 Software tools and technologies

Given the opportunity to choose any tools and technologies for the development of this web application was a very exciting task. However, I had to be certain that the right decisions are made. After a research period followed by a trial-error week it was decided that the backend of the application will be built with Node.js and JavaScript combined with the web application framework Express. Node.js was chosen on the grounds of being event-driven, non-blocking I/O model which contributes to a very efficient and lightweight software. A Node.js JavaScript engine is also used in the Google Chrome browser. JavaScript servers have incredible performance due to their asynchronous I/O. Node.js appears to be single-threaded from a developer's point of view, as there is no thread management involved in the development process. However, behind the scenes Node.js handles threading, file system events, implements the event loop, feature thread pooling etc. Coming from a Java background, the Maven equivalent in Node.js is NPM. By using NPM commands the developer is able to install a variety of different modules to help the implementation process. NPM executes the function of a package manager. Express is the standard server framework for Node.js. It is usually described as a minimal and flexible Node.js web application framework. Many popular frameworks such as KeystoneJS, Kraken and Sails, are built on Express.

AngularJS 1 was chosen for management of frontend functionality. Even though there is a newer version of the product, the lack of documentation and support online, was a sufficient reason for using the older AngularJS. It uses HTML as a template and enables the developers to extend it to express the application's components clearly. AngularJS supports features such as data binding and dependency injection which decreases the amount of code that a developer would usually write to implement them.

The database system chosen for the project is PostgreSQL. Considering the size of the project an object-relational database was chosen. In addition, it decided on PostgreSQL in particular because it is open source and has gained a reputation of a reliable database system. Also previous experience with PostgreSQL from developers point of view made the decision easier.

* maybe database schema will be added here *

3.2 Development Process

3.2.1 User Stories

The first step of the implementation was understanding the requirements and creating user stories. Some of the most important user stories are:

- As a researcher, I want to be able to see participant's answers to questions, so that I can analyse the data.
- As a participant, I want to be able to see a graph and a questions at a time, so that I can complete the experiment.
- As a researcher, I want to keep participant's scores anonymous, so that the experiment comply with ethics requirements.
- As a participant, I want to be unable to go the next question before completing the present question first, so that I am certain that I haven't missed a question.
- As a researcher,

3.2.2 Wire-frames

After the requirements gathering analysis, development of wire-frames followed. Balsamiq Mockups 3 is the software used for the creation of wire-frames. An example of the research question page can be found in figure 3.1.

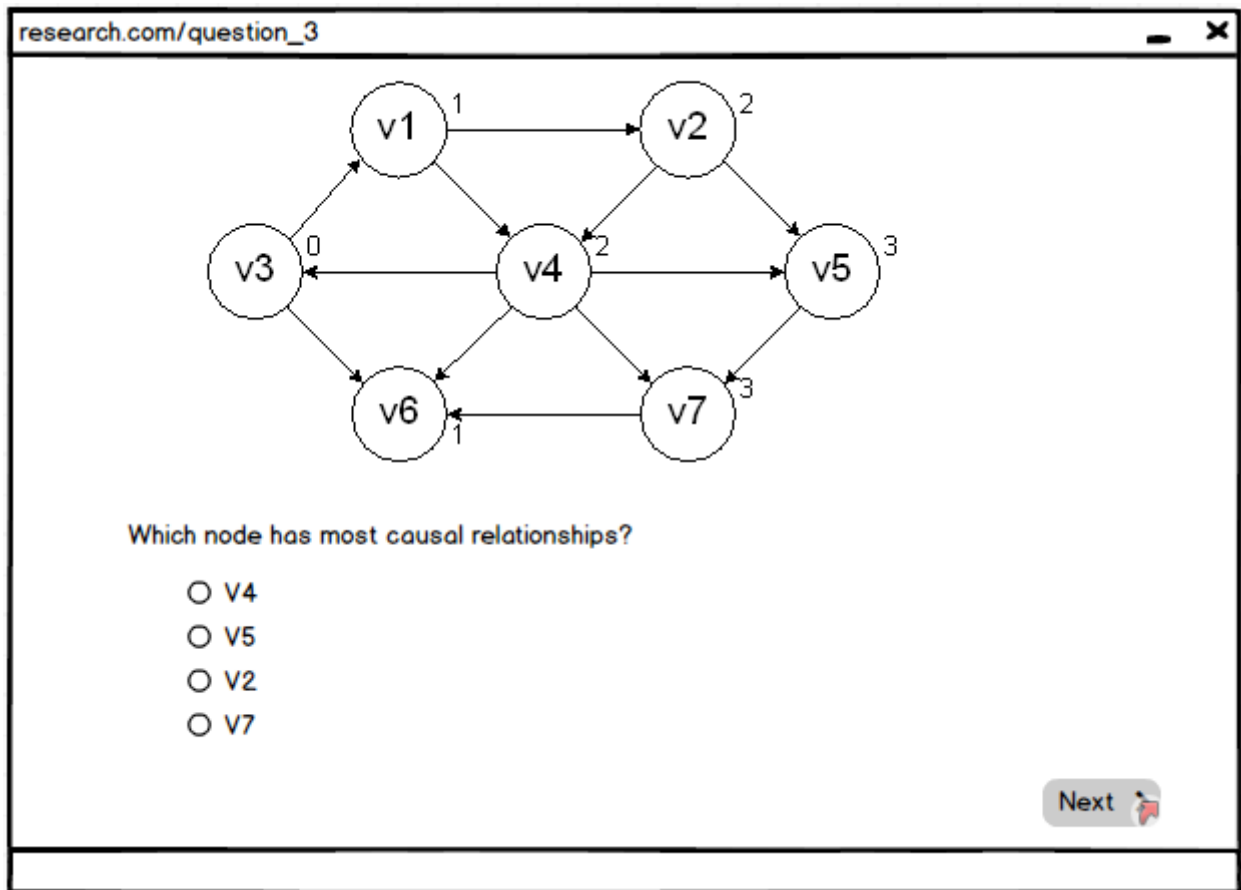


Figure 3.1: Research Question Wire-frame

After discussing the wire-frames it appeared that some important features are missing. One of those features was a participants training session. The aim of the experiment is to test which graph is more readable for people who do not have regular exposure to such data visualisation. Therefore, it is important to make the participants aware of how to read each graph before the actual experiment. This way, the requirements specification became an iterative process during which a better understanding of the product evolved.

3.2.3 System Design

Designing the system was the next stage in the process.

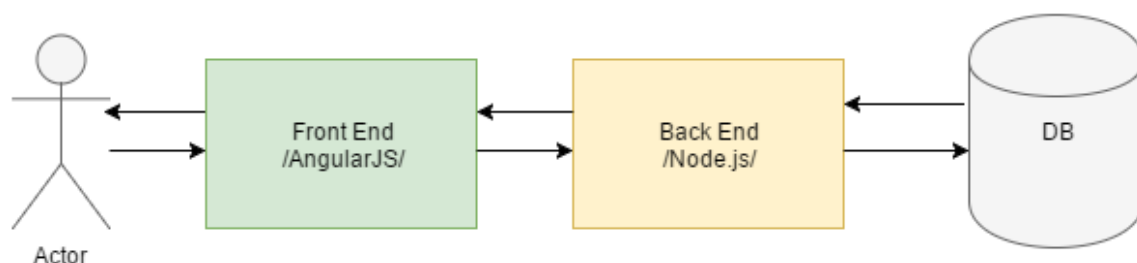


Figure 3.2: An abstract representation of the system design

Figure 3.2 shows an abstract view of the system design. There is an Actor who will either be a participant

in the study or a researcher. They will interact with the front-end which will be in the form of a web application in a browser. The front end will communicate with the back-end which will be implemented in Node.js. The back-end will make requests to the database to retrieve and send information.

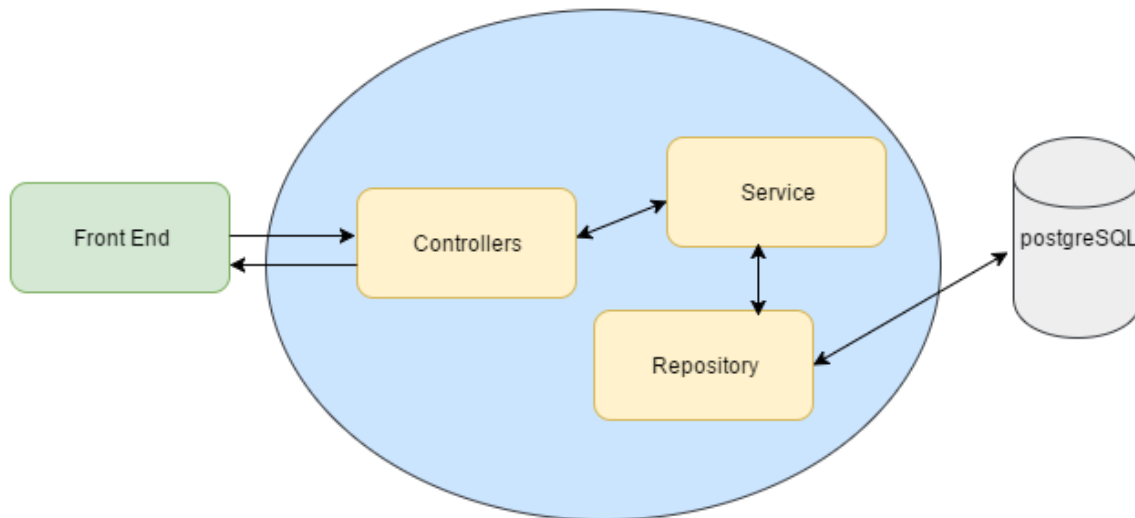


Figure 3.3: A more specific representation of the system design

Figure 3.3 displays a more detailed version of the system design. This particular design has been implemented to separate the different concerns in this specific system. When the Actor interacts with the application, the front-end will send information to the Controllers. There are many controllers because there is a controller for each page with front-end functionality. The Controller component decides what should the next action be according to the user input. It has control over the front-end logic and sending requests to the service if information from the database or the server is requested. For example, the answers to all questions are kept in the front-end until a "Submit" button is clicked on. This action triggers a request to the Service. The Service component works with the back-end logic. It can send and retrieve data from the database and keep the information in the Repository. The Service also deals with the requests for the different web-pages. Also it ensures that the project dependencies are loaded.

3.2.4 Implementation

The development process was split into front-end and back-end. Without using any frameworks, the front-end Html pages were created following the wireframes. Bootstrap was added to the html to improve the UI design and make it look more appealing for the participant. Furthermore, a hierarchical page set up was implemented by having all html pages extending one layout file. This also helped avoid repeated code as needed libraries were imported only once and all other files inherited them. A considerable amount of time was spent on developing the graphs and converting node-link graphs into matrix-based graphs. The different sizes were evolving from each other, which means that the information contained in the small graph was present in the medium one. Close to piloting the experiment, the front end had to be drastically changed in order to fit the large graphs on one screen.

By this time, it was clear that Node.js will be used to create a server so the next step was to implement it. The decision to use Express as a framework with Node.js followed and the html pages were mapped to a handlebars or hbs files.

The following couple of weeks were dedicate on work on the database system: creating database schema, and tables, and work on connecting it with the server.

3.3 Challenges

Spring idea failed
changed to Node.js
Angular compatibility with Node.js

In the beginning of this project the Java framework Spring was going to be used in the implementation as it is among the most widely used frameworks in industry [32]. This decision was supported by extensive previous experience with Java from developer's point of view and the applicability of the skills to be acquired. However, one of the reasons why Spring is used in industry is because of the large and complex systems that exist there. The Spring framework works on a very high level of abstraction where you can easily write configuration files to add dependencies from different project. Therefore, it is considered rather unfriendly for small independent projects and developers with limited Spring experience. The reasoning behind this conclusion was provoked after a couple of unsuccessful attempts to set relative paths to different CSS and JavaScript files. The issue was found to be in the web application configuration file. This is how the very simple task of reading a css file turned to be a long tedious debugging process after which the realisation that Spring is unnecessary abstract and complex for this project occurred.

A new research for web-application frameworks followed. Node.js backend was chosen because of its event-driven, non-blocking I/O model which creates an efficient and lightweight server-side of the application. Another challenge appeared when trying to incorporate AngularJS with Node.js. Usually in AngularJS one uses curly braces to reference data structure from the AngularJS controller. However, Node.js also uses curly brackets to reference information from the backend in the frontend. After a long research it was found that Node.js overrides the use of curly braces and the application is not displaying Angular data as it expects it come from the backend. Unfortunately, an appropriate error message does exist and it all had to be discovered during the development process. Instead of using curly brackets one can also use "ng-bind" and achieve the same result. This approach solved the issue until "ng-bind" information was needed in "ng-src" to display the appropriate graph image. It is not possible to use "ng-bind" inside "ng-src" so the present solution at the time was no longer solving the problem. Therefore, the Angular configurations had to be altered to use a different symbol. Implementing this solved the problem entirely.

Another issue was the size of the graphs. During the implementation process it was necessary that a scroll bar appears in order to show the real size of the graphs. This was inconvenient for reading the graphs, and participants were going to need more time to read these graphs. Therefore, this was going to be a confounding variable in the experiment. In order to avoid this issue, the front-end pages had to be restructured. The bootstrap navigation bar had to be removed and the question had to be displayed on the left instead of below the graph. This also meant that this particular sequence of pages had to extend a different layout page, which does not contain the navigation bar and has a different bootstrap grid.

3.4 Software reliability testing

Chapter 4

Evaluation

4.1 Design

This is a within subject design experiment with two conditions. In the first one participants answer questions related to causal relationships in node-link graphs and in the second one they are asked to answer similar questions on matrix-based representation. Each condition has three factors - size, layout, and question domain. Figure 4.1 illustrates the pattern which was used to create the questions and the graphs. The domain questions are healthy and unhealthy gym behaviour, drinking issues, and student exams. The three different sizes were small (10 nodes), medium (20 nodes) and large (30 nodes). For each of these, there were three different layouts for both matrix-based and node-link graphs. The matrix-based layouts are alphabetical, in degree descending and out degree descending. The node-link layouts are hierarchical, orthogonal, and series parallel. There was a direct relationship question and an indirect causal relationship question asked for each of the created graphs. The first four questions of the experiment are trial questions which do not count towards the final results.

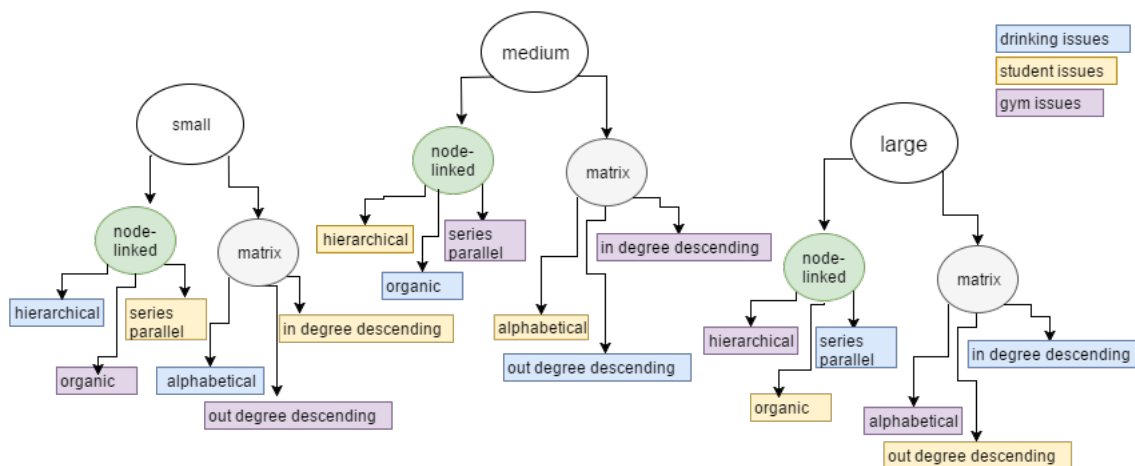


Figure 4.1: Experimental Design

4.2 Stimuli

The specified sequence of questions with matched layouts, question themes, and different sizes was likely to influence the results of the experiment because of an exposure effect of that particular sequence. Therefore, to

ensure that all participants receive the questions in a different order, Latin square randomization was used. The sequence of questions was different for each participant and the possibility of the questions sequence affecting the results was eliminated.

Node-link graphs were created using yEd, and the equivalent matrix-based representations were created using software which converts XGML node-link graphs into matrix-based graphs. As long as the graphs are from one domain questions, they all represent the same information. However, all questions require different information to be read from the graphs in order to avoid learning effect. At the same time the the graphs represent the same information which ensures that the complexity of the data is the same. Both types of representations are displayed in their real size on the left side of the screen and the corresponding question is on the right. Special attention is paid to the fonts of the letters and retaining the original size of the image. Participants were not able to go to the next question before they answered the current one.

4.3 Pilot

Conducting a pilot was an essential part of this study. It brought light into how people who have never been exposed to the graphs and the software before interact with them. It was vital to learn whether the tasks are clear and the software is easy to use. Three volunteers took part in the pilot. They were asked to sit on a chair in a silent environment and complete the experiment starting from the information sheet and finishing with the debrief form. At the end the participants were asked questions regarding the training, the clearness and enjoyment of the tasks. All participants said that the training in the beginning of the experiment is sufficient at explaining how to read the graphs. One of the participants, however, wanted to confirm with the experimenter that the correct direction for reading the graphs is from left to top. This is why, it was decided to turn the first three questions in trial questions that the participants are aware of and informed that they can ask as many questions as they need in order to answer the questions. The second three questions (4,5,6) are also going to be trial questions in order to give them more practice questions but they won't be explicitly informed about that. Consequently, during the first six questions would be expected to fully learn how to interpret the graphs and their answers will not be recorded and included in the results.

An interesting problem regarding the nature of the trial questions was also considered. Initially, it was suggested that the trial instances are always the first 4 questions of each sequence. However, as those questions would not be accounted for in the results and the Latin Square randomizes each sequence, this was going to lead to uneven number of answer for each type of graph. If this was to happen, the results of the experiment were going to be negatively influenced. In order to avoid this, 6 new questions were created using a combination of layouts, sizes and domains that has not been used for the original 36 questions. A mixture of different sizes and types was included to ensure that the participants have exposure to the main challenges of the experiment during the practice trials. These 6 questions were going to be added at the beginning of each sequence so that all participants are exposed to the same trial questions.

All participants in the pilot said that the tasks are clear, the graphs font size is readable, and different layouts and sizes are appropriately displaying the data. One participant mentioned that in the organic layout in one of the node-link graphs, the label is on the arrow which makes it hard to read. The reason for this is that both labels and arrows are in black colour. Their note was taken into consideration but if changes are made to align the arrow and the label differently this would alter the organic yEd layout. Consequently, the results will provide information about a manually created layout similar to the organic yEd layout. Another solution of this issue is to change the label names to letters instead of actual factors such as "depression", and "pass exams". However, this was going to influence the complexity of the experiment and the participant's level of interest. Thus, no changes were made to the organic layout.

Furthermore, the pilot brought light into how entertaining the tasks are. It is vital to have the participants

engaged with the tasks to ensure that they are concentrated and not easily distracted by third factors. Two participants said that the experiment was interesting and were eager to answer the questions. They said that the topics were easy to understand and it was interesting to find the causal relationship. One participant said that they do not consider the task to be very interesting, but they were in a hurry for another meeting and likely to look for an excuse to leave earlier.

One volunteer in the pilot mentioned that sometimes it is hard to find the label you are looking for and they guessed the answer to the question by logical reasoning. To avoid this problem in the experiment, a note will be made to tell participants that there is no logical relationship between the causal relationships and they should not attempt to guess the correct answer. This is supposed to encourage them to look for the labels rather than guess the correct answer. It was also noted, that when a particular node has more than one causal relationship, it is much harder to follow a path, than in a complex matrix-based graph with many relationships. It will be interesting to see whether this observation will be confirmed by the experimental results.

The pilot was also extremely helpful for spotting technical issues. If it was not conducted and some of these problems were not accounted for, they were going to have catastrophic influence on the results. First, it was found that in one question the correct answer was being evaluated to incorrect because of a typo in the csv file that populated the database. In addition, the answers were not written in a consistent way, which meant that the participants might have been influenced to choose the answer that is written differently from the rest of the answers. In order to fix these problems, the database table with the questions had to be deleted, and the csv file containing the questions' information needed to be reviewed. Finally, the database table had to be populated again with the updated data.

Another technical issue found was concerned with the time recorder. It was found that the timing was starting and stopping when required, but the record of the time taken for a particular question was wrong. The issue was found to be caused by a wrong `startTime` variable during the `timeTaken` calculation. This was a scope issue, which was fixed for the real experiment. Moreover, in order to submit their answers the participants had to click a "Next" button. The timer used to stop once this button is clicked, record the time taken for the current answer, and start the timer for the next question. However, it was found that this extra click influences the recorded time and it would be more appropriate to have the time start and stop ones the participants choose an answer. This led to the decision to completely remove the "Next" button and submit the answers when one of the radio buttons has been selected. On that click the next graph and question would be loaded. It is not natural for the participants to expect clicking on a radio button to submit their answer, and show the next question, but they will be instructed about this feature in the information sheet and will have 6 trial questions to get used to that functionality.

In general, the pilot helped in identifying small issues regarding the graphs layout and the manner in which questions were asked. Other bigger issues, that were going to influence the results, were also found such as typos leading to wrong evaluation of correctness and inaccurate time recorder. The pilot also inspired a new way in which the trial questions should be created and accounted for. The present experiment would not have been as accurate and precise if the pilot was not conducted. If the changes that it provoked implementing were to be found in the process of testing participants, the testing had to be started all over again and new participants had to be recruited.

4.4 Participants

There were [] participants who took part in the experiment, aged between [-]. The excluding criteria restricted people specialized in the subjects of Maths, Engineering and Computing Science from participating. The reason for this is that these fields are likely to include preparation in reading graphs. In addition, findings are expected to be representative for health and social sector, who are not likely to have background in those degrees.

4.5 Procedure

In order to have a more environmentally valid experiment, the study was conducted only on the experimenter's laptop in their presence. The participants were asked to sit on a chair and complete the experiment in a calm, silent environment. They had to read through an information sheet, explaining what is to following in the next 30 minutes, consent from, asking them to agree to participate in the study, and training showing them examples of how to solve the upcoming 36 research questions. At the end of the experiment participants were shown a short demographic questionnaire and a debrief.

4.6 Results

4.7 Discussion

Appendices

Appendix A

Database Schema

```
participants_answers(question_id, participant_id, answer, time)
participants_info(participant_id, participant_name, email, uni_degree, age)
questions(question_id, question, one, two, three, four, correct, image)
```

Appendix B

Running the Programs

An example of running from the command line is as follows:

```
> java MaxClique BBMC1 brock200_1.clq 14400
```

This will apply *BBMC* with *style* = 1 to the first brock200 DIMACS instance allowing 14400 seconds of cpu time.

Appendix C

Generating Random Graphs

We generate Erdős-Rényi random graphs $G(n, p)$ where n is the number of vertices and each edge is included in the graph with probability p independent from every other edge. It produces a random graph in DIMACS format with vertices numbered 1 to n inclusive. It can be run from the command line as follows to produce a clq file

```
> java RandomGraph 100 0.9 > 100-90-00.clq
```


Bibliography

- [1] James Abello, Frank Van Ham, and Neeraj Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, 2006.
- [2] Basak Alper, Benjamin Bach, Nathalie Henry Riche, Tobias Isenberg, and Jean-Daniel Fekete. Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 483–492. ACM, 2013.
- [3] Katy Börner, Adam Maltese, Russell Nelson Balliet, and Joe Heimlich. Investigating aspects of data visualization literacy using 20 information visualizations and 273 science museum visitors. *Information Visualization*, page 1473871615594652, 2015.
- [4] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.
- [5] A.J. Cardno, P.S. Ingham, A. Lewin, and A.K. Singh. Data visualization methods, July 10 2014. US Patent App. 14/200,903.
- [6] Tuan N Dang, Paul Murray, and Angus G Forbes. Pathwaymatrix: visualizing binary relationships between proteins in biological pathways. In *BMC proceedings*, volume 9, page S3. BioMed Central Ltd, 2015.
- [7] Kasper Dinkla, Michel A Westenberg, and Jarke J van Wijk. Compressed adjacency matrices: untangling gene regulatory networks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2457–2466, 2012.
- [8] Tim Dwyer, Nathalie Henry Riche, Kim Marriott, and Christopher Mears. Edge compression techniques for visualization of dense directed graphs. *IEEE transactions on visualization and computer graphics*, 19(12):2596–2605, 2013.
- [9] Ronak Etemadpour, Robson Motta, Jose Gustavo de Souza Paiva, Rosane Minghim, Maria Cristina Ferreira de Oliveira, and Lars Linsen. Perception-based evaluation of projection methods for multidimensional data visualization. *IEEE transactions on visualization and computer graphics*, 21(1):81–94, 2015.
- [10] Jean-Daniel Fekete, Jarke J Van Wijk, John T Stasko, and Chris North. The value of information visualization. In *Information visualization*, pages 1–18. Springer, 2008.
- [11] Emden R Gansner, Yehuda Koren, and Stephen C North. Topological fisheye views for visualizing large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):457–468, 2005.
- [12] Mohammad Ghoniem, J-D Fekete, and Philippe Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 17–24. Ieee, 2004.
- [13] Alyssa A Goodman. Principles of high-dimensional data visualization in astronomy. *Astronomische Nachrichten*, 333(5-6):505–514, 2012.

- [14] Evgeniy Yur'evich Gorodov and Vasiliy Vasil'evich Gubarev. Analytical review of data visualization methods in application to big data. *Journal of Electrical and Computer Engineering*, 2013:22, 2013.
- [15] Lars Grammel, Melanie Tory, and Margaret-Anne Storey. How information visualization novices construct visualizations. *IEEE transactions on visualization and computer graphics*, 16(6):943–952, 2010.
- [16] Sander Greenland. Quantifying biases in causal models: classical confounding vs collider-stratification bias. *Epidemiology*, 14(3):300–306, 2003.
- [17] Sander Greenland, Judea Pearl, and James M Robins. Causal diagrams for epidemiologic research. *Epidemiology*, pages 37–48, 1999.
- [18] Taylor Hanken, Sam Young, Karen Smilowitz, George Chiampas, and David Waskowski. Developing a data visualization system for the bank of america chicago marathon (chicago, illinois usa). *Prehospital and Disaster Medicine*, pages 1–6, 2016.
- [19] Chun-Cheng Hsu. Evaluation criteria for blog design and analysis of causal relationships using factor analysis and dematel. *Expert Systems with Applications*, 39(1):187–193, 2012.
- [20] Imre Janszky, Anders Ahlbom, and Anna C Svensson. The janus face of statistical adjustment: confounders versus colliders. *European journal of epidemiology*, 25(6):361–363, 2010.
- [21] René Keller, Claudia M Eckert, and P John Clarkson. Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? *Information Visualization*, 5(1):62–76, 2006.
- [22] John King, Kathy Sonderer, and Kevin Lynch. Cognitive benefits of a simple visual metrics architecture. In *International Conference on HCI in Business, Government and Organizations*, pages 319–329. Springer, 2016.
- [23] Xiaotong Liu and Han-Wei Shen. The effects of representation and juxtaposition on graphical perception of matrix visualization. In *Proceedings of the 33rd annual ACM conference on Human factors in computing systems*, pages 269–278. ACM, 2015.
- [24] William JR Longabaugh. Combing the hairball with biofabric: a new approach for visualization of large networks. *BMC bioinformatics*, 13(1):1, 2012.
- [25] Christina Niederer, Wolfgang Aigner, and Alexander Rind. Survey on visualizing dynamic, weighted, and directed graphs in the context of data-driven journalism. *Proceedings of the International Summer School on Visual Computing*, pages 49–58, 2015.
- [26] Judea Pearl. An introduction to causal inference. *The international journal of biostatistics*, 6(2), 2010.
- [27] Zeqian Sheny and Kwan-Liu Maz. Path visualization for adjacency matrices. In *Proceedings of the 9th Joint Eurographics/IEEE VGTC conference on Visualization*, pages 83–90. Eurographics Association, 2007.
- [28] Ben Steichen, Giuseppe Carenini, and Cristina Conati. User-adaptive information visualization: Using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 317–328, New York, NY, USA, 2013. ACM.
- [29] Marit M Suttorp, Bob Siegerink, Kitty J Jager, Carmine Zoccali, and Friedo W Dekker. Graphical presentation of confounding in directed acyclic graphs. *Nephrology Dialysis Transplantation*, 30(9):1418–1423, 2015.
- [30] Balaji T Thattai, Karthikeyan Shanmugam, Chi Yao, and Yee Lang Lau. Systems and methods for generating data visualization applications, February 24 2016. US Patent App. 15/052,449.
- [31] Colin Ware, Eric Neufeld, and Lyn Bartram. Visualizing causal relations. In *Proceedings of IEEE Information Visualization*, volume 99, 1999.

- [32] Kuikui Liu, Xiuji Shi, and Yue Li. Integrated Architecture for Web Application Development Based on Spring Framework and Activiti Engine. *The International Conference on E-Technologies and Business on the Web (EBW2013)*, pages 52–56, 2013.
- [33] Hao Zhang, Maoyuan Sun, Danfeng (Daphne) Yao, and Chris North. Visualizing traffic causality for analyzing network anomalies. In *Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics, IWSPA '15*, pages 37–42, New York, NY, USA, 2015. ACM.