



Matrices or node-link diagrams: which visual representation is better for visualising connectivity models?

René Keller¹
Claudia M. Eckert¹
P. John Clarkson¹

¹Engineering Design Centre, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, UK

Correspondence: René Keller, Engineering Design Centre, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, U.K.
Tel.: +44 1223 332828;
Fax: +44 1223 766956;
E-mail: rk313@cam.ac.uk

Abstract

Adjacency matrices or DSMs (design structure matrices) and node-link diagrams are both visual representations of graphs, which are a common form of data in many disciplines. DSMs are used throughout the engineering community for various applications, such as process modelling or change prediction. However, outside this community, DSMs (and other matrix-based representations of graphs) are rarely applied and node-link diagrams are very popular. This paper will examine, which representation is more suitable for visualising graphs. For this purpose, several user experiments were conducted that aimed to answer this research question in the context of product models used, for example in engineering, but the results can be generalised to other applications. These experiments identify key factors on the readability of graph visualisations and confirm work on comparisons of different representations. This study widens the scope of readability comparisons between node-link and matrix-based representations by introducing new user tasks and replacing simulated, undirected graphs with directed ones employing real-world semantics.

Information Visualization (2006) 5, 62–76. doi:10.1057/palgrave.ivs.9500116

Keywords: Graph visualisation; readability; evaluation; design structure matrix; engineering change

Introduction

Connectivity models play an important role in many disciplines. Their application ranges from social networks¹ and street networks² to models that predict the spread of epidemics³ and product models in engineering design.^{4,5} The underlying structure of such a connectivity model is a graph or a relational data set that describes interactions between the elements of the model. The interpretation of these links depends on the application. Examples of how to interpret links are task precedence information in process models,⁶ internet traffic between websites,⁷ change links between components of a product⁴ or friendship relations in a social network.¹

Connectivity models can be used in various ways. Firstly, they are a means to store relevant information about the piece of reality that is being modelled. The information can be communicated to other users of the model, they are then serving as boundary objects,⁸ or further calculations can be performed to gain further insights, for example, PERT (program evaluation and review technique) analysis of task networks.⁹ Additionally, the act of building connectivity models alone (for instance, modelling a process) can be beneficial as it can indicate problems that might be

Received: 31 August 2005
Revised: 8 December 2005
Accepted: 2006
Online publication date: 7 April 2006

overlooked otherwise. Matrix-based methods (especially the design structure matrix (DSM)) are popular among the engineering research community and attempts are being underway to introduce techniques based on DSMs into industry. This paper will focus mainly on this kind of matrix. However, the implications are equally valid for a wider scope of matrices describing a graph structure, such as adjacency matrices.

To make efficient use of these models, it is important that necessary information can be extracted easily. For example, it should be possible to read relevant data from the visual representation of the underlying graph structure with little effort and with or without computer support. Both matrix-based representations and node-link diagrams are equally valid visual representations of graphs as they can show the same relational information. This raises two questions: which representation is more suitable for showing certain properties of a graph, and which attributes have an influence on the readability in each representation? Here we concentrate on simple information retrieval tasks that can be read from the visual representation without deep analyses and which allow users to quickly assess relevant attributes of the underlying graph structure. For example: finding out how many other nodes a particular node is connected to can give a quick indication of how integrated the node is with the entire graph.

This paper reports the results of two empirical studies. The first experiment was conducted in order to identify key attributes that influence the readability of matrix-based representations of graphs. The second experiment, the major contribution of this paper, examines which representation is more suitable for displaying certain aspects of the relational structure.

Terminology

This paper concentrates on one particular type of model: connectivity models with an underlying directed graph, referred to here as a 'graph'. Such a graph consists of two sets of elements: a set of nodes or vertices, and a set of (directed) links between nodes. Each link has a tail node, which can be interpreted as the origin of the relation, and a head. Usually a relation describes an influence that the tail exerts on the head. A neighbour of a vertex is a node that has any kind of relation to it. The size of a graph refers to the number of nodes n and the density of the graph refers to the number of links m divided by the number of possible links in the case of a directed graph (Figure 1): $d = \frac{m}{n(n-1)}$.

Change as an example for applications of connectivity models

The design of complex products often involves the coordination of hundreds of person years over multiple sites to assure the quality creation of complex products with thousands of parts, which need to work reliably over many decades. Practitioners struggle to understand the complexity of the products and processes involved in

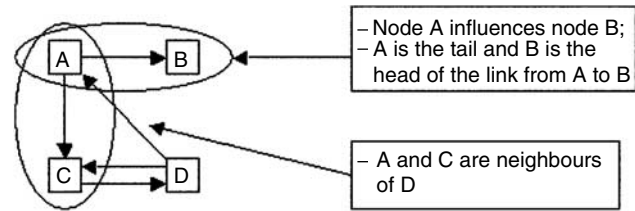


Figure 1 Graph terminology used throughout this paper.

them, and they require accessible and intuitive models to interact with the information describing these complex products and processes.¹⁰ Simon characterises engineering products as 'almost decomposable' systems,¹¹ which inherently have a high degree of connectivity. The same applies to the processes used to generate them. Consequently, connectivity models can effectively describe both design processes^{6,12} and component connectivity in complex products.^{4,13} The application of component connectivity models to assess the risk of component changes is relatively new, and yields interesting findings, such as improved strategies for presenting change data^{14,15} or change prediction.¹⁶ Change models are used here as an example as they have attracted some attention recently, and because a number of real-world models are available (see the diesel engine model described later, that was originally elicited to predict change propagation using the change prediction method (CPM)).

The need for a proper representation of connectivity models (which are essentially graphs) forms the motivation for comparing the different representations introduced in this paper. It is of great benefit to understand factors that influence the readability of such representations (see experiment 1), and the comparison of the two most common representations for graphs (experiment 2) should allow the selection of the best representation given a particular task. The tasks introduced later for the second experiment reflect how easily information about direct links (finding a link or a node or the number of in and outgoing links) and indirect ones (common neighbours and shortest paths) can be extracted from these representations.

Most of the tasks that are used for analysis later in this paper represent very simple information retrieval exercises. More complex tasks such as finding clusters, which can be beneficial for some applications, are not considered. In applications such as predicting change propagation, where overlooked hidden links can cause problems, representations that support more complex tasks (by reordering the entries in a matrix for example) can further draw attention away from these hidden links. A matrix, that shows clusters (and thus probably the subsystems of a product), will make readers assume that the clusters are independent and change propagation caused by a part of one cluster can be contained by changes within the cluster only. However, change propagation can happen through a number of

component connections, within and outside of the cluster and vital information in the in-between cluster links is hidden.

Most computer tools can easily provide the user with the results of the simple tasks used in this paper. However, in many cases, the representations are used as boundary objects⁸ in the design process, which facilitate the negotiation between different groups or people in an organisation. The model of the product or process is taken (in paper form) to other designers and information is shared based on this representation alone. In this light, it is important that also simple, mundane tasks, such as finding the shortest path between two components and counting the number of in or outgoing connections from a node, can be very important. For example, the matrix representation of a diesel engine model used by a U.K. company hides the indirect link between the *Fuel Injection Assembly* and the *Wiring Harness*, which could be easily shown by a node-link representation. As it turns out with more thorough analyses,¹⁵ this is one of the riskiest component connections found in the entire design and overlooking it can cause costly problems. Such indirect dependencies can also be observed in process visualisations and other simple information retrieval tasks have a counterpart in the analysis and visualisation of process models that share the same underlying data structures.

Visual representations of connectivity data

A directed graph can be displayed through a number of different representations, which make different aspects of the same data set salient. Examples include adjacency matrices, node-link diagrams and incidence matrices.

Adjacency matrices (DSMs)

Adjacency matrices, which engineers call design structure matrices (DSMs), offer a very compact representation of graphs. These and other matrix-based techniques, such as N-square diagrams, used in Systems engineering,¹⁷ are popular in engineering research¹⁸⁻²⁰ and researchers are pushing DSMs into industrial use. A DSM is a square matrix, where each node of the underlying graph is represented by a row and a column. An entry in a DSM means that there is a link from the node represented by a column of the matrix to the node represented by a row.

However, this definition varies among different research communities, as it is not inherent in the matrix how it should be read: in some disciplines, it is read from row to column. Figure 2 shows two similar binary DSMs of the core components of a simple car engine and their spatial relations. For example, the ringed crosses mean that there is a link from 2 (crankshaft) to 3 (cylinder block). This layout resembles standard layouts of DSMs used in engineering software tools.^{4, 21} The black boxes on the diagonal are a visual cue for very large matrices.

In such matrix-based techniques, the possible number of different layouts is restricted to the order of the elements in the horizontal and vertical directions.²² Industrial case studies indicate that once an order for the components is established, subjects find it especially easy to find a component in a DSM they have seen before. This behaviour was also recognised when users were instructed to find menu items.²³ Typically, elements of a DSM are ordered in some obvious way, for instance alphabetical order, or are ordered to make particular features salient, or at least to be immediately understandable to the user.

Changing the order of the elements of the matrix²² can be beneficial for further analyses of a DSM. Techniques such as sequencing or clustering⁵ of the matrix change the given order to show aspects of the data that cannot be easily examined within the original view (see Figure 2 for two different node orders of the same DSM).

Node-link diagrams

In a node-link diagram, each node of the underlying graph is represented as a node, and arrows between nodes represent links between them (see Figure 3 for node-link representations of a simple car engine model). For the layout of such a node-link diagram, the entire two-dimensional space can be used. Thus, the number of potential layouts is much larger than with a DSM. This larger variety of possible layouts allows one to focus on different aspects of the data, especially for large graphs.

An extensive collection of possible layout algorithms for graph data using node-link diagrams can be found in di Battista et al.²⁴ See Figure 3 for some examples of different layouts for simple node-link diagrams describing a car engine.



Figure 2 Two different DSMs of a car engine, (A) alphabetical order, (B) clustered.

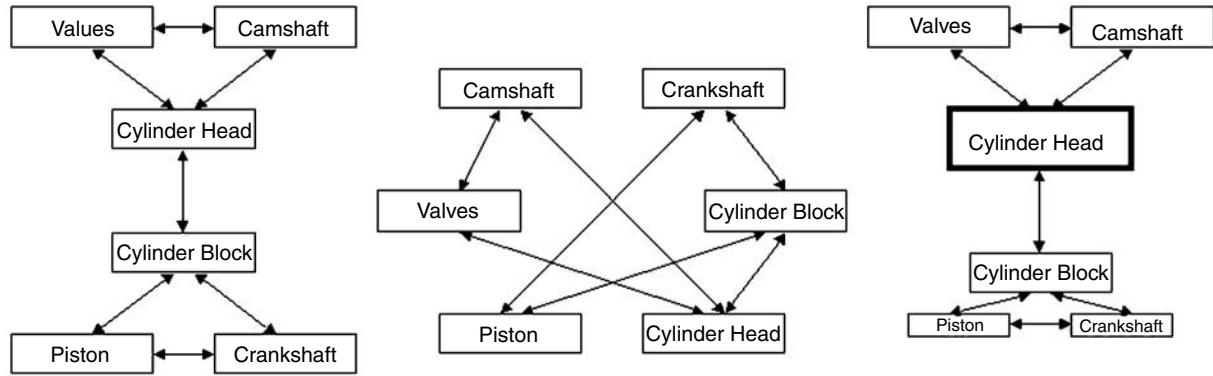


Figure 3 Different node-link diagrams of a car engine.

| Simple Engine | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Camshaft | -1 | -1 | | | | | | 1 | | | | | | 1 |
| Crankshaft | | | -1 | -1 | 1 | | | | | | | 1 | | |
| Cylinder Block | | | 1 | | -1 | -1 | -1 | | 1 | | | | | |
| Cylinder Head | 1 | | | | | 1 | -1 | -1 | -1 | 1 | | | | 1 |
| Piston | | | | 1 | | | 1 | | | | -1 | -1 | | |
| Valves | | 1 | | | | | | | | 1 | | | -1 | -1 |

Figure 4 Incidence matrix of the car engine model.

Displaying graphs within a node-link representation, however, has some disadvantages and problems. Different layouts, for instance, create ambiguity. In the cases of very large and dense graphs especially, the problems of edge-crossings and overlapping nodes can be very severe, as reading the display becomes extremely complex. While small and sparse graphs can often be drawn without edge-crossings (these graphs are then called planar; see Figure 3, left for a planar drawing of the simple car engine model), especially large graphs and graphs that are highly connected cannot be laid out properly.

Incidence matrices

Incidence matrices are another matrix-based format for displaying graphs. These are rectangular (not necessarily squared) matrices with each row representing a node of the graph and every column representing a link. In every column, ‘-1’ represents the tail of the relation, ‘1’ shows the head of the relation. This means that in digraphs there are exactly two entries in every column of the matrix, ‘-1’ and ‘1’. See Figure 4 for the incidence matrix of the simple car engine model.

The incidence representation of graphs is usually the basis for extensions of the standard digraph notation to include relationships between more than two objects. Using incidence matrices it is, for instance, easy to display concepts like higraphs,²⁵ where each link can connect more than two vertices. However, we decided not to include the incidence matrix in the experiments reported in this paper. When displaying dense graphs

(graphs with a large number of links) especially, these matrices tend to be very large. Incidence matrices can also be displayed using a node-link diagram, the so-called entity relationship diagram²⁶ with two different kinds of node, one representing nodes of the original graph, and the other representing links.

Comparing matrix-based and node-link representations

While several comparison studies have already been carried out in order to reveal differences in the readability of different representations for randomly generated graphs, few studies have compared visual representations of graphs with a semantic meaning.

Related literature

DSMs and adjacency matrices offer a ‘compact and visual representation’ of graphs⁵ and are mainly used among researchers and practitioners in engineering.^{16,18} However, traditional approaches of visually representing graphs in the form of node-link diagrams have several benefits over matrix-based visualisations. Node-link diagrams offer a higher degree of freedom when laying out graphs with nodes and links, making it possible to draw attention to certain aspects of the data set. This is reflected in a large body of literature on layouts for node-link diagrams (see di Battista *et al.*²⁴ for an extensive collection of layout algorithms). In this paper, the layout of node-link diagrams resembles additional semantic structure, such as, for example, the true geographical location of cities modelled in a road network.

Comparisons between different layouts have attracted attention from the information visualisation discipline. Purchase *et al.*^{27–29} analysed how graph layout aesthetics and layout algorithms influence the ability of users to assess attributes such as finding the shortest path between two components, or assessing the connectivity of two nodes. In another study,³⁰ user preferences for the layout of UML diagrams were elicited. Blythe *et al.*¹ examined how graph layouts affect the ability of users to identify subgroups in graphs describing social networks. A common characteristic of all these studies is that they compared different graph layouts, rather than comparing different representations (i.e. matrix-based against node-link diagrams) of graphs.

One study by Ghoniem *et al.*³¹ compared matrix-based representations with node-link diagrams and came to the conclusion that for most of the tasks on large and dense graphs (with the exception of finding a path between two nodes) participants performed better (meaning that participants were faster and more accurate) using a matrix representation rather than a node-link diagram. For small and sparse graphs, node-link diagrams were shown to behave better.

The graphs used in this study were purely simulated and had no further semantic meaning. Real-world data usually has a semantic structure. For example, in an engineering design context, the graph carries rich information about the process, product or organisation and is known, at least partly, to all stakeholders involved. Additionally, interviews with senior design practitioners indicate that, although matrix-based representations are very common in engineering design research, node-link diagrams are favoured. One leading engineer said: ‘*Let’s face it, a DSM is not a representation designers like using.*’ If this is so, the study carried out by Ghoniem *et al.*³¹ does not meet the needs of engineering design users, which are the target group for the visualisations in our case.

Research questions

Three different strategies for presenting connectivity models were introduced earlier. All three can visualise the same set of data, but present information very differently to the potential user and it is not clear which attributes of the data influence how users can gather information from each of them. Ghoniem *et al.*³¹ investigate this question, but they focus on random data sets. There is no evidence of how prior knowledge of the data set (experience) influences which representation is more suitable. This leads to the following research questions:

1. Which attributes of the connectivity model influence how well users can read relevant information from a DSM or a node-link representation?
2. Which representation (DSMs or node-link diagrams) is better suited for certain tasks involving reading information from connectivity models of real data sets?

3. How does familiarity with the data set influence users’ ability to retrieve information from a visual representation of a connectivity model?

Question 1 is answered by the experiment described in the next section, which investigates the influences of the data set on the readability of DSMs (the main representation used in an engineering design context). A discussion on which factors influence the readability of node-link diagrams is presented by Purchase,²⁷ where the effects of different layout aesthetics rather than effects of size and density measures on the readability of graphs was described. However, the experiments indicate that the density of a graph has a huge effect on its readability.

This discussion is further developed with experiment 1, which focuses on matrix-based representations rather than on node-link diagrams. Answers to questions 2 and 3 are derived from the second experiment, which compares users’ ability to extract relevant information about real data sets from DSMs and node-link diagrams. User performance in the experiments described later is assessed according to the standard variable measures of response time and error rate.

Experiment 1: influences on the readability of DSMs

Previously, different representations for visualising graphs were introduced. All these techniques have advantages and work well for certain kinds of graphs. In this section we will examine which attributes of the data have an influence on the readability of DSMs only. The findings presented here will be used to outline the test scenario for the second experiment.

Possible influencing factors

Related literature describing similar studies on readability of graphs^{27,31} and findings from user observation in several case studies indicate that the following factors influence the readability of DSMs:

Size The visual representation should be able to show data sets of different size. In the case of product modelling, a diesel engine has of the order of 50 main assemblies, but helicopters can have as many as 10,000 single components. However, the bigger the matrix, the longer it will take to find the corresponding row/column and the longer it will take to count the marks in a row/column. Additionally, the larger the matrix, the more difficult it is to read a row/column.

Density Complexity varies in different data sets. One measure of complexity is the density, the degree to which nodes in the graph are connected to each other.³² Highly connected graphs, such as the model of a diesel engine, need to be represented as well as very simple ones with much less connectivity between nodes. Higher complexity results in more marks in a row/column and can be measured by the average proportion of marks in a row.

Directionality The standard appearance of a DSM as established in the engineering design community⁵ and shown in Figure 2 has labels only on the vertical axis. Owing to this lack of symmetry, we would expect longer response times and higher error rates while assessing the number of outgoing links of a component, as the user first has to find the component in the row and then switch to the corresponding column.

Size and density are standard measures for the complexity of a graph and it is argued that the bigger the size and the higher the density, the more complex the visual representation.^{32,33} The directionality is a consequence of how DSMs are usually displayed, with labels on the vertical axis only rather than on both axes.

Experimental design

Participants For this experiment, 21 participants (13 males, 8 females) were recruited. All of them were either engineering Ph.D. students or professionals. Seven stated that they had previous experiences with DSMs (14 didn't) and 16 stated that they had an engineering design background.

Apparatus The test program was an online Java applet, so participants were able to conduct the experiment from remote locations. This program only supports the display of matrix-based representations of the graphs and no further interaction capabilities are included. Between the (timed) tasks, each user had the possibility to rest for as long as he or she wanted, before he or she decided to continue with the next task.

Thirty-two randomly generated graphs were included in the test program. Of these, 12 had a size of 10 nodes, 12 were of size 20 and 8 of size 40. Half of the matrices had a density of 10%, the other half had a density of 20%. The different sizes and densities reflect average values for these parameters from product models collected in case studies conducted by members of our group and resemble graphs of medium size. We used matrices without any further semantic structure, which is different from the second experiment reported later in this paper where real data sets were used. A training session at the beginning of the experiment allowed participants to get used to the use of the test program and the meaning of DSMs.

Design In the first part of the experiment, the users had to assess as to how many other components a particular component was connected using 12 DSMs of size 10. The second part consisted of 12 DSMs of size 20, the third of 8 DSMs of size 40. Half of them had a density of 10% and the other half a density of 20%. For each DSM, the user was asked to count either the number of outgoing links or the number of incoming links for a particular node. In the first case, the user had to count all the marks in a column (outgoing), in the latter one to count all the marks in a row (incoming). The same tasks were also used in the second experiment described below. For each task,

the time spent solving the task (response time) and the correctness of each task (error rate) were measured.

The following null-hypotheses were the basis for the first test:

H1. Size of the DSM has no influence on the user's ability to retrieve information from a DSM (response time and error rate).

H2. Density of the DSM has no influence on the user's ability to retrieve information from a DSM (response time and error rate).

H3. Directionality has no influence on the user's ability to retrieve information from a DSM (response time and error rate).

Results

An initial standard analysis of variance (ANOVA) revealed that all the factors (size, density and directionality) have significant impacts on the description of both dependent variables (response time and error rate). For each factor we used *t*-tests (under the assumption that response times can be modelled as normally distributed) and box plots³⁴ as a visual representation of the collected data to investigate whether the hypotheses stated earlier hold.

Size Figure 5 and Table 1 show the differences in the task response time for various sizes of the DSM. It can be seen that assessing direct connectivity of a DSM of size 40 was slower than reading DSMs of smaller sizes. Interestingly, users needed more time reading DSMs of size 10 than of size 20. This might be the effect of the order in which the test was carried out, DSMs of size 10 being the first group of DSMs to be read. It seemed that learning had a large effect in this case and that the differences between matrices of such small sizes are negligible. However, the response times for matrices of size 20 were significantly less than that for the matrices of size 40 ($P < 0.01$), so hypothesis *H1* can be rejected.

Size also had an effect on the error rate of the participants (see Table 1). Participants performed better with larger matrices; again this is most probably due to greater experience with the test procedure, or to the fact that the participants were more careful when reading large matrices, an explanation given by one participant after doing the experiment. These results contradict the assumption that users would take the same care for all sizes of matrix and for this reason the second experiment (which was based on the experiences learned doing this one), had a completely random order of tasks and different matrix sizes.

Density The null hypothesis *H2* that the density had no effect on the readability of adjacency matrices was rejected for matrices of size 40 ($df = 20$, $P = 0.008$). On the matrices of smaller size, the results were not significant. This shows that the density has an effect on

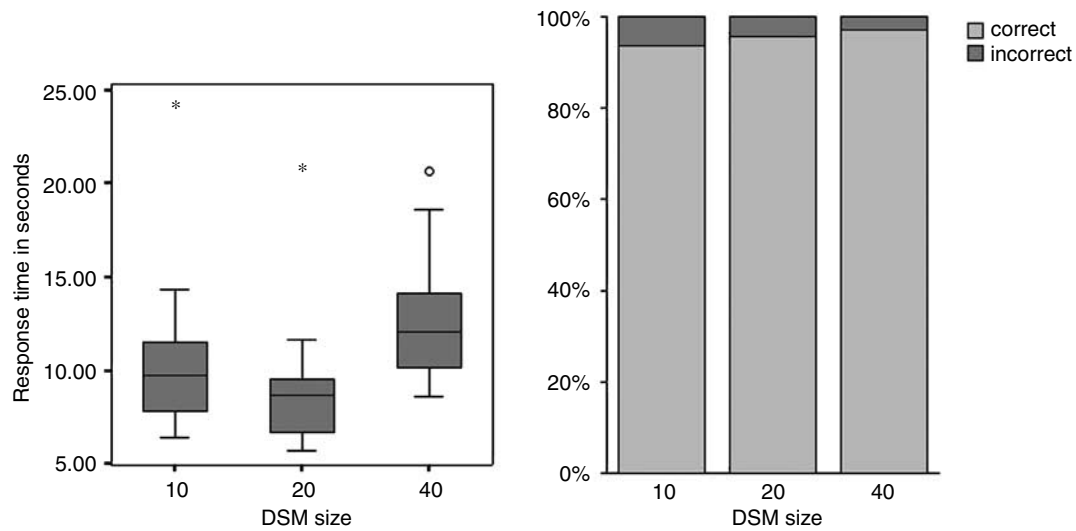


Figure 5 Effect of size of the DSM on response time and error rate.

Table 1 Effects of size and direct/indirect link assessment on error rate and response time

| Size of the DSM | Average error rate | Average response time in seconds |
|-----------------|--------------------|----------------------------------|
| 10 | 6.35% | 10.4 |
| 20 | 4.37% | 8.9 |
| 40 | 2.98% | 12.5 |

Table 2 Effects of density and directionality for 40 component DSMs

| Factor | Average response time in seconds | P (accept H_0) |
|---------------------|----------------------------------|---------------------|
| Density: 10% | 11.2 | 0.008 |
| Density: 20% | 13.9 | |
| Direction: outgoing | 14.1 | 0.008 |
| Direction: incoming | 11.0 | |

the readability of the DSM as the sizes of the matrices grow (see Table 2). However, no significant effects of the error rate could be detected, as the error rates were generally very low.

Directionality The effect of direction on user response times is only noticeable on matrices of size 40, where there is a significant difference between response times when assessing the number of outgoing links vs assessing the number of incoming links (see Table 2), so H_3 could be rejected for large matrices. Again, the differences in the smaller matrices showed no significant results. The effects on the error rate were also negligible.

Discussion

This experiment showed that the previously defined factors (size, density and directionality) all have significant effects on the readability of DSMs. However, it seems that the initial variability between subjects in time needed to read the question and understand it outweighed the effects of other influencing factors for small (10 components) and medium-sized (20 components) matrices.

Other factors in this variability could be different experience of users with DSMs, varying experiences with computer tools, various academic backgrounds, learning effects and disturbances while executing the test. However, for large matrices the results hold statistically. Nevertheless, it was shown that multiple factors influence the ability to read data from a DSM effectively.

Other graph measures, such as the distribution of node degrees, which might also have an influence on the readability of DSMs (and node-link diagrams) were not included in this experiment, as it seems from other studies and experience that the factors used for this experiments will have the highest impact on the readability. Further research considering such effects could also contribute to information visualisation research.

Another important result for DSMs indicates that the traditional layout of the DSM, with labels only on the vertical axis, has the disadvantage that assessing the number of outgoing links is more difficult than assessing the number of incoming links. Participants actually complained after the test that they often read the wrong column especially for large matrices. This leads us to suggest that the standard layout of DSMs can be improved by labelling both horizontal and vertical axes.

Experiment 2: readability of DSMs and node-link diagrams

Having previously identified the main factors in the readability of DSMs, the following experiment aims to

answer the question: ‘Which representations of graphs, DSMs or node-link diagrams, are more suitable for reading the underlying graph structure?’ The findings here are expected to correlate to the results described in Ghoniem *et al.*³¹ The main differences between this study and the study described in Ghoniem *et al.*³¹ are:

- The graphs used this paper are not random: data sets with a semantic underlying structure are being used;
- A different set of tasks is being used;
- Directed graphs are used.

Task descriptions

Some of the tasks included in this experiment have been used in other studies^{27,31} and cover a wide range of the cases for which a connectivity model might be used. They are simple information retrieval tasks that do not require deep analyses of the graph and reflect what information designers might want to gather from such a model. For example, the shortest path between two components in a product model can give a hint of how likely a change is to propagate from one of the components to the other.¹⁵

Selecting a node: Finding and selecting one specific node is important when interacting with linkage models, especially when people want to see which nodes are connected to one particular node. Using DSMs, this task is expected to be quite easy, as only the column showing the names of the nodes has to be read. The task can even be simplified by providing the nodes in a specific order. If the reader has no experience with the underlying model, alphabetical ordering might be a good choice. Other orderings, such as by importance of a component, are equally valid. Finding a node in a node-link representation requires a search of the entire two-dimensional space. The many ways of laying out a node-link diagram offer many possibilities, such as laying it out in a way that reflects the ‘real’ layout of the underlying structure. The data set of the German motorway system described later is such an example, where the layout of the node-link diagram reflects approximately the geographical location of German cities. If a user has this geographical knowledge, it should be easier to find nodes as the node representing a city will appear roughly on its true geographical position. This task has its equivalent in the study undertaken by Ghoniem *et al.*³¹ in the ‘findnode’ task.

Selecting a link: Finding and selecting a link between two nodes essentially means first locating the two nodes in both representations and then finding the connecting link. In DSMs, this task can be broken down into finding the tail of the link (which is essentially a search for the node as described earlier, but in the column header) and then searching for the head of the link (which is again just a search for a node). Then the corresponding cell in the matrix has to be looked up in order to decide whether there is a link or not. Finding a link in a node-link diagram means localising the tail node of the link. The connected components are then only those nodes to which an outgoing connection goes from the node in

question. Connected nodes are clearly distinguishable from other nodes in a node-link diagram representation. This task of finding a link was also analysed by Ghoniem *et al.*³¹ as ‘findlink’.

Counting the number of incoming/outgoing links: Counting the number of incoming and outgoing links for a node can be crucial when one wants to cluster the nodes on the basis of their impact on other nodes (the number of incoming and outgoing links). Counting the number of outgoing and incoming links in a matrix involves first finding the corresponding node and then counting all the marks in the corresponding row (incoming links) or column (outgoing links). In node-link diagrams, again, the node has to be found and then the number of arrows going out or coming in has to be counted. This can be problematic if overlapping edges prevent the user from distinguishing incoming from outgoing.

Counting the number of common neighbours: The number of common neighbours between two nodes can be an indicator of how closely two nodes are related. Counting the number of common neighbours in an adjacency matrix can be a difficult task. Firstly, the two relevant nodes have to be found. Then each column and row must be checked to see if both nodes are connected to another node. The fact that the underlying graphs are directional adds another level of difficulty to this problem when using matrices for this task. For example, in the case that $A \rightarrow X \rightarrow B$, the mark for the common neighbour is once in the column (link coming from A) and in the row (link going to B). Using a node-link diagram, this task is expected to be easier. After having found the two relevant nodes, one only has to search for other nodes that are connected to these two.

Finding the length of the shortest path: Finding the shortest path with both representations means following paths from one component to another. Path following in node-link diagrams is relatively straightforward and just involves iteratively following outgoing links from a node. In DSMs, path following is expected to be much harder, as one has to repeatedly switch between different columns and rows. Based on all possible paths, the user then has to decide which one is the shortest. All participants found this task very difficult to perform with adjacency matrices.

Calculating the node/edge connectivity: The node connectivity between two nodes is defined as the minimal number of nodes that have to be removed from a graph in order to separate these two nodes. The edge connectivity is the number of links that have to be removed. We decided not to include these tasks in the experiment because we found that calculating the connectivity with an adjacency matrix was an almost impossible task, and even with a node-link representation only simple examples could be solved properly.²⁷

Experimental design

Participants For this experiment, 16 participants, all engineering graduate students and professionals, were

recruited. Of these participants, six were German nationals. This allowed us to analyse how familiarity with one of the data sets influences user performance when reading information from visual representations of graphs. Of the 16 participants, 5 were females, 11 were males. Of these, 11 stated that they had an engineering design background, five had a different background. All were familiar with how to read DSMs, as all of them had participated in the first experiment.

Apparatus The test was executed on an AMD Athlon 1700 XP portable computer with 512MB RAM and a high-resolution display of 1600*1200 pixel. The test program was implemented in Java and relied on the freely available libraries ‘graphviz’³⁵ and ‘grappa’³⁶ for the display of the graphs. The interactive capabilities of the program for node-link diagrams allowed users to select nodes and links. The matrix-based representation allowed the selection of rows, columns and cells. These operations were equivalent to the selection options in the node-link diagrams. After each task, the user had time to rest, before continuing with the next one.

Into the test program, three real-world data sets were incorporated.

- The *Heart model* (see Figure 6) describes the physical layout of the human heart. It consists of eight ‘components’ and was only used in the training phase.
- The *German road model* describes direct motorway connections between major German cities. The underlying graph has 50 nodes (big cities in Germany) and 142 links, resulting in a density of 5.8%. The order of the components in the DSM is alphabetical; the graph layout is determined by the geographical location of the city and the ‘neato’ program³⁵ to remove node overlaps. This data set is symmetrical.
- The *VistaD model* is the connectivity model of a diesel engine and is used to predict change propagation. Most of the participants of the experiment were familiar with this model. It consists of 22 components as nodes with 126 links between them, making a density of

| Heart | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----------------|---|---|---|---|---|---|---|---|---|
| Aortic Valve | 1 | 1 | | | X | | | | X |
| Left Atrium | 2 | | 2 | X | | X | | X | |
| Left Ventricle | 3 | | X | 3 | X | | | X | |
| Mitral Valve | 4 | X | | X | 4 | | | | X |
| Pulmonary Valve | 5 | | X | | | 5 | | | |
| Right Atrium | 6 | | | | | | 6 | X | X |
| Right Ventricle | 7 | | X | X | | | X | 7 | |
| Tricuspid Valve | 8 | X | | | X | | X | | 8 |

Figure 6 DSM representation of a human heart model with eight ‘components’.

27.3%. The node-link diagram was laid out using the ‘neato’ program (see Gansner and North³⁵ and Figure 7); the order in the nodes of the DSM was alphabetical. For the node-link layout, the ‘neato’ program was used to avoid links overlapping with nodes, which resulted in not-straight links. This data set is not symmetrical.

Design The layout of the test was similar to that described by Ghoniem *et al.*,³¹ in respect to task description, completion time limits (one min maximum) and interactive capabilities of the software (selection of nodes and links, but no ‘mouseover’ highlighting). The participants were asked to perform each task as quickly and as accurately as possible. The test program started with the simple model of the human heart (Figure 6). All the different tasks were explained with both representations (matrices and node-link diagrams), and the users were able to practice the interactive capabilities of the test program. The following test session consisted of 48 tasks, that is, 24 different tasks (6 tasks × 2 models × 2 representations) carried out twice. That means that each task had to be completed eight times (with different models and representations). The individual questions were created randomly (e.g. which node had to be selected) as was the order of the tasks (there was no assigned order in which the tasks had to be executed) and the given representation (matrix-based or node-link).

The experiment covered the following tasks; the initial state of the selection of nodes is given in brackets:

- Selecting a node (no nodes are selected initially);
- Selecting a link (no initial selection);
- Counting the number of incoming links to one node (the node in the node-link diagram and the equivalent row/column in the matrix are selected);
- Counting the number of outgoing links from one node (the node in the node-link diagram and the equivalent row/column in the matrix are selected);
- Counting the number of common neighbours between two nodes (both nodes are initially selected);
- Finding the length of the shortest path between two nodes (both nodes are initially selected).

The task of assessing connectivity between two nodes was not included in the test. An initial test trial suggested that it was almost impossible to represent connectivity visually in an adjacency matrix and, in this case, a rather thorough analysis of the data would be necessary. In order to reduce the total time spent for the experiment (which was even without this task, about 50 min) we decided to omit this task.

The expected results should provide the following:

- Verification of the results described in Ghoniem *et al.*,³¹ The layout of this experiment is similar to the test described in this paper.
- A comparison of the readability of DSMs and node-link diagrams for presenting semantic, directed graph data

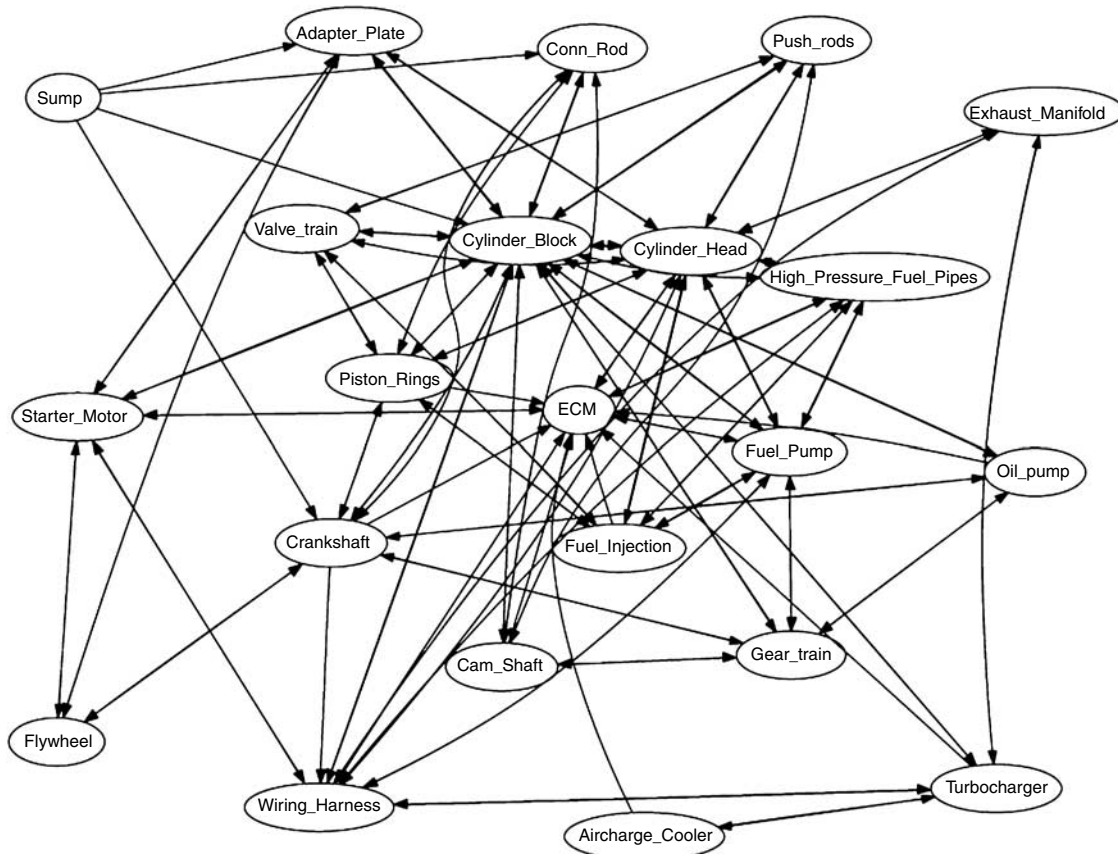


Figure 7 Node-link diagram of the VistaD data set, laid out using the 'neato' spring algorithm.

(in contrast to the simulated undirected graph approach taken by Ghoniem *et al.*³¹).

The following two null-hypotheses are the basis for this test:

H4. User performance (response time and error-rate) is independent of the used visual representation of the connectivity model for each of the tasks described earlier.

H5. User performance (response time and error-rate) is independent of prior knowledge of the connectivity model for visual representations.

Results

In this section we present the results of the experiment described above. An initial univariate ANOVA of the correct answers showed that the following factors have a significant influence on the response time (the factors are ordered according to their influence):

- The data set (German road model or VistaD model);
- The task (whether, for example, the user was searching for a node or the length of the shortest path);
- The visual representation (whether a DSM or a node-link diagram was used);

- Whether the test-taker had prior semantic knowledge (in this case: was familiar with German geography).

That the task and the data set have an influence on the response time was clear, as the data sets differed in size and the tasks varied in difficulty level. Thus, two factors remained: the type of visual representation, and whether the participant was familiar with German geography, that is, had prior semantic knowledge of the model. The remainder of this section will analyse the effects of these two factors on the participants' performances. The data collected on age and gender showed no influence on the response times and error rates.

Another ANOVA analysis showed that the only variables that have an effect on the error rates of participants' answers were the task type and the visual representation; thus, there was no influence of the dataset and whether the test-taker had prior semantic knowledge.

In order to judge the effects of the visual representation, we used quantitative (*t*-tests to compare response times of correct answers) and qualitative graphical methods (stacked bar charts to compare correctness and error rates, and box plots to compare response times) for each single combination of the variables *data set* and *task*. As a null-hypothesis we assumed that the visual representation has no influence on the response time and

error rate (H4). The results of all tests are summarized in Table 3.

Selecting a Node For both data sets (the *Road* data and the *VistaD* data), no significant differences could be detected between the visual representations. The error-rates are very low (see Figure 8). The box plot in Figure 8 suggests that the response times using DSMs are marginally smaller than using node-link diagrams. The experimental results by Ghoniem *et al.*³¹ (using their linear regression results) are different to these findings for both data sets (expected response times: street, matrix: 6.02, street, node-link: 11.12, vista, matrix: 6.02, vista, node-link: 6.09), as they would expect slightly smaller response times for both data sets using matrix-based representations.

Selecting a link For the *Road* data set, there was a significant difference between the response times of

participants using graphs or DSMs, resulting in significantly faster response for node-link diagrams ($P < 0.05$). For the *VistaD* data set, no significant differences could be detected. The error rates, again, show hardly any differences (see Figure 9). The results for these data sets correspond to the findings of Ghoniem *et al.*³¹ (expected response times: street, matrix: 9.73, street, node-link: 4.42, vista, matrix: 6.02, vista, node-link: 9.05), where node-link diagrams are expected to outperform matrix-based representations for such graphs.

Counting the number of outgoing links Again, for the *Road* data set, the response times using node-link diagrams were significantly smaller than using DSMs ($P < 0.05$). However, no significant differences between the two factors were detected for the *VistaD* data set. The box plots in Figure 10 confirm these findings. The error rates were again negligible (see Figure 10), but generally higher for the *VistaD* data set.

Table 3 Summary of experiment 2

| Task | Data set | Mean response time in seconds | | Error rate | | Significance in response time ($P < 0.05$) |
|---------------------------|----------|-------------------------------|-------------|------------|-----------------|--|
| | | DSM | N-L diagram | DSM (%) | N-L diagram (%) | |
| Find node | Road | 9.5 | 11.4 | 3 | 3 | |
| | VistaD | 7.1 | 8.1 | 0 | 9 | |
| Find link | Road | 27.5 | 17.3 | 22 | 6 | + |
| | VistaD | 22.2 | 20.2 | 9 | 3 | |
| Count outgoing links | Road | 15.8 | 8.6 | 0 | 0 | + |
| | VistaD | 15.3 | 16.5 | 6 | 13 | |
| Count incoming links | Road | 13.3 | 8.1 | 3 | 0 | + |
| | VistaD | 12.6 | 14.4 | 13 | 16 | |
| Count common neighbours | Road | 29.4 | 15.7 | 9 | 0 | + |
| | VistaD | 31.9 | 30.3 | 47 | 22 | |
| Find shortest path length | Road | 39.5 | 22.9 | 84 | 9 | + |
| | VistaD | 30.8 | 20.3 | 41 | 19 | + |

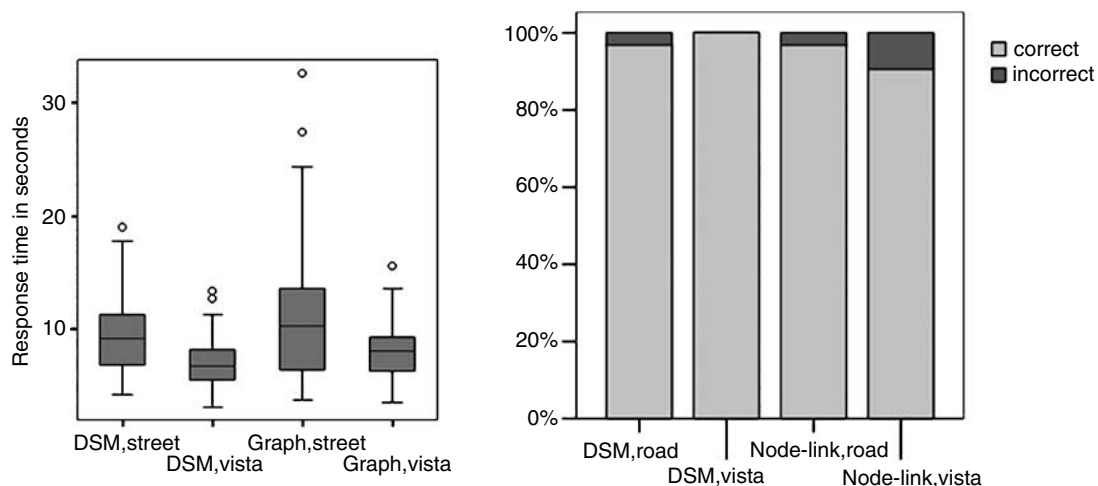


Figure 8 Completion times and error rates for finding a node.

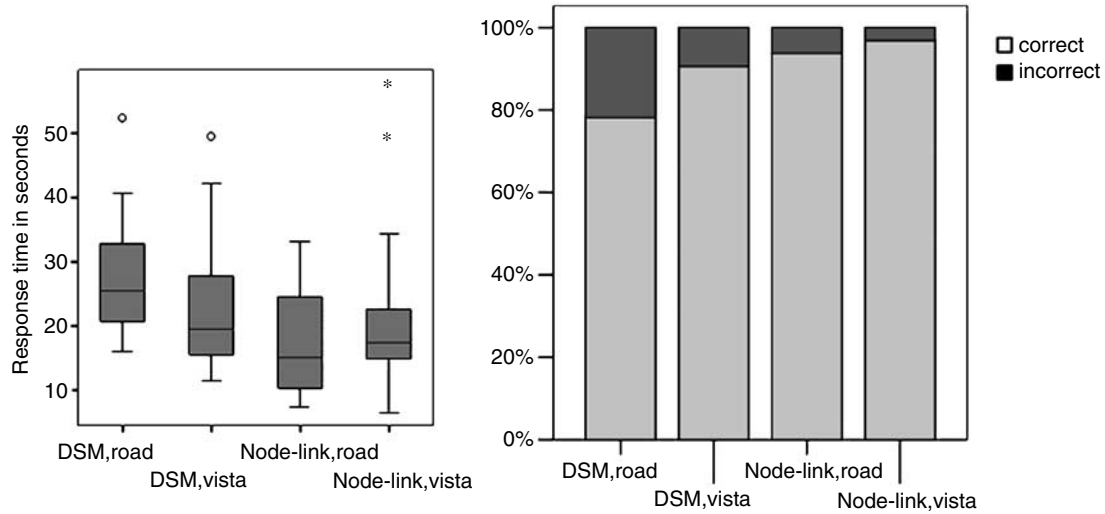


Figure 9 Completion times and error rates for finding a link in the graph.

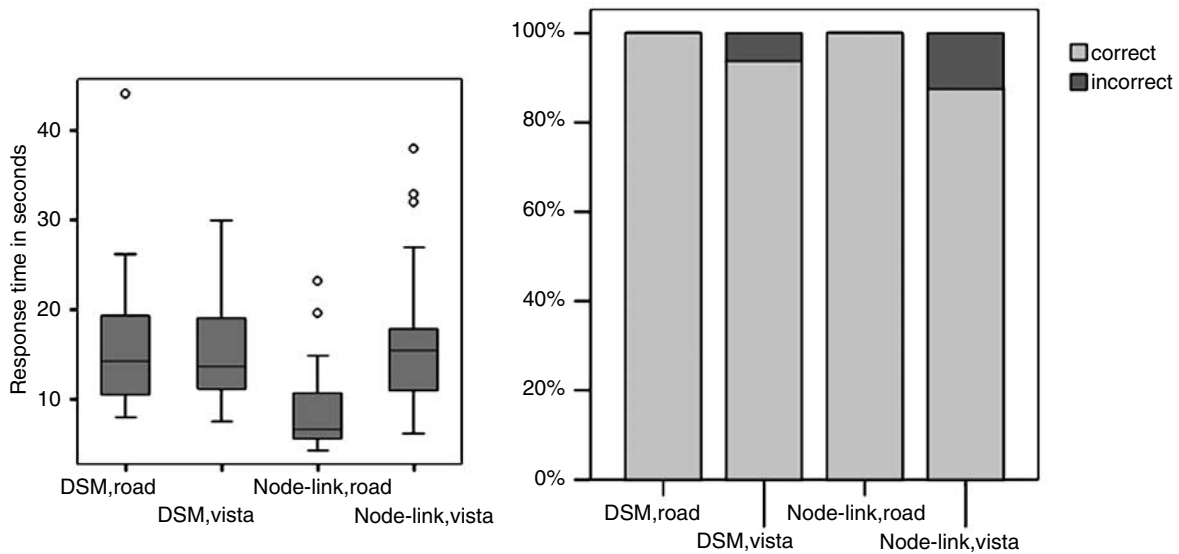


Figure 10 Completion times and error rates for finding the number of outgoing links.

Counting the number of incoming links The results for the number of incoming links to a node are similar to the ones observed above. There is a significant difference for the *Road* dataset ($P < 0.05$), but no statistical differences between node-link diagrams and DSMs for the *VistaD* data set. The error rates were again negligible (see Figure 11) and higher for the *VistaD* data set.

Counting the number of common neighbours Here, participants using node-link diagrams were again faster when assessing the *Road* data set ($P < 0.05$), while there were no significant differences for the *VistaD* data set (although response times were shorter with node-link representations). However, for this task, the error rates were quite high (see Figure 12), especially for the *VistaD*

data set, particularly when using DSMs, where approximately 50% of the answers were wrong.

Finding the length of the shortest path This task was completed significantly faster using node-link diagrams for both data sets ($P < 0.05$ for both data sets). However, the error rates, especially for the DSM representation, were very high (84% answered wrongly in using DSMs for the *Road* data set and 41% using DSMs for the *VistaD* data set, see Figure 13). This is significantly higher than the rate for those using node-link representations.

Effect of experience The initial ANOVA showed that participants of German nationality (who were familiar with German geography) performed the tasks faster than

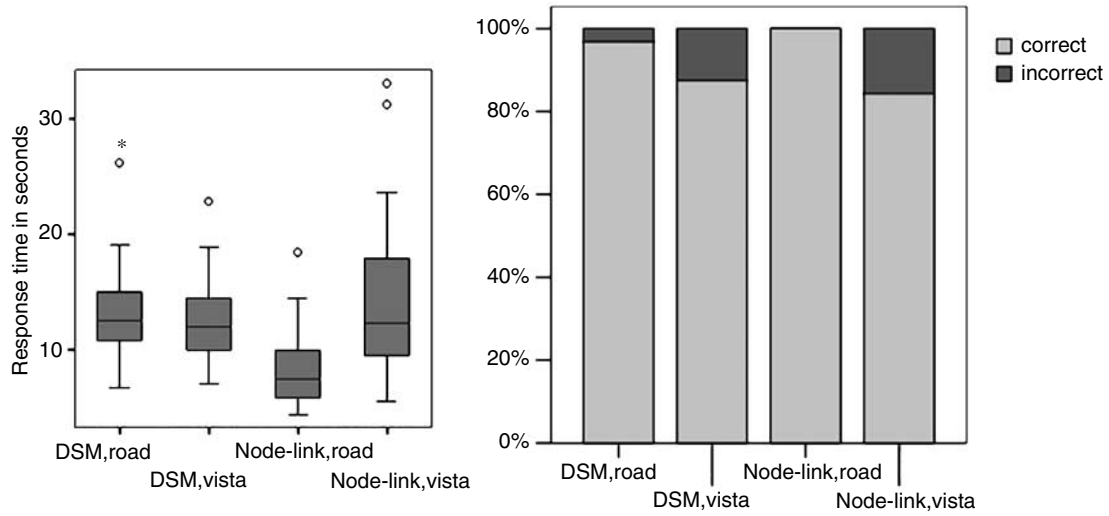


Figure 11 Completion times and error rates for finding number of incoming links.

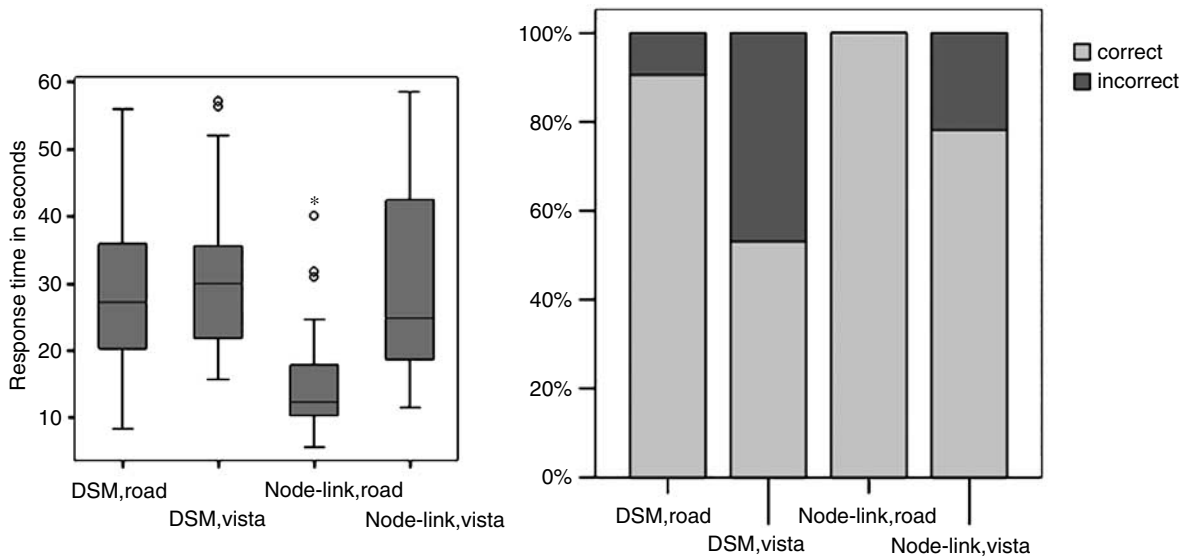


Figure 12 Completion times and error rates for finding common neighbours of two components.

those participants of other nationalities, who were expected to be less familiar with the locations of German cities. More thorough analyses showed that this effect is only observable for the *Road* data set (significantly smaller response times, $P < 0.05$), where participants of German nationality had the advantage of being familiar with both city names and locations. We were also interested in whether the representation (DSM or node-link diagrams) has a different effect on these two groups of people with different prior knowledge about the data set (German nationals vs non-German nationals). However, a univariate ANOVA showed no significant difference.

Discussion

It can be seen from Table 3 that for the *Road* data set, most of the tasks were performed faster using the node-link representation, while for the *VistaD* data set only the task of finding the shortest path was performed significantly faster with node-link diagrams. Thus, the results of this test support the findings of Ghoniem *et al.*,³¹ which showed that matrix-based representations were better suited for most of the tasks on large and dense graphs, but that node-link diagrams were more suitable for small and sparse graphs (as in the experiment reported here). An exception to this rule respects the task of finding paths between two nodes in a graph. It was shown in both

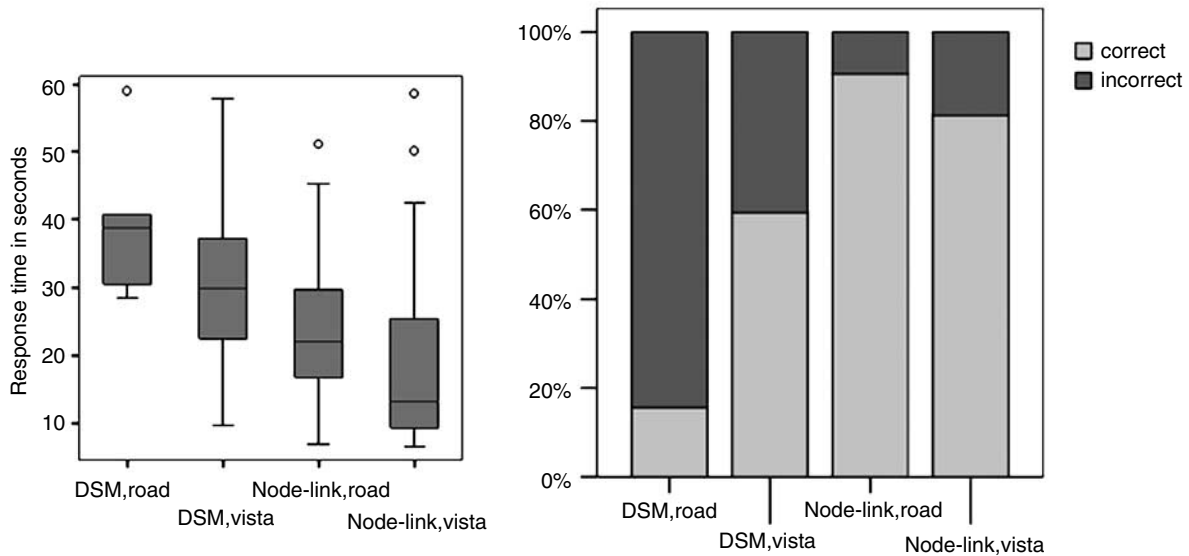


Figure 13 Completion times and error rates for finding the shortest path between two components.

studies that node-link representations outperform matrix-based displays in this task. It seems to hold that, for graph data with an underlying semantic context, and for sparse (or almost planar) graphs, node-link displays are the more intuitive representation (in respect to error-rates and response time).

Another finding was that error rates in particular tended to be higher for the *VistaD* data set. This is suspected to be due to the higher density of the data and corresponds to the findings of the first experiment. These errors can be explained by reference to overlapping links in the node-link display, and to the high number of marks in the matrix-based representation.

Additionally, this experiment showed that experience with a data set has a high impact on the user's performance. It also does not seem to be influenced by the graphical representation although the node-link representation includes much more information on geographical locations than adjacency matrices do, although adjacency matrices incorporate alphabetical ordering which was thought to ease node lookup for both participant groups, those who were familiar with German city location and those who were not. We would have expected that experience would have a greater effect on users of the node-link representation, but this is not shown by the data.

The implications of this experiment for representing graphs, or for any model that uses underlying graph structures, such as engineering change models described earlier, are that the appropriateness of a particular representation depends highly on the nature of the graph and of the task in question. For finding indirect links between nodes, a task necessary in the particular case of change propagation, node-link representations seem to be the best choice. For certain tasks on large or dense graphs, such as identifying highly connected

nodes, matrix-based representations can be advantageous.

The use of multiple views (in this case: the use of both matrix-based representations and node-link diagrams) to visualise graphs can be the solution for this problem. Having both representations available at the same time would highly benefit users as depending on the task it would be possible to choose the appropriate form of visualisation.¹⁴

Conclusions

In this paper we presented the results of two user-studies on the readability of DSMs and node-link diagrams. The first experiment was aimed at identifying the main factors that have an influence on the readability of DSMs, and clearly showed that the size and density of the underlying graph structure significantly influence both response times and error rates of participants. The second experiment compared how users read information from directed graphs with real-world semantics using matrix-based representations and node-link diagrams. It was shown that experience and prior knowledge of a network have a great effect on how well users can read information from the visual representation of a graph. As shown in other studies, these experiments also confirmed that node-link diagrams are better suited for reading information from small and sparse graphs and when assessing indirect paths between two nodes.

Another conclusion is that the most appropriate choice of representation depends on the detailed properties of the connectivity model and the specific task that needs to be carried out. In another study,³⁷ which focused solely on connectivity model building, we came to the same conclusion: depending on the model, and even on personal preference, either representation can be advantageous and leads us to propose a multiple view strategy to

properly support design engineers. To reflect this, the current implementation of the CPM tool¹⁴ – which uses connectivity models to predict and visualise change propagation in order to support designers in industry – offers both node-link diagrams and DSMs as alternative ways to visualise connectivity models.

Further research will look into how different representations afford more complex tasks. Especially in product

modelling, the search for and visualisation of clusters, is an important task and both representations introduced in this paper support the visualisation of such patterns in the product model.

Acknowledgments

This research is funded by EPSRC. We thank Tomás Flanagan and Felix Ballestrem for their contributions to this paper.

References

- 1 Blythe J, McGrath C, Krackhardt D. The effect of graph layout on inference from social network data. *Symposium on Graph Drawing 1995* (Passau, Germany).
- 2 Johnson JH. Some structures and notation of Q-analysis. *Environment and Planning B* 1981; **8**: 73–86.
- 3 Newman MEJ. The spread of epidemic disease on networks. *Physical Review E* 2002; **66**. <http://pre.aps.org/> (accessed 18 March 2006).
- 4 Jarratt T, Eckert CM, Clarkson PJ. Development of a product model to support engineering change management. *Proceedings of Tools and Methods of Competitive Engineering (TCME)*. 2004 (Lausanne, Switzerland), 331–342.
- 5 Browning TR. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering Management* 2001; **48**: 292–306.
- 6 Browning TR. Process integration using the design structure matrix. *Systems Engineering* 2002; **5**: 180–193.
- 7 Dodge M, Kitchin R. *Atlas of Cyberspace*. Pearson Education: Edinburgh, 2001.
- 8 Bucciarelli LL. *Designing Engineers*. MIT Press: Cambridge, 1996.
- 9 Malcolm DG, Roseboom JH, Clark CE, Fazar W. Applications of a technique for research and development program evaluation (PERT). *Operations Research* 1959; **7**: 646–669.
- 10 Eckert CM, Keller R, Earl C, Clarkson PJ. *Supporting change processes in design: complexity, prediction and reliability*, Technical Report CUED/C-EDC/TR136, University of Cambridge: Cambridge, 2005.
- 11 Simon HA. *The Sciences of the Artificial*. MIT Press: Cambridge, MA, 1998.
- 12 Clarkson PJ, Hamilton JR. ‘Signposting’, a parameter-driven task-based model of the design process. *Research in Engineering Design* 2000; **12**: 18–38.
- 13 Sosa ME, Eppinger SD, Rowles CM. Identifying modular and integrative systems and their impact on design team interactions. *ASME Journal of Mechanical Design* 2003; **125**: 240–252.
- 14 Keller R, Eckert CM, Clarkson PJ. Multiple views to support engineering change management for complex products. *Proceedings of the Third International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV '05)*, IEEE Computer Society: London, UK, 2005; 33–41.
- 15 Keller R, Eger T, Eckert CM, Clarkson PJ. Visualising change propagation. *Proceedings of the 15th International Conference on Engineering Design (ICED '05)* (Melbourne, Australia), 2005.
- 16 Clarkson PJ, Simons C, Eckert CM. Predicting change propagation in complex design. *ASME Journal of Mechanical Design* 2004; **126**: 765–797.
- 17 INCOSE, *Systems Engineering Handbook* 2004. <http://www.inco-se.org/ProductsPubs/products/sehandbook.aspx> (accessed 18 March 2006).
- 18 Steward DV. The design structure system: a method for managing the design of complex systems. *IEEE Transactions on Engineering Management* 1981; **28**: 71–74.
- 19 Warfield JN. Binary matrices in system modeling. *IEEE Transactions on Systems, Man, and Cybernetics* 1973; **3**: 441–449.
- 20 Eppinger SD, Whitney DE, Smith RP, Gebala DA. A model-based method for organizing tasks in product development. *Research in Engineering Design* 1994; **6**: 1–13.
- 21 Problematics [WWW document] <http://www.problematics.com/> (accessed 7 November 2005).
- 22 Siirtola H, Mäkinen E. Constructing and reconstructing the reorderable matrix. *Information Visualization* 2005; **4**: 32–48.
- 23 Card SK. User perceptual mechanisms in the search of computer command menus. *Proceedings of the 1982 Conference On Human Factors In Computing Systems* (Gaithersburg, Maryland, US), 1981, 190–196.
- 24 Di Battista G, Eades P, Tamassia R, Tollis IG. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry: Theory and Applications* 1994; **4**: 175–198.
- 25 Harel D. On visual formalisms. *Communications of the ACM* 1988; **31**: 514–530.
- 26 Chen P. The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems* 1976; **1**: 9–36.
- 27 Purchase HC. Effective information visualisation: a study of graph drawing aesthetics and algorithms. *Interacting with Computers* 2000; **13**: 147–162.
- 28 Purchase HC, Cohen RF, James MI. An experimental study of the basis for graph drawing algorithms. *Journal of Experimental Algorithmics* 1997; **2**. <http://www.jea.acm.org/> (accessed 18 March 2006).
- 29 Purchase HC. Performance of layout algorithms: comprehension, not computation. *Journal of Languages and Computing* 1997; **9**: 647–657.
- 30 Purchase HC, Allder JA, Carrington D. Graph layout aesthetics in UML diagrams: user preferences. *Journal of Graph Algorithms and Applications* 2002; **6**: 255–279.
- 31 Ghoniem M, Fekete J-D, Castagliola P. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization* 2005; **4**: 114–135.
- 32 Summers JD, Shah JJ. Developing measures of complexity for engineering design. *Proceedings of ASME-DETC Design Engineering Technical Conferences, Design Theory and Methodology Conference (DTM)*: (Chicago, Illinois, USA), 2003.
- 33 Earl C, Johnson J, Eckert CM. Complexity. In: Clarkson PJ and Eckert CM (Eds). *Design Process Improvement*. Springer Verlag: London, 2005.
- 34 Chambers J, Cleveland W, Kleiner B, Tukey P. *Graphical Methods for Data Analysis*. Wadsworth: Belmont, CA, USA, 1983.
- 35 Gansner E, North S. An open graph visualization system and its applications to software engineering. *Software: Practice and Experience* 2000; **30**: 1203–1233.
- 36 AT&T-Labs. <http://www.research.att.com/~john/Grappa/> (accessed 9 June 2005).
- 37 Keller R, Eckert CM, Clarkson PJ. Building connectivity models in design: representations and tools to support cognitive preferences. *Second International Conference on Design Computing and Cognition (DCC'06)* 2006, (accepted) (Eindhoven, Netherlands).