BUILDING CONNECTIVITY MODELS IN DESIGN: REPRESENTATIONS AND TOOLS TO SUPPORT COGNITIVE PREFERENCES

RENÉ KELLER, CLAUDIA M ECKERT AND P JOHN CLARKSON
University of Cambridge, United Kingdom

Abstract. Reasoning about the connectivity within a product is an integral part of many core design activities. Due to the complexity of a product and the sheer number of potential links, designers often overlook vital connections resulting in problems later in the process, leading to errors or costly rework. Product connectivity models, which are essentially graphs, are a promising approach for capturing these links between components in a complex product. The primary visual representation used to create such connectivity models is the Design Structure Matrix (DSM). However, other representations of graphs may be superior for creating connectivity models of products. This paper presents node-link displays as equally valid representations for product connectivity models and reports on an experimental study that investigates whether DSMs or node-link diagrams are more suitable for building such models.

1. Introduction

Models of products and processes play an integral part in reasoning about engineering design. The final product does not exist during the design process in a physical form until a prototype can be made. Even when a physical reference design does exist, designers don't always have access to it, for example when it is too large (Boeing 747 jet) or too small (microchip); the behaviour and function of an object is not immediately visible from its physical form.

Process models and plans, on the other hand, are the only representation of the design process. However, they are notoriously inaccurate and ambiguous regardless of which of their three potential roles they play at any one time: as monitoring aids; prescriptions of the process; or records of the process (Eckert and Clarkson 2003). For very complex products or processes, it is impossible for one person to remember and recall all necessary (detailed) information. This makes models of products and

J.S. Gero (ed.), Design Computing and Cognition '06, 41–60.

processes vital as externalisations of collectively-held knowledge that enable individuals to reason about that knowledge and for teams to communicate. Thus it is very important for the success of any design project that every member of the design team has a consistent understanding of the models they use (Henderson 1999).

However, each product or process has multiple models, which might overlap, and which are subject to different interpretations depending on the person and the time in the design process. Some models are used within one group, with a joined understanding; some are boundary objects and serve as communication aids between groups. While reading models is, to some extent, an acquired skill, these boundary objects need to be easily understandable to be accessible. This paper talks about boundary type objects (Star 1989) that bridge the gap between groups with different expertise while expressing very complex information. Thus, these models must be fairly abstract to capture complex information in a concise form, but also fairly rich to provide a meaningful range of connections.

As models are often the only reference to the process or the product, understanding and interacting with these models is an integral part of the design activity. Through design processes, models can be used in very different ways and have different functions.

- Cognition: Human beings are severely limited in the complexity of the things they can keep in mind at one time (Cowan 2001). As Simon (1998) pointed out, designed complex systems are organized as 'nearly decomposable' hierarchical structures with components whose interactions are much simpler than their internal workings, so that it is feasible to understand each element in terms of its behaviour and the interactions of its subcomponents. For large and complex products and processes, models can be a means of abstracting knowledge to break it down into an understandable and manageable piece of information of medium size (Zeitz 1997). An alternative, and complementary, approach to reducing the complexity of the thinking designers need to do is to consider different *aspects* of a design separately. For instance, Hoover et al. (1991) observed designers employing different abstractions and corresponding graphic representations to perform analyses of different aspects of their designs. By "blending out" unnecessary information, such as using fisheye approaches (Furnas 1986) and grouping similar things together, they can allow effective information retrieval even for complex products and processes, and form the basis for analyses of certain aspects of the product or process.
- Communication: One important application of models is to communicate information to other stakeholders. Designers carry models to other people's desks in order to clarify certain aspects of a design, and bring models to meetings. Especially when people have different backgrounds, hence a different knowledge of the design, which is

inevitable the case in the design of very complex products (Jarratt et al. 2004b), suitable models can be seen as a "least common denominator" that is understood by the whole design team. Models are also a way of communicating ideas about the design to people outside the design team, or to managers who only have a very broad overview rather than a detailed technical understanding of the process or product.

- Recording: Models can be used to store information for further designs. Models of previous design processes and products, for instance, can be used as a starting point for new designs and are also a way to train novice designers. However, when using models there is always a trade-off between the amount of effort it requires to create and update the model and the expected value it can provide.

Models (not only process and product models) also have several different possible visual representations, e.g. connectivity models can be represented for instance as node-edge diagrams or matrices. The chosen representation can have a huge impact on how people interact with and interpret information provided. Each type of presentation has a structure and only "affords" the representation of certain type of information, as "*there is no one representation that allows detailed considerations of all possible concerns*" (Gero and Reffat 2001). Larkin and Simon (1987) highlight the importance of a proper visual representation and state that "*whether a diagram (or any other representation) is worth 10,000 words depends on what productions are available for searching the data structure, for recognizing relevant information and for drawing inferences from that information.*"

Thus, a proper representation must allow the user to perform the desired task on the underlying model easily. Many representations of a product or a process must serve very different purposes and should support different designers in all stages of the design process. They should provide a means to communicate about the underlying thing (the product or process). The representations should also support users when building the model in the first place, thus reducing the effort required for creating and maintaining the model. However, this paper will show that how easily information can be extracted from a representation, depends highly on the knowledge of the person interacting with the model and their personal preference.

This paper discusses which representation of a particular kind of model (connectivity model) is generally better suited for the particular task of building such a model. In order to achieve this, experiments were undertaken that compared the two most common representations for connectivity models, namely node-link diagrams and matrix-based representations, in the context of product connectivity models. This research completes the contribution on representations of connectivity models, where in a second study (Keller et al. 2005a), the differences between representations for information retrieval tasks were analysed.

The remainder of this paper includes:

- A description of current connectivity models of products and processes and their applications in industry (Sections 2);
- An introduction and comparison of Design Structure Matrices and node-link diagrams used to represent product connectivity models of complex products (Section 3);
- Presentation and discussion of an experimental user study that shows how the representation (matrix-based or node-link) influences connectivity model-building (Section 4).

Understanding the implications of such a comparison may be very beneficial to the design process as well as for other disciplines where such connectivity models are common (e.g. social networks in social psychology).

2. Connectivity Models

As in any complex, almost decomposable system, linkages between parts are a common structural feature in both processes (dependencies between tasks) and products (e.g. components that have spatial relations). Connectivity models capture those connections and can be used by designers and project managers to model products as well as processes and projects. For example, task dependencies in process models can indicate the order of tasks in a process. The main underlying structure is a graph, with nodes and links modelling interactions between nodes. A famous process model of this kind is the PERT method (Malcolm et al. 1959), which is used to find critical paths in the tasks that constitute a project. Design Structure Matrices (DSMs) are a method to represent and restructure dependencies of the binary representation of a matrix-based representation. For example, DSMs are used to improve the design process by reordering the task sequence (Browning 2001). Signposting models of design processes (Clarkson and Hamilton 2000) is a generalization of the relational model of task-dependencies, where dependencies are driven by the state of parameter descriptions.

2.1. PRODUCT CONNECTIVITY MODELS

The range of general product models includes detailed CAD models of geometric and electrical properties, functional models, component breakdowns of products (such as the bill of material), and prototypes and sketches (Henderson 1999). All represent information about the product at different levels of abstraction and from different viewpoints. CAD models, which use a very concrete representation of the spatial relations of components of a product, are widely used in industry (see (Henderson 1999) for the description of case studies revealing the role of CAD models in engineering companies) and play an integral role in the design of new products. More abstract product models like functional models (Pahl and Beitz 1996) are used in conceptual design. However, currently very few models indicate relations between parts and combine these different aspects of the product into one model or even a coherent set of models.

A product connectivity model captures the components of a complex product and the different interactions between its parts in an abstract way as a graph. The possible relations between components of a product can be manifold and depend on the particular application of the model. In recent case studies, an extensive list of different linkage types was used (Jarratt et al. 2004a) to take this into account. The list groups individual parameters to provide an abstract yet specific view.

A challenge for product connectivity models is building models for large and complex products. Products such as helicopters or gas turbines consist of several thousand components, connected in various ways, designed by multidisciplinary teams. Such models need to be hierarchical, representing different levels of abstraction. The models in this paper are small and represent component breakdowns with a size of not more than 100 components. These are usually, again, an abstraction of a more complex model. Thus it is vital that a proper component breakdown is established before the actual model-building exercise.

2.2. PRODUCT CONNECTIVITY MODELS FOR CHANGE PREDICTION

One application of product connectivity models is change prediction (Clarkson et al. 2004). The change prediction method computes risks associated with component changes once the design of a product is finished and allows designers to foresee effects of changes to components before these changes are implemented.

The product connectivity models used for change prediction are elicited in a group meeting with experienced designers who all have different views and knowledge of the product, due to their different backgrounds and responsibilities. The main representation in this model-building exercise is a Design Structure Matrix. Such a model-building session with a group of designers consists of four steps:

1. A component breakdown of a medium level of abstraction is established;
2. A list of possible linkage types is created;
3. The group methodically goes through the list of all pairs of components and decides whether there is a link between these two components and of which type this link is;

4. Direct change likelihood and impact values are assigned to each of the established component connections.

With this methodology it is hoped that most of the links (even very subtle ones) can be detected, while balancing individual biases. As usually very few designers have a complete overview over the entire product (Jarratt et al. 2004b), designers responsible for different aspects of the design are invited to such a meeting to contribute their rows and columns. The values for change impact and likelihood and links between the components elicited in this way reflect the experience of these designers and are usually based on previous designs. However, as Ayton and Pascoe (1995) point out, it is questionable whether these change values truly affect real-world change propagation probabilities, because people generally make mistakes when judging uncertainties.

The benefits of creating a product connectivity model for change prediction are three-fold.

- Learning in the group: A group of designers collectively build the model (Jarratt et al. 2004b) where many of the 'war stories' of the design process emerge. Each designer puts in his or her knowledge and throughout the exercise all designers gain knowledge, even on parts of the final product they are not directly involved with. The exercise also helps to reveal the need for interaction between different design teams.
- Model for analysis: The information on direct change likelihood and impact values assigned to component connections alone can reveal risky component connections. They can also be used to calculate indirect change risks (Clarkson et al. 2004) using the change prediction method and thus to predict the impacts and risks associated with changes to components resulting from indirect connections.
- Product overview: In design decision-making, product connectivity models can provide a necessary overview of the product that can be updated during the product lifecycle. It can even provide a way to integrate new members into the design team, by providing the information stored in the model.

2.3. INDUSTRIAL RELEVANCE OF PRODUCT CONNECTIVITY MODELS

Product connectivity models for change prediction proved their value in several case studies carried out by members of our group. The range of companies involved includes a helicopter company (Eckert et al. 2004), a diesel engine manufacturer (Jarratt et al. 2004a) and a UK aerospace company. Throughout these case studies the change prediction method and a corresponding software support tool (CPM tool) for analysing change propagation were developed and applied. The industrial success of such a change propagation tool, however, also depends on finding a way to present all the desired information visually in such a way that the user (in this case the designer) is not overwhelmed by the amount of information.

The feedback from all the companies that used the product connectivity models was generally positive. The models were used for very different applications, ranging from risk assessment of component change to storing important information on the product. All companies agreed that especially the model-building exercise had a positive impact on the design team when designers involved in the process came together and everyone added his knowledge to the model. We are currently looking for means to improve the CPM tool and especially its human–computer interfaces for building such connectivity models, which is the main motivation for the comparison between DSMs and node-link representation for model-building introduced in the remainder of this paper.

3. Visual Representations of Connectivity Models: DSMs and Node-Link Diagrams

As mentioned earlier, graph structures - the basis for connectivity models - can be represented in different ways. Most common are matrix-based representations and node-link displays. Both representations are equally valid representations of graphs. This section will introduce these two most common visual representations for relational data. It will also show how both representations support the visual display of product connectivity models. These visual representations not only show information provided by the model once it is built, but can also be used to build the model in the first place, which will be the main focus in this paper.

3.1. DESIGN STRUCTURE MATRICES

A Design Structure Matrix, (Steward 1981) is presented as "*a simple, compact, and visual representation*" (Browning 2001) in various literature sources. However, a DSM is not the primary representation that designers would use. For example, a design manager from a UK gas-turbine company responded to the question of whether he used a DSM or a node-link representation when he created a functional model of a gas turbine that he "*created a network* (node-link diagram) *first and then transformed it into a DSM*". Another designer had more general reservations, saying: "*Lets face it, a DSM is not a representation designers like using*". In another user-study conducted on the different factors that influence model-building with DSMs an experienced designer with dyslexia had particular problems building a product model with a DSM and was keen to use a node-link representation. This leads to the question whether a DSM is the most appropriate representation for product model-building as used in current methodologies.

DSMs are essentially adjacency matrices and thus squared matrices; each node of the underlying graph is represented by a row and a column. A mark in a DSM means that there is a link from the element represented by the column of the matrix to the component represented by the row. The definition of how to read such matrices varies amongst different research communities, as it is not inherent in the matrix how it should be read (in this paper, a mark always represents a link from the column to the row). Figure 1 shows two resulting DSMs of the core components of a simple car engine example based on different component orders. These DSMs show only mechanical links.

Simple Engine		1	2	3	4	5	6
Camshaft	1	1			X		X
Crankshaft	2		2	X		X	
Cylinder Block	3		X	3	X	X	
Cylinder Head	4	X		X	4		X
Piston	5		X	X		5	
Valves	6	X			X		6

a)

Simple Engine		6	1	4	3	2	5
Valves	6	6	X	X			
Camshaft	1	X	1	X			
Cylinder Head	4	X	X	4	X		
Cylinder Block	3			X	3	X	X
Piston	5				X	2	X
Crankshaft	2				X	X	5

b)

Figure 1. Two DSMs of a simple car engine, a) alphabetical order, b) ordered by location in the engine. Each mark in the matrix represents a link from the column-component to the row-component.

The advantage of matrix-based representations is that the possible number of different layouts for such a DSM is restricted to the order of the elements in horizontal and vertical direction. Anecdotal evidence indicates 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 doesn't seem to be influenced by what the original order of the components is as long as it is maintained during the entire lifetime of the model so that subjects are familiar with the order. These findings correspond with research into the order of menu items (Card 1981). Somberg (1987) for instance found that users performed fastest with a fixed menu item order when users had experience with the position of the items. Typically, elements of a DSM are ordered by importance, either by their arrangement in the product or by alphabetical order.

However, changing the order of the elements of the matrix can be beneficial for further analyses of a DSM. Techniques such as sequencing or clustering (Browning 2001) of the matrix change the given order to show aspects of the data that cannot be easily examined within the original view (see Figure 1 for two different orders of the same DSM). The advantages of this reordering can be manifold. Sequencing allows the establishment of an order in which (process) tasks have to be executed. Clustering techniques can identify highly-connected clusters of components, which can be the basis for a component breakdown of the entire product.

3.2. NODE-LINK DIAGRAMS

A node-link representation of a connectivity model potentially carries the same information and can be easily transformed into a DSM and vice versa. This begs the question whether it is as useful for representing connectivity models in general, as is a DSM. We think that designers and other users who need to interact with connectivity models should use the representation that is best suited to their particular needs, when building or interacting with models. The main contribution of this paper is the comparison between a DSM and a node-link representation for connectivity model-building. How useful DSMs and node-link diagrams are for analysing and showing such data will not be addressed here and is an area of ongoing research (Keller et al. 2005a). A comparison between the visual affordances of graphs and matrices for reading important information from the representation of this kind of data can also be found in Ghoniem et al. (2004).

In a node-link diagram, each component of a product or process connectivity model is represented as a node, and edges between nodes represent links between these components (see Figure 2 for node-link representations of the car engine model also shown in Figure 1 with a matrix-based representation). For the layout of such a node-link diagram, the entire two-dimensional space can be used. Thus, the number of layouts is much larger than is possible with matrix-based representations. This larger variety of possible layouts allows displays to focus on different aspects of the data. These include:

- Spring Layouts (Huang et al. 1998) to show clusters.
- Hierarchical networks (Schaffer et al. 1993) that can visualize the component hierarchy of products for instance.
- Fisheye views (Furnas 1986) and radial layouts (Jankun-Kelly and Ma 2003) for focusing on one node or a group of nodes.

An extensive collection of possible layout algorithms for node-link diagrams can be found in di Battista et al. (1994). See Figure 2 for some examples of different layouts for a simple graph.

Displaying relational data in a node-link representation, however, has some disadvantages and problems. Different layouts, for instance, give room for ambiguity and especially for very large graphs, the problems of edge-crossings and overlapping nodes can be very severe. While small and sparse node-link diagrams can often be drawn without edge-crossings (these graphs are then called planar), especially large graphs and graphs that are highly connected (with many links between components) cannot be laid out properly.

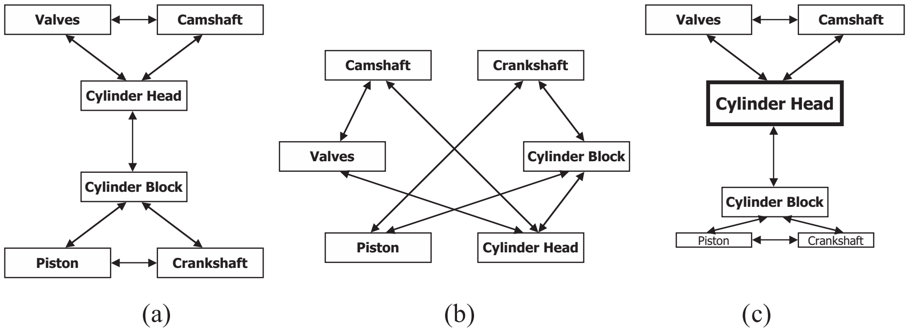


Figure 2. Three node-link diagrams of a simple car engine, a) planar layout without edge-crossings, b) circular layout, c) fisheye view of the graph with a focus on the 'Cylinder Head'.

See Ghoniem et al. (2004) for the implications of size (number of nodes in the underlying graph) and density (number of links divided by the possible number of links) for the readability of node-edge diagrams and matrix-based representations. Unfortunately, some product models are very complex: very large (a helicopter for example has more than 10,000 distinct parts) or very dense (in a diesel engine model, we found that each single component is on average connected to 6 other components which gives a density of 28%). In that case it is vital (also if a matrix-based representation is used) to find an appropriate level of abstraction.

3.3. IMPLICATIONS OF DIFFERENT REPRESENTATIONS FOR PRODUCT CONNECTIVITY MODELS

The main representation implemented in current state-of-the-art software for building connectivity models is the Design Structure Matrix. One example of such a program is the CPM tool described earlier (Jarratt et al. 2004a) for engineering change prediction, which uses a DSM as the main representation for model-building as well as for further analyses of change propagation paths.

Observations of product connectivity model-building in industrial case studies have shown that there is a gap between current methodologies and software support tools on the one hand and what designers seem to prefer on the other hand. While computer support tools, such as the CPM tool, usually incorporate DSMs as the primary representation, we found that the preferred representation depends highly on the designer and the task. In this paper we investigate which representation is more suitable for connectivity model-building using both qualitative (case study observations of using DSMs and node-link diagrams for model-building) and quantitative (an experiment that tests user performance with both techniques) methods. The goal is to include this representation in a computer tool that supports model-building of product connectivity models.

4. Experimental Study

In order to quantify whether a DSM or a node-link representation is better for product connectivity model-building, we conducted a psychological user-experiment (Martin 2003). The aim was to reveal differences in user performance using the two different representations.

For this experiment, 27 participants, all engineering students, ranging from first year students to PhD candidates with practical design experience and PostDocs, were recruited. In a video-recorded pre-study consisting of individual sessions with 6 participants, the general layout of the study was evaluated. Subsequently we held the experiment with the remaining 21 participants. The participants were paid £10 for their time.

Each participant was given sketches of two products, a drawing of the human heart consisting of 8 components and a drawing of a car engine with 16 components, Figure 3. Both examples represented simple systems, with which most of the participants were familiar. This is also true for the heart model, which does not originate from the engineering domain, but due to its simplicity, we believed that is represented a similar level of familiarity to the participants. Time constraints prevented us from using larger models as for instance the diesel engine model that was used in previous studies (Jarratt et al. 2004a).

The given component breakdown was necessary in order to be able to compare different results and also represented a medium level of abstraction. The participants were asked to complete the two models sequentially in the order they were given them. However, we observed one participant working on both examples simultaneously, and not following this order (this is the outlier in the completion time dataset discussed below in Section 5.1.1).

In this study we were interested in how to model spatial relationships, as these are the easiest to infer from a drawing. The participants were asked to create a connectivity model of each of the products, where the linkages should reflect mechanical or spatial relationships between components of the product. We were not interested in the thermal or fluid flow relationships, as we believed that only experts could assess most of these additional links properly.

Each participant then had to complete one model using a DSM and the other one using the node-link representation (so half of the participants completed the heart with a DSM, the other half with a node-link diagram). Additionally, we asked for their experiences with DSMs and their current level of study. No tools other than a pen and paper were allowed. There was no time limit so all the participants had as much time as they needed. The total time spent by the participants including an introduction phase was about 50 minutes with a maximum of 1 hour.

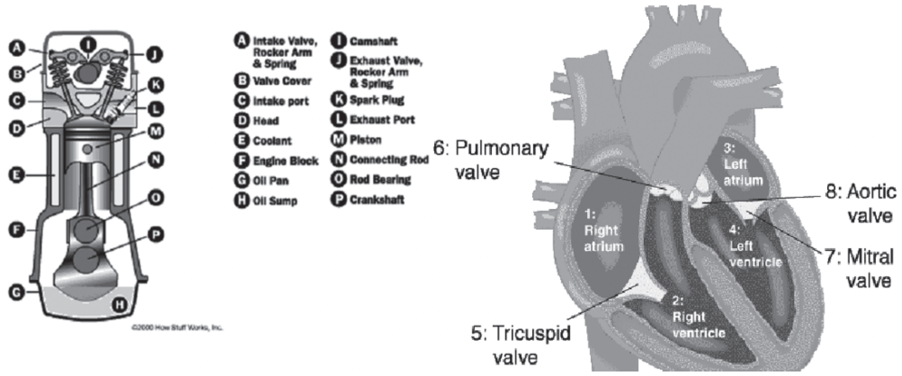


Figure 3. Example 'products' used in the experimental study: An engine with 16 components and a heart with 8 'components'.

Response times were recorded, as well as the resulting node-link diagrams and DSMs. In order to create a single representation, and for calculation issues, all resulting models were transformed into DSMs and then analysed. A link between two components was coded as 1, no link as 0. The results of one user were removed from the dataset as he filled out the DSM for the car engine with a regular pattern.

4.1. COMPARISON BETWEEN DSMS AND NODE-LINK DIAGRAMS

In this section we show the results of the experimental study for the comparison between a DSM and node-link representation. We focus on three different variables, namely the completion time, the number of links found, and the variations found amongst different solutions. We were particularly interested in the effect of the visual representation on these variables and not so much in the correctness of the answers given, as we did not expect anyone to have a very detailed understanding of a car engine or a human heart. For the analysis of the results we used parametric statistical tests as well as a qualitative graphic method (box plots, see Chambers et al. (1983)). Box plots are a well-known way of representing the density of data by showing important statistics (the box of the plots for instance shows 50% of the data, the central horizontal line represents the median).

4.1.1. Completion Time

Initially we were interested in the differences in the completion times for the entire product model. Completion time is a standard measure used in several other studies when comparing readability of node-link diagrams and matrix-based representations (Ghoniem et al. 2004). The null-hypothesis H_0 that there are no differences between the two groups could not be rejected for both models, the car engine and the human heart (using a t-test, as we consider response times as normal-distributed). This means that the differences were not statistically significant.

However, the box plot for the car engine in Figure 4 shows that there is one strong outlier in the response times for the large product model of the car engine. One participant needed significantly more time than any other participant filling out the node-link diagram (see the explanation earlier, he did both models simultaneously). A Grubbs's outlier test also showed that this value is an outlier with a probability of more than 99%. Without this outlier the completion time for the node-link representation was significantly (with a significance level of α=5%) shorter compared to the completion time needed for a DSM. In the model of the heart, even removing the outlier shown in Figure 4(b) does not change the fact that there are no significant differences in the completion times, although in that example, the completion time of a DSM was shorter than that of a node-link representation.

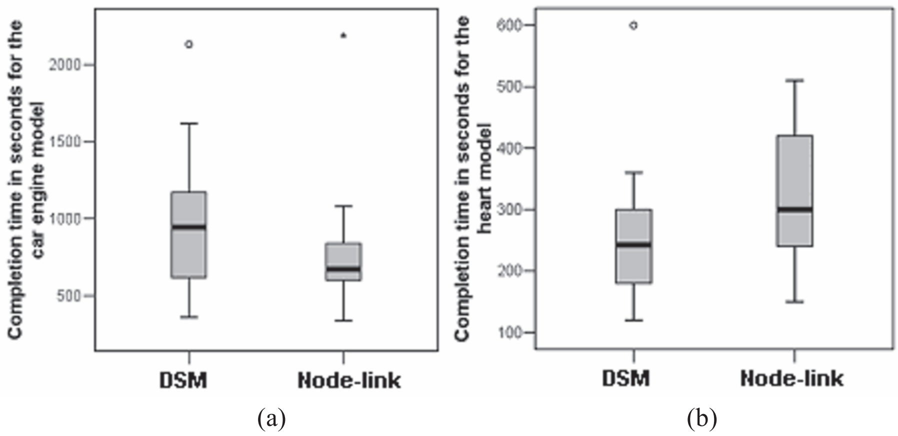
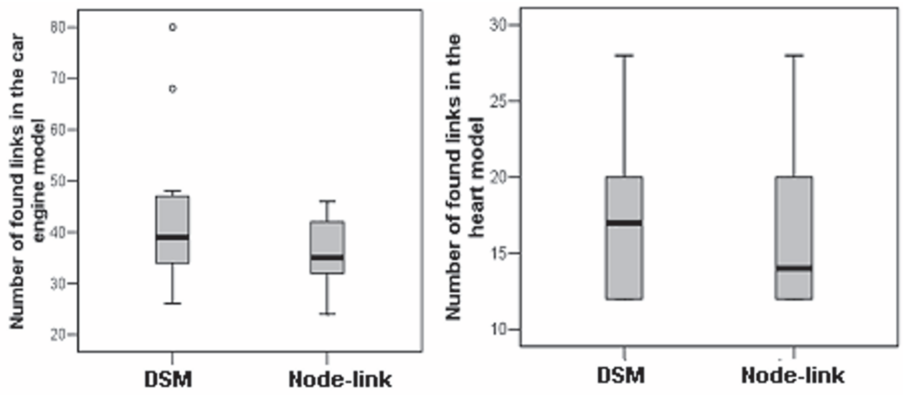


Figure 4. Box plots of the completion time for the car engine model (a) and the heart model (b).

4.1.2. Number of Links

Secondly, we were interested in the number of links found by participants using DSMs and node-link diagrams. The number of links shows whether subjects really considered every possible connection between components and found even links that are not obvious in the first place. We argue that the more links that were found, the more attention was paid to even very weak links and thus, the more complete the model.

We found that the number of links found by subjects using a DSM for the car engine model was significantly larger (α=10%) than the number of links found with a node-link diagram using a t-test. The number of links is binomial-distributed, and as n is large (n=56 or n=240 possible links in the models), the total number of links can be modelled as normal-distributed. Generally, more links were found with a DSM. See Figure 5 for the corresponding Box plots that support this thesis. For the heart model again, no significant differences were detectable. However, it can be seen in the corresponding Box plot, Figure 5, that there is a higher number of links found with the DSM representation.



(a) (b)

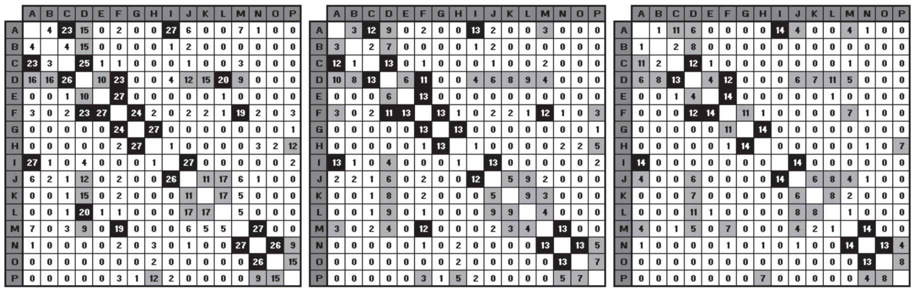
Figure 5. Box plots of the number of found links for the car engine model (a) and the heart model (b).

4.1.3. Variance

In this section we will analyse how the representation of the connectivity model influences the variation of solutions qualitatively. Figure 6 shows the results graphically. On the left there is the DSM that incorporates all 27 solutions (by the 27 participants) of the car engine (the node-link diagrams were transformed into DSMs and then all these DSMs were added). The DSM in the centre shows the solutions that were created by participants using only the DSM, the one on the right shows the results of the users using the node-link representation.

As one can easily see, in general, there is a lot of variation between different solutions. The colour coding in the cells represents whether the number of solutions that had this link is significantly ($p<0.05$ under a binomial distribution) smaller than 0.5 (white background), significantly greater than 0.5 (black background) or not significantly different from 0.5

(grey). The cells with a grey background represent links where there was a high controversy amongst the different solutions in a group.



(a) (b) (c)

Figure 6. DSMs showing all solutions, the DSM solutions and the node-link diagram solutions.

However, as Figure 6 shows, there are hardly any differences between the DSM and node-link diagram solutions. The only real difference is the link between components M (Piston) and F (engine Block), which was found by almost all participants using the DSM (12 out of 13) but was only found by roughly half of the participants using the node-link representation (7 out of 14).

Additionally, one can see that there are more black boxes (i.e. cells with a link that is significantly bigger than 0.5) amongst the DSM solutions (22) than amongst the node-link diagram solutions (16). This corresponds with the finding that, generally, more links were considered using a matrix representation.

4.2. SUMMARY

The experimental study showed that the difference between a DSM and a node-link representation small when used to build product models in an engineering context. We found that the participants using a DSM assessed more links than those using a node-link representation. The time spend on the node-link representation, however, was shorter (for the car engine model). This might be the result of the smaller number of links considered with the node-link diagrams, and for the smaller heart model, participants were faster using DSMs. After the study, we also asked which representation each subject favoured. The answers were even (13 liked DSMs more and 13 node-link diagrams, while one participant could not decide), so there were no detectable user preferences.

5. Discussion

The study introduced in the previous section showed that the differences in user performance using DSMs and node-link representations for product model-building are small. It could be observed that subjects were faster with node-link diagrams (at least for larger product models, such as the engine), and that, using DSMs, in general more links were considered. However, the combined solutions - as seen in Figure 6 - do not significantly differ, but individuals seem to have strong opinions on when to use which representation. Other factors, such as experience of the participants, however, can have huge impacts on how subjects perform at a product model-building exercise, but were not considered in this study.

As stated earlier we were looking for the “best” possible representation for building product connectivity models so that it could be incorporated into the CPM tool. The study showed that the differences between the two representations are small and we see that there seems to be no best representation for building product models. However, it was clearly shown that participants using DSMs filled in more links than those using the node-link representation. Especially in the context of change prediction and propagation, where hidden links can cause huge problems, this is a strong argument for using DSMs for model-building rather than node-link diagrams. The structured way of filling out a DSM (people responded that they usually went column or row-wise through the matrix) lets them consider even links that are not obvious in the first place.

This and the fact that, on the other hand, some users have problems using matrix-based techniques lets us propose that the users should be able to decide which representation suits them best for model-building and thus, as an implication for future software tools, both representations should be incorporated into a software tool in order to effectively support designers in building connectivity models in general. Although this approach means more work for the programmer responsible for the implementation of adequate user interfaces, the benefits for users can be very high and almost no user will be excluded, as opposed to current approaches, which just provide a DSM. The requirements for such a tool must also include a means of linking both representations. Changing the underlying model in one representation should have immediate effects on the other representation. This concept of linking different graphs and representations is especially common in the field of interactive statistical graphics (Unwin 1999). It also follows that, especially when building product models in a group meeting, computer tools are necessary, because it might be beneficial to swap between different representational modes during the meeting to benefit from different representations of the product model.

The CPM tool for predicting change propagation in complex products is capable of supporting the model-building side of the design process. However, while multiple representations are already used for analysing and presenting change data and the resulting matrices (Keller et al. 2005b), the model-building capabilities of the software are still limited to using only DSMs as the primary representation. Our final goal is to bring an updated version of the tool back into an industrial model-building meeting.

Another implication for the use of product models in engineering design is that one cannot rely on models created by one person. As the experimental study showed, there is a lot of variation between different solutions regardless of the visual representation. Future research should reveal whether the current strategy of model-building (a group meeting) is an appropriate methodology. This evidence however supports the current methodology of having a group meeting with different designers rather than relying on a single opinion. The differences among the results detected in this study (where all participants shared a common understanding of the product) will be even greater in an industrial setting where every designer is responsible for a different part of the design and has a different (academic) background.

Furthermore, the implications from this study can be generalized to all sorts of representations that rely on graph structures, such as process models. Building process models involves similar strategies; the structured way of creating models using a matrix-based representation can be very beneficial as even hidden links can be detected more easily.

6. Conclusions

In this paper we compared the visual affordances of DSMs and node-link representations and their ability to support construction of product connectivity models. We found that DSMs and node-link representations can both benefit product model-building differently, as they propagate different building strategies. A supporting experimental study for connectivity model building was conducted, which indicated that DSMs are slightly better for assessing linkages between components of a complex product as participants considered more links when using a DSM than using a node-link representation. It was argued that this is due to the more structured way of representing the data using matrix-based techniques. This finding did not correspond to anecdotal evidence from designers in industry who preferred node-link representations for building connectivity models. However, the overall differences between subjects using different representations were relatively small due to individual preferences. This study also rounds up the comparison of matrix-based representations and node-link representation for connectivity models. As shown in previous studies (Ghoniem et al. 2004;

Keller et al. 2005a), node-link representations are better for reading information from small and sparse graphs and are generally better suited for showing information about indirect links between two components, while matrix-based representations seem to outperform node-edge diagrams when a more structured way of interaction with the underlying model is required, such as was shown here for product connectivity model-building.

Due to the small differences, we propose a strategy for connectivity model-building that makes use of multiple representations rather than one primary visual representation. This would allow designers in industry to choose their preferred representation, and would benefit the design process by facilitating the integration of their knowledge into the product models. The differences between individual solutions suggest that it would be preferable to rely on multiple opinions than on the judgments of one single expert, which is supported by current methodologies (Jarratt et al. 2004a; Austin et al. 2001). However, whether this knowledge should be elicited in a meeting with different designers or whether several solutions by single designers should be incorporated into one model by a researcher is still unresolved and is a topic of future research.

The results of this paper mainly refer to building product connectivity models in an engineering context. However, as connectivity models of processes (as for instance task DSMs or PERT methods) follow similar concepts in terms of connectivity and the creation of such models is also highly dependent on the visual representation of the underlying graph structure, these findings should be also applicable for building such models.

Acknowledgements

This research is funded by EPSRC.

References

Austin, S, Steele, J, Macmillan, S, Kirby, P and Spence, R: 2001, Mapping the conceptual design activity of interdisciplinary teams, *Design Studies* **22**(3): 211-232.

Ayton, P and Pascoe, E: 1995, Bias in human judgement under uncertainty?, *The Knowledge Engineering Review* **10**(1): 21-41.

Browning, TR: 2001, Applying the design structure matrix to system decomposition and integration problems: A review and new directions, *IEEE Transactions on Engineering Management* **48**(3): 292-306.

Card, SK (ed): 1981, User perceptual mechanisms in the search of computer command menus, *Proceedings of the 1982 Conference on Human Factors in computing systems*, Gaithersburg, Maryland, US, pp. 190-196.

Chambers, J, Cleveland, W, Kleiner, B and Tukey, P: 1983, *Graphical Methods for Data Analysis*, Wadsworth.

Clarkson, PJ and Hamilton, JR: 2000, 'Signposting', A parameter-driven task-based model of the design process, *Research in Engineering Design* **12**(1): 18-38.

Clarkson, PJ, Simons, C and Eckert, CM: 2004, Predicting Change Propagation in Complex Design, *ASME Journal of Mechanical Design* **126**(5): 765-797.

Cowan, N: 2001, The magical number four in short-term memory: A reconsideration of mental storage capacity, *Behavioral and Brain Sciences* **24**: 87-185.

Di Battista, G, Eades, P, Tamassia, R and Tollis, IG: 1994, Algorithms for drawing graphs: An annotated bibliography, *Computational Geometry: Theory and Applications* **4**(5): 175-198.

Eckert, CM and Clarkson, PJ: 2003, The reality of design process planning, *ICED 03*, Stockholm, Sweden (on CD).

Eckert, CM, Clarkson, PJ and Zanker, W: 2004, Change and customisation in complex engineering domains, *Research in Engineering Design* **15**(1): 1-21.

Furnas, GW: 1986, Generalized fisheye views, *Proceedings of CHI '86*, Boston, Massachusetts, USA, pp. 16-23.

Gero, JS and Reffat, RM: 2001, Multiple representations as a platform for situated learning systems in designing, *Knowledge-Based Systems* **14**: 337-351.

Ghoniem, M, Fekete, J-D and Castagliola, P: 2004, A comparison of the readability of graphs using node-link and matrix-based representations, *Proceedings of InfoVis 2004*, Austin, Texas, USA, pp. 17-24.

Henderson, K: 1999, *On Line and On Paper*, The MIT Press, Cambridge, MA, USA.

Hoover, SP, Rinderle, JR and Finger, S: 1991, Models and abstractions in design, *Design Studies* **12**: 237-245.

Huang, ML, Eades, P and Wang, J: 1998, Online graph drawing using a modified spring algorithm, *Australian Computer Science Comm: Proceedings 21st Australasian Computer Science Conference, ACSC* **20**(1): 17-28.

Jankun-Kelly, TJ and Ma, K-L: 2003, MoireGraphs: Radial focus+context visualization and interaction for graphs with visual nodes, *Proceedings of InfoVis 2003*, Seattle, Washington, USA, IEEE Computer Science Press, pp. 59-66.

Jarratt, T, Eckert, CM and Clarkson, PJ: 2004a, Development of a product model to support engineering change management, *Proceedings of Tools and Methods of Competitive Engineering (TCME)*, Lausanne, Switzerland: 331-342.

Jarratt, T, Eckert, CM, Clarkson, PJ and Stacey, MK: 2004b, Providing an overview during the design of complex products, *in* JS Gero (ed), *Design Computing and Cognition, DCC'04*, Kluwer, Dordrecht, pp. 239-258.

Keller, R, Eckert, CM and Clarkson, PJ: 2005a, Matrices or network diagrams: Which visual representation is better for visualising connectivity models? *CUED/C-EDC/TR137*, Cambridge, University of Cambridge.

Keller, R, Eger, T, Eckert, CM and Clarkson, PJ: 2005b, Visualising change propagation, *Proceedings of the 15th International Conference on Engineering Design (ICED '05)*, Melbourne, Australia (on CD).

Larkin, JH and Simon, HA: 1987, Why a diagram is (sometimes) worth ten thousand words, *Cognitive Science* **11**: 65-99.

Malcolm, DG, Roseboom, JH, Clark, CE and Fazar, W: 1959, Applications of a technique for research and development program evaluation (PERT), *Operations Research* **7**(5): 646-669.

Martin, D: 2003, *Doing Psychology Experiments*, Wadsworth.

Pahl, G and Beitz, W: 1996, *Engineering Design - A Systematic Approach*, London, Springer-Verlag.

Schaffer, D, Zou, Z, Bartram, L, Dill, J, Dubs, S, Greenberg, S and Roseman, M: 1993, Comparing fisheye and full-zoom techniques for navigation of hierarchical clustered networks, *Proceedings of Graphics Interface '93*, Toronto, Canada, Morgan-Kaufman, pp. 87-96.

Simon, HA: 1998, *The Sciences of the Artificial*, Cambridge, MA, MIT Press.

Somberg, BJ: 1987, A comparison of rule-based and positionally constant arrangements of computer menu items, *SIGCHI/GI conference on Human factors in computing systems and graphics interface*, Toronto, Ontario, Canada, pp. 255-260.

Star, SL: 1989, The Structure of ill-structured solutions: Boundary objects and heterogeneous distributed problem solving, *in* L Gasser and MN Huhns (eds), *Distributed Artificial Intelligence*, Pitman London, UK, pp. 37-54.

Steward, DV: 1981, The Design Structure System: A method for managing the design of complex systems, *IEEE Transactions on Engineering Management* **28**(3): 71-74.

Unwin, AR: 1999, Requirements for interactive graphics software for exploratory data analysis, *Computational Statistics* **14**: 7-22.

Zeitz, CM: 1997, Some concrete advantages of abstraction: How experts' representations facilitate reasoning, *in* PJ Feltovich, KM Ford and RR Hoffman (eds), *Expertise in Context: Human and Machine*, AAAI Press, pp. 43-65.