INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED11
15 - 18 AUGUST 2011, TECHNICAL UNIVERSITY OF DENMARK

IMPROVING DATA QUALITY IN DSM MODELLING: A STRUCTURAL COMPARISON APPROACH

Steffen F- Schmitz[1], David C. Wynn[2], Wieland Biedermann[1], P. John Clarkson[2] and Udo Lindemann[1]
[1]Technische Universität München, [2]University of Cambridge

ABSTRACT
The Dependency Structure Matrix (DSM) has proved to be a useful tool for system structure elicitation and analysis. However, as with any modelling approach, the insights gained from analysis are limited by the quality and correctness of input information. This paper explores how the quality of data in a DSM can be enhanced by elicitation methods which include comparison of information acquired from different perspectives and levels of abstraction. The approach is based on comparison of dependencies according to their structural importance. It is illustrated through two case studies: creation of a DSM showing the spatial connections between elements in a product, and a DSM capturing information flows in an organisation. We conclude that considering structural criteria can lead to improved data quality in DSM models, although further research is required to fully explore the benefits and limitations of our proposed approach.

Keywords: Design Structure Matrix, Knowledge elicitation, Structural similarity

1 INTRODUCTION
The Dependency/Design Structure Matrix (DSM) is a modelling approach which can help to visualise and manage the structure of dependencies in a complex system such as a product, process or organisation [1]. It is a useful tool for eliciting the structure of dependencies, as each cell in the matrix can be systematically considered to determine whether a dependency exists and what its nature is. Many analysis approaches based on DSM modelling have been proposed to assist the design, optimisation and maintenance of complex systems. To illustrate, Table 1 summarises some methods which have been used to analyse system structures modelled as a DSM.

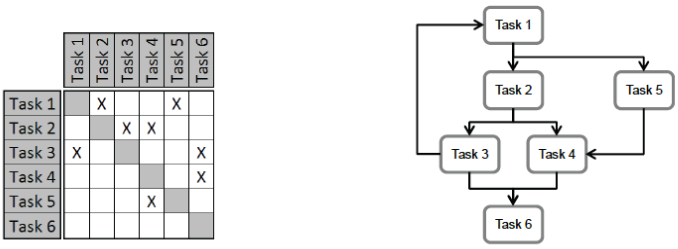

Figure 1: The Design Structure Matrix represents the dependencies in a complex system

The DSM has proved to be a useful tool for eliciting and analysing system structures. However, as with any modelling approach, the insights gained from analysis are limited by the quality and correctness of input information [1,2]. Relatively few studies have provided methods to create better quality DSMs or to evaluate the quality of DSMs. Furthermore, in a recent survey of DSM modellers, "methods for data elicitation" was identified as one of the most pressing opportunities for improving the methodology [3]. In this paper, we discuss methods which allow DSM modellers to raise the quality of their models by cross-checking, helping to pick up accidental mistakes during the elicitation process. In turn this should increase the accuracy, objectivity, and confidence in DSM-based analyses. We limit the analysis in this paper to consider acquiring only the existence, or not, of binary dependencies between a predefined list of elements in a system. We do not consider procedures for eliciting the 'strength' of connections or for eliciting the elements that comprise a system.

Table 1: Some approaches to analysing a system structure represented as a DSM

Analysis	Description, application
Clustering	Identifies clusters, where elements within a cluster possess many dependencies with each other but few with elements in other clusters.
Distance matrix	The distances between the system elements. It can be realigned to identify groups of indirect dependencies.
Matrix of indirect dependency	The number of paths between any pair of elements.
Partitioning, sequencing	The reordering of the rows and columns of a DSM in order to place all elements a) on one side of the diagonal or b) at least close to the diagonal has several names such as partitioning or sequencing.
Banding	Indicates mutually-independent groups of consecutive elements in a given sequence.
Change propagation	Identify how change initiated in one subsystem can propagate through dependencies to ultimately require rework in many others
Process simulation	Identify how rework generated by interdependent tasks adds to project lead time

2 BACKGROUND

It has been said that all models are created for a purpose; and that while all models are wrong, some are more useful than others. This section discusses some of the aspects of data acquisition which influence the utility of a model to support a given purpose, and reviews some prior research regarding data acquisition for the Dependency Structure Matrix (DSM).

2.1 Different sources of information

A number of acquisition-related activities are required to create any DSM. Some of the key activities are shown in Table 2 (in reality these steps may be disordered and iterative).

Table 2: Some key activities in acquiring information for a DSM model

Activity in acquisition process	Example (for vacuum cleaner)
1. Identify breadth of modelling	Entire product (but not user or use context)
2. Identify depth of modelling	Major parts (all mouldings but not screws etc.)
3. Identify types (and sub-types) of element	Modules (e.g., cyclone) and parts (e.g., mouldings)
4. Identify types (and sub-types) of dependency	Physical connections only – spatial mechanical, spatial static
5. Create (possibly hierarchical) list of elements	Handle, cyclone rear moulding, cyclone top moulding, etc.
6. Elicit dependencies between elements	Via dismantling workshop
7. Check data	Via cross-checking

Considering the elicitation of dependency information, which is the focus of this paper, several methods may be used. These include:

- **Direct extraction of dependencies from databases or other 'hard data'** (e.g., a PLM system or links on the world-wide web)
- **Analysis and superposition of existing documentation** (e.g., process maps or org. charts, each covering some part of the system to be modelled)
- **Questionnaires/Surveys in which dependencies are elicited directly** (e.g., asking: who do you talk to in an organisation?)
- **Workshops with domain experts** (e.g., working systematically across rows and columns and considering each possible dependency in turn)

2.2 Different perspectives of dependencies in a system

The more complex a system is, or the more stakeholders interact with it, typically the more perspectives are available from which information can be elicited and from which a model of dependencies can be built. A complex system such as an organisation or a process, which to a large extent exists only in the minds of its stakeholders, can be considered to contain several sorts of information that can be viewed from different angles and perspectives. For instance, some people are closely involved in a particular design process and therefore have detailed knowledge, whereas others may only have a broad overview of that process, on a more abstract level. Some would consider work in a process to be divided into business functions whereas other would organise their view according to lifecycle phases. Some participants would see the pertinent border of a system to be wide, whereas others would set the modelling scope more narrowly. Furthermore, not all participants have equal knowledge of a system – which should be considered when acquiring data. Depending on the type of a system, knowledge about its dependencies may take several forms. These can be considered along different axes, including: *Objective vs. Subjective dependencies* (e.g., connections in a product vs. communication flows in an organisation; *Tacit vs. Explicit knowledge*; and Internal vs. *External information* (i.e., information which must be elicited directly from process participants vs. that which can be identified from documents).

2.3 Prior research on data acquisition for the DSM

One of the main issues associated with data acquisition for the DSM is overcoming problems of scaleability. While a benefit of the DSM method is its suitability for systematic consideration of possible connections, this benefit is quickly eroded as the number of elements increases, because the number of potential dependencies is $n^2 - n$, where n is the number of elements. Thus, eliciting the connections between 10 elements requires consideration of up to 90 cells, while a matrix with 100 elements requires consideration of 9,900 cells. Complex engineering systems, such as aircraft or design organisations, may easily contain thousands of elements which could be modelled.

In general, moderately large DSM models may be difficult to elicit for a number of reasons, including:

- **High effort.** A lot of effort is required to consider all possible dependencies, as explained above.
- **Distributed knowledge.** People may not know about the whole system; it may be difficult to identify who should be asked about what, and it may be necessary to negotiate regarding the existence (or not) of particular dependencies.
- **Disorientation.** In a large matrix, dependencies may be easily placed in the wrong cells – due to alignment difficulties or disorientation when faced with a large grid.
- **Ambiguity.** System elements may be misinterpreted if they are similar or given similar names, so that dependencies may be mistakenly identified.
- **Interface limitations.** It is difficult to visualise large matrices, using computer software or paper methods – “like viewing a map through a letterbox”.
- **Fatigue.** Modellers may become fatigued; typically the cells in the first few rows of a large matrix are more carefully considered than the last few when identifying dependencies.

A number of authors have discussed approaches to mitigate such concerns when acquiring data for the DSM, aiming to reduce acquisition effort, to improve modelling quality, or both. Many of these authors have quantified the effort reductions of their approaches, in terms of the number of elicitation operations required for different schemes. Some of the existing approaches are summarised in Table 3.

Table 3: Examples of approaches to support DSM data acquisition in the literature

Approach	Explanation	Effort implications	Quality implications
Element hierarchical-sequential (effort reduction)	Elicit one matrix connecting dependencies between subsystems, to rule out possible dependencies at lower level (e.g. [1])	Reduce effort by focusing elicitation (not all cells require consideration)	Possibly leads to missed dependencies between weakly connected subsystems
Element hierarchical-sequential (quality improvement)	Elicit one high-level matrix and one low-level matrix. Look for discrepancies (e.g. [1])	Increase effort (subsystem dependencies also required)	Improves quality by hierarchical comparison
Element hierarchical-concurrent	Divide the system into subsystems. Have groups of experts elicit dependencies within a subsystem, and interface dependencies (e.g. [4])	Reduce effort by focusing elicitation (although interfaces elicited multiple times)	Improve quality (of interface descriptions only)
Dependency hierarchical-concurrent	Break down dependency type into sub-types, elicit matrices, then combine. (e.g. [5])	Increases effort	Improves quality by deeper consideration of the existence (or not) of a dependency

2.4 Summary

The ultimate objective of methods to support data acquisition for the DSM is to reduce effort and/or improve quality. In one sense, improving quality can be considered as minimising errors in the model. However, for many systems which are too complex or subjective to access directly, it is difficult to quantify 'error' – even in principle, because the model itself provides the only baseline for comparison. Therefore, in this paper we aim to develop methods to help identify and minimise disagreement between views of a system, which we view as helping to 'triangulate' between perspectives and thus reach a better understanding. We consider that such disagreements may include:

- **Disagreements in perspective/interpretation** – i.e., different models take a different scope or focus, or disagree on the existence (or not) of a dependency.
- **Disagreements in data** – i.e., data does not reflect the system as represented by other data.
- **Disagreements in transcription** – i.e., the modeller may make mistakes while creating a model, which as a result unintentionally differs from their understanding of it.

3 APPROACH OVERVIEW

The premise of this paper is that disagreements in models can be highlighted by comparisons between different points of view, allowing their reconsideration and ultimately improving data quality. We consider comparisons between a given model and:

- A prior understanding of the system that model represents;
- Another model elicited by another person or method;
- Another model elicited from a different perspective or different level of abstraction.

In particular, comparisons are made using the following guiding principles:

- **Network structure constraints can highlight disagreements between a model and an understanding of the system type.** Knowledge of the type of system represented in a DSM can provide information to the elicitation process which results in better-quality DSM models.
- **Structural comparison between models can highlight disagreements between models**. Multiple views of the dependencies in a system, elicited from different perspectives, can be compared from a structural point of view, highlighting potential weaknesses in the data and thus assisting the modeller in raising its quality.

Each of these principles is discussed in greater detail below.

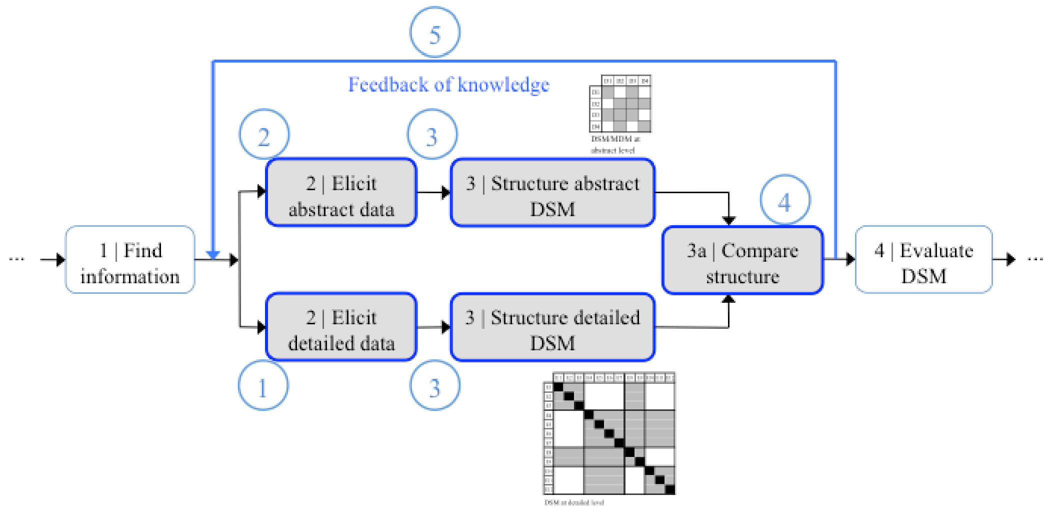

Figure 2: Comparison of data on different levels of abstraction

3.1 Network structure constraints

If prior knowledge about a system is taken into account, it is possible to place very basic network structure constraints on the DSM. For instance, a matrix of structural interactions between physical parts, or of communication frequency between individuals in an organisation should be symmetric. Likewise, a reporting network in an organisation, or certain workflows, should contain no cycles. Considering these constraints, it is possible to identify marks which are not bidirectional or cycles in the network, and highlight them for consideration by the modeller. Each item thus identified suggests that a mistake has been made, and helps pinpoint its possible location.

3.2 Structural comparison between models

A system can be viewed from many different perspectives. However, all these views should have some aspects in common. Thus it should be possible to spot some inconsistencies in a system model by comparing two or more different perspectives. This strategy is used by several of the approaches listed in Table 3; but is expanded here as it is central to our proposed approach.

The most conceptually straightforward comparison approach is to elicit the dependencies between a set of system elements twice, by different methods or from different stakeholders, and compare directly. However, this is effort-intensive.

A second approach, summarised in Figure 2 is to compare a detailed DSM with a more abstract DSM, where multiple elements in the detailed DSM can be mapped to one in the abstract DSM. (For instance, a set of parts in the detailed DSM of a product might be mapped to a single module in the abstract DSM). Apart from reducing effort in comparison to the first approach, a secondary benefit of this method is that the dependencies in the abstract DSM can potentially be elicited by people that are broadly familiar with the system concerned but do not necessarily have all the detailed knowledge. The two matrices can then be compared directly, by assuming that a mark between two elements in the abstract DSM should correspond to one or more dependencies between their sub-elements in the detailed DSM. The approach taken in this paper aims to extend this idea by comparing matrices according to their structural characteristics, as explained below.

From a structural analysis standpoint, all discrepancies between two supposedly-congruent views of a system are not equally important. Where multiple discrepancies exist, it should be possible to focus on those which are likely to have most significance to a structural analysis of the system. Thus, we propose that dependencies in two perspectives of a system can be ranked according to their structural importance, according to multiple criteria. If the two models do not agree, but the structural significance of the disagreement is low according to some criteria, it might be said that comparison of the perspectives indicates high-quality data. On the other hand, if the comparison indicates that the matrices have high structural disagreement, it might be said that further elicitation work is necessary to refine the information. Furthermore, it may be possible to pinpoint the important disagreements, thus helping to focus efforts to improve the data.

4 CRITERIA FOR STRUCTURAL COMPARISON BETWEEN DSM MODELS

In this section, we discuss metrics which were considered as the basis for structural comparisons between DSM models in this paper. The comparison approach is then illustrated through two case studies of dependency structure elicitation.

4.1 Network comparison metrics

Numeric comparison metrics convert each matrix into a single number. The numbers can then be compared directly to see if the matrices agree according to the criterion of interest.

Degree of Connectivity

All existing edges (EE) between the elements (n) are put in relation to the quantity of all possible edges (PE) for both DSMs being compared. The ratio (R) between EE and PE is known as the degree of connectivity. Mathematically this can be described as follows:

$$EE = \text{count of all existing edges} \qquad \textit{Equation 1}$$

$$PE = n \times (n - 1) \qquad \textit{Equation 2}$$

$$R = EE / PE \qquad \textit{Equation 3}$$

This ratio results in the degree of connectivity (DoC) which can assist plausibility checks in similar systems or models of the same system [6]. To enable comparisons, the DoC for a detailed matrix should be less than or equal to that for a abstract matrix of the same system (because one connection in a cell in the abstract matrix implies at least one in the equivalent cells of the detailed matrix).

4.2 Dependency comparison metrics

Dependency comparisons convert each matrix into a matrix of numbers, where each cell in the result indicates the importance of the corresponding dependency in the original matrix according to a particular criterion. These metrics can then be compared for different views of the same system to highlight large discrepancies. To compare metrics from matrices of different sizes, it is necessary to map the score for each dependency in the abstract DSM onto the equivalent dependencies in the detailed DSM. The means by which this is achieved depends on the metric under consideration, as discussed below.

Minimum distance

The minimum distance metric describes the shortest possible distance between two given elements by traversing dependencies in the structure. A distance of one, for example, means that a direct dependency exists between the two elements. A distance of two indicates that the shortest path between the two elements is of length 2. A matrix containing binary direct dependencies (DD) is equivalent to a distance matrix of length one (ID1), and can be transformed to a distance matrix with indirect dependencies (ID2) of length two by squaring – and so on for higher powers. Mathematically:

$$ID1 = DD \qquad \textit{Equation 4}$$

$$ID2 = ID1 \times DD = DD^2 = ID1^2 \qquad \textit{Equation 5}$$

$$Dn = ID(n\text{-}1) \times DD \qquad \textit{Equation 6}$$

The minimum distance metric between any pair of elements may thus be calculated by finding the lowest value of n for which the dependency first appears in Dn.

For the structural comparison, it is necessary to compare minimum distances for matrices of different size. This requires consideration of the number of detailed sub-elements which are wrapped up into a single abstract element. For instance, if two elements in the detailed matrix are wrapped into one in the abstract matrix, the four distance values are added, divided by four and divided by its square value. This considers the fact that that elements which are connected through a short chain of dependencies generally have stronger influence on each other than pairs that are connected only via long chains.

$$RV\ (detailed) = \sum 1/\ (4 * SEV2) \qquad \textit{Equation 7}$$

A further step is required to take the higher number of subelements on the detailed level into account. For each of the two matrices, cell values that are above the average value of all cells in the given matrix are set to 1; otherwise, the cell value is set to 0. The matrices are then combined by setting the value of a given cell to 1 if the two matrices are in agreement for that cell (both 0 or both 1); otherwise the value is set to 0. This single matrix is then unwrapped to yield an x-y graph as shown in Figure 4. To illustrate application of this metric, the top graph shows comparison of a detailed DSM to an abstract DSM which exactly summarises it. The lower graph shows the same matrix with one incorrect dependency introduced. This highlights the impact of the incorrect dependency on the minimum distance between several other pairs of elements. The 'amount' of discrepancy as revealed by this metric depends both on the structure of the data and the location of the error. It can be used to highlight and compare the potential importance of discrepancies.

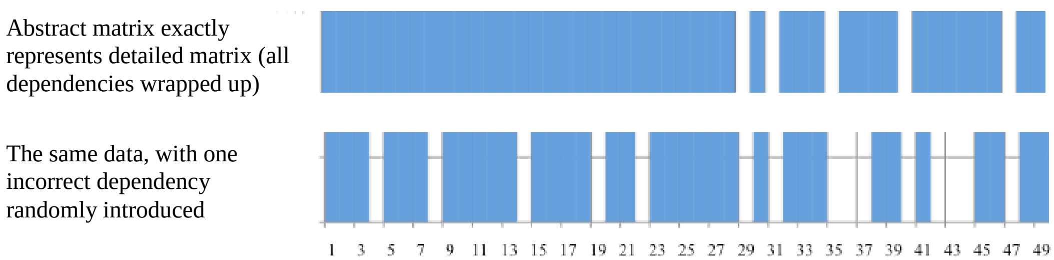

Figure 4: Example comparison using minimum distance criterion

Number of indirect dependencies of length N

Whereas the distance matrix is built from the lowest possible length of indirect dependency, the number of indirect dependencies metric, concerns the number of indirect dependencies between a given pair of elements for a given path length.

The value (RV) of a given cell in the abstract matrix is generated as follows. The value (V) shown in the matrix of indirect dependencies of a specific length is multiplied by the amount of relations (AR) in the detailed DSM. This takes into account that the relations between subelements do not all need to exist if the corresponding relation exists on the abstract level.

$$RV\ (abstract) = V \times AR \qquad \text{Equation 8}$$

The value (RV) of the same cell for the detailed matrix is calculated as follows. The degree of filling of subelements is calculated dividing the number of existing relations (EE) by the number of possible relations (PR) for the given matrix. This value is multiplied by the amount of indirect dependencies per sub-element of length four and scaled by an appropriate value to allow direct comparison.

$$RV\ (detailed) = 0{,}125 \times (EE / PR) \times \sum SEV \qquad \text{Equation 9}$$

These values are calculated for each cell in the matrix and plotted on a 2D chart using similar technique to that described above. Using the same example as above, this is plotted in Figure 5 to illustrate. In each chart, the RV for each cell in the detailed matrix is shown in grey; the corresponding RV for the abstract matrix is shown in blue. This metric clearly highlights the location of discrepancies, but also the magnitude of their impact in terms of the number-of-indirect-dependencies metric.

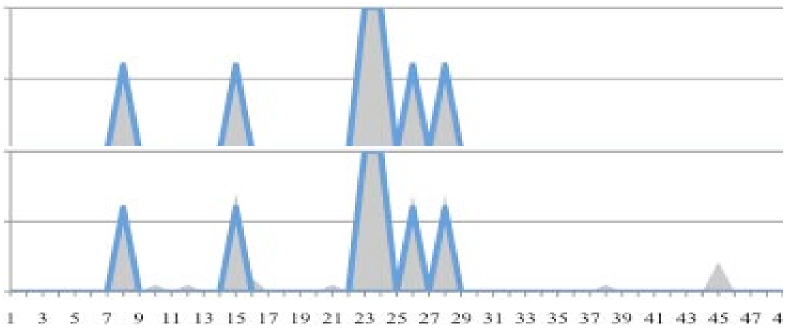

Figure 5: Example comparison using number of indirect dependencies criterion

4.3 Summary

The three metrics discussed above are not an exhaustive selection of the comparison criteria that could be used. They were chosen for illustrative purposes and are used in the case studies described below. Any network structure metric could be used for identifying important discrepancies; for instance, Kreimeyer [7] identifies over 100 metrics that could be adapted to serve this purpose. Ultimately, the particular set of metrics which should be used would depend upon the purpose for creating the matrix. For instance, the sensitivity of a process simulation or change propagation analysis to differences in certain metrics might be more pronounced than others. In such cases, the more critical metrics should be identified and used as the basis to support elicitation. On the other hand, if the objective is to elicit an overall system structure without a particular analysis in mind (for instance, to gain overview of a product architecture), or to create a model for multiple purposes, it might be appropriate to consider a broader set of metrics when comparing multiple views of a system to consider their quality.

5 CASE STUDIES

To illustrate the proposed approach, case studies were undertaken in which multiple models were elicited of 1) the structure of components in a product; and 2) the communication flows in an organisation. These DSMs were created using the Cambridge Advanced Modeller software, and a structural comparison was performed to highlight discrepancies in the DSMs for each case.

5.1 Product DSM of a vacuum cleaner

System overview

The vacuum cleaner fulfilled the criteria of being simple enough to understand and comprising a manageable amount of elements, while remaining complex enough to justify modelling and being able to sensibly cluster the list of parts into modules to obtain an abstract DSM for structural comparison. Figure 6 shows the device which was used, the *Argos Value VC9730S-6*.

Data acquisition

Acquiring lists of elements. The vacuum cleaner was taken apart and 30 parts identified, focusing on 'major' parts such as mouldings and not including connectors, screws, clips etc. On the abstract level two different modularisations were identified – namely, a 5 x 5 and 7 x 7 DSM.

Acquiring dependencies for detailed DSM. To identify all spatial connections between the parts, disassembly workshops were held with each of eight participants. Each participant was given the vacuum cleaner with fasteners removed to ease disassembly. They were also given the empty 30x30 matrix, with instructions to capture spatial connections between the listed parts. The time taken to elicit the detailed level DSM (435 possible relations) was about an hour for each participant.

Acquiring dependencies for abstract DSMs. All three abstract level DSMs (37 possible relations) were filled during similar workshops. Five people did this independently. These people were not involved in the detailed elicitation workshops, thus had not encountered the detailed description of the vacuum cleaner. On average, this took about 10 minutes for a participant to elicit all three abstract level DSMs, for a total of 20 possible relations.

Overview of acquired data

In the abstract DSMs shown in Figure 6, each relation between clusters is shown if at least three of the participants see a relation between clusters. In the detailed DSM, a mark is shown if at least four of the eight participants had identified a relation between those elements.

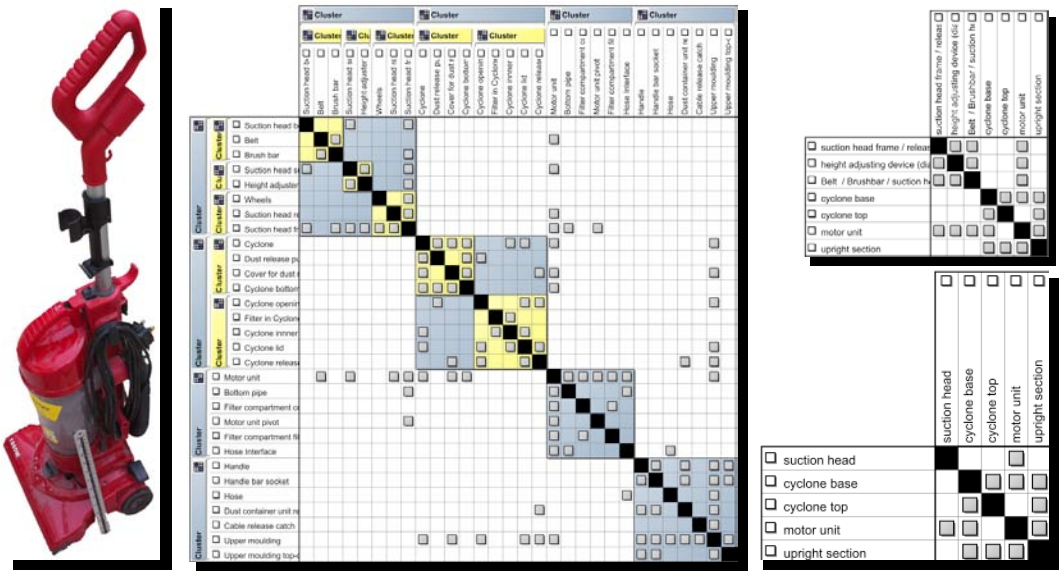

Figure 6: Different views of the vacuum cleaner and connections within it.

Analysis of acquired data

Comparing the 30x30 and 5x5 matrices according to the hierarchy criterion results in a 90% match. Both DSMs are entirely symmetric, which does not highlight any possible errors on the symmetry criterion and suggests a good-quality model. As expected, the degree of connectivity for the detailed DSM is lower than for the abstract DSM. Finally, Figure 7 shows the comparison between the detailed (30 x 30) and the abstract level DSM (5 x 5) using the two dependency comparison metrics.

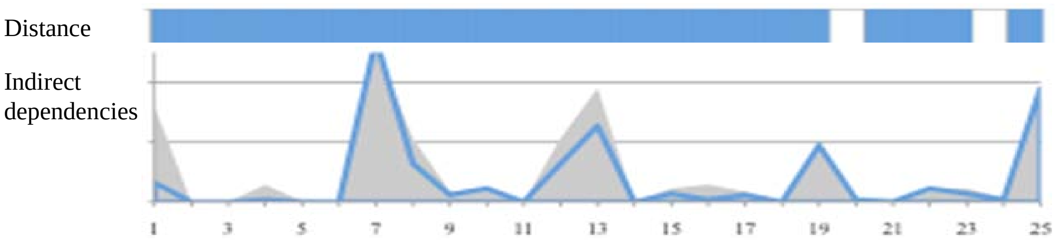

Figure 7: Cell-wise comparison of the dependency comparison metrics shows few disagreements between models

The metrics thus indicate that the two DSMs are strongly in agreement with regards to the basic structural criteria discussed. This suggests that the model is of high quality, which was expected because the system being modelled was largely objective, not too complex, and the two models each represented an agreement between multiple modellers working independently.

5.2 Organisation DSM of a research group

System overview

The second case study is the elicitation of an organisational DSM based on the communication flows in the Engineering Design Centre (EDC) in the Engineering Department at the University of Cambridge. This differs from the first case in that the existence (or not) of a dependency is potentially far more subjective, and because knowledge of communication flows is distributed among many people. Thus, more disagreements would be expected than for the vacuum cleaner model.

Data acquisition

Acquiring lists of elements. The lists of elements were acquired directly from the EDC website. For the detailed DSM, 47 researchers working in the EDC were identified (not including academic staff). For the abstract DSM, the seven research themes within the EDC were listed.

Acquiring dependencies for detailed DSM. An online survey was constructed and distributed to capture the interaction between individuals. Each member of the EDC identified on the list of elements was asked to rate the frequency and intensity of communication with every other member (i.e., to work down the list of 47 and select either none, low, medium or high for each of frequency and intensity). Individuals were also asked to identify which of the 6 research themes they work in, where each person may work in more than one theme. 45 of 47 members completed the survey.

Because only binary dependencies are considered in the structural comparison metrics, the responses were filtered to show a dependency between two people if a medium or high level of communication was described for both frequency and intensity. This filtered out the weaker dependencies within what would otherwise be a very strongly-connected model, and allowed its treatment as a binary matrix.

Acquiring dependencies for abstract DSMs. The interaction between research themes was extracted in two ways, resulting in two abstract matrices. Firstly, each EDC member was asked, while filling out the survey described above, to also indicate the levels of communication (frequency and intensity) that they had with each of the 7 themes as a whole. Thus, if they spoke to any person within a given theme on a daily basis, they would select 'high' for frequency of communication with that theme. These responses were compiled into a single 7x7 abstract DSM using the filtering procedure outlined above. Secondly, five Senior Research Associates were asked to separately fill a 7x7 DSM indicating the frequency and intensity of communication they believed occurred between the seven research themes. These 5 DSMs were filtered individually, then accumulated into a single abstract DSM by including only those dependencies which at least 4 of the 5 participants had identified.

Overview of acquired data

The three DSMs which were acquired are shown in Figure 8.

Analysis of acquired data

The comparison between the detailed matrix and each of the two abstract matrices, according to the structural metrics, is shown in Figure 9 and discussed below.

Comparison of detailed DSM to abstract DSM obtained through survey

The hierarchy constraint shows well over 50% correlation between the detailed organisation DSM and the abstract DSM obtained through the survey. This shows that relationships between members of the EDC and between research themes match fairly well. In cases where no relations between research themes exist, no or few relations between their members exist, and vice versa. Considering the symmetry constraint, both levels of abstraction were expected to be completely symmetric, yet this was not entirely the case. Closer examination of the underlying data set suggested that certain respondents responded across all interactions that the level of communication to their colleagues is far higher than their colleagues perceive it. This systematic bias, perhaps due to imprecise wording in the survey, accounted for the missing symmetry and could be corrected.

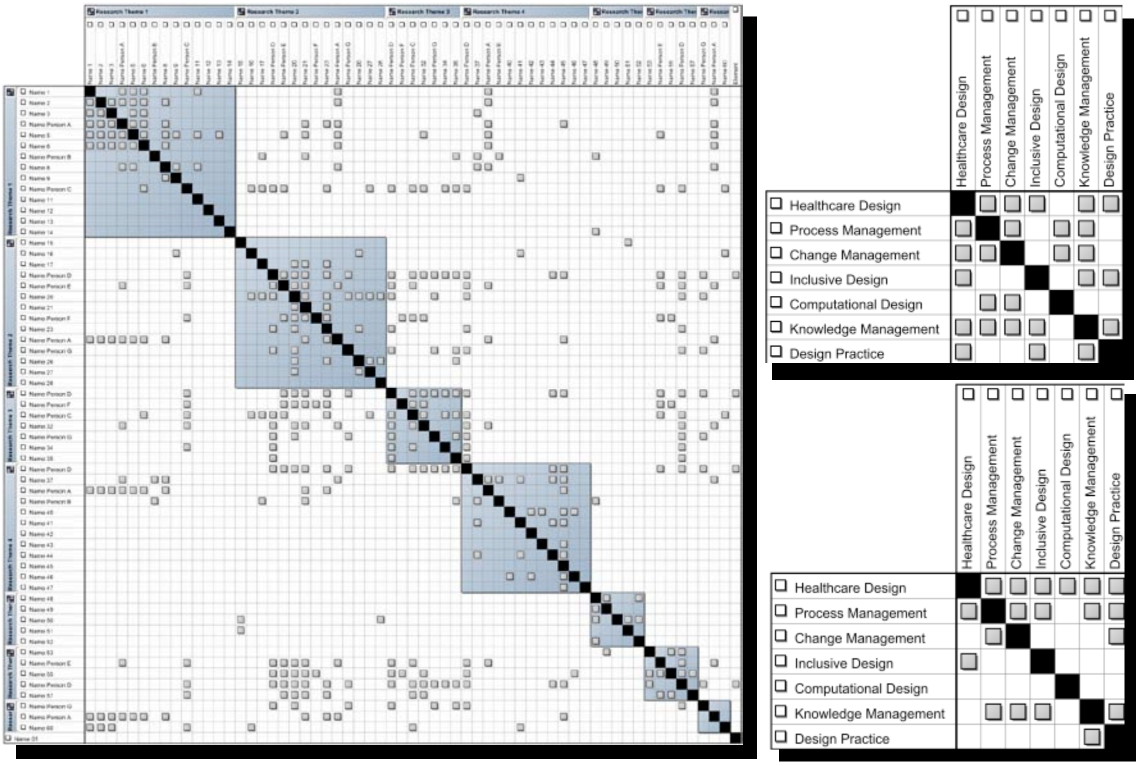

Figure 8: Three views of the communication flows in the EDC, between individuals elicited from survey (left), between themes elicited from survey (right, top) and elicited directly from senior researchers (right, bottom)

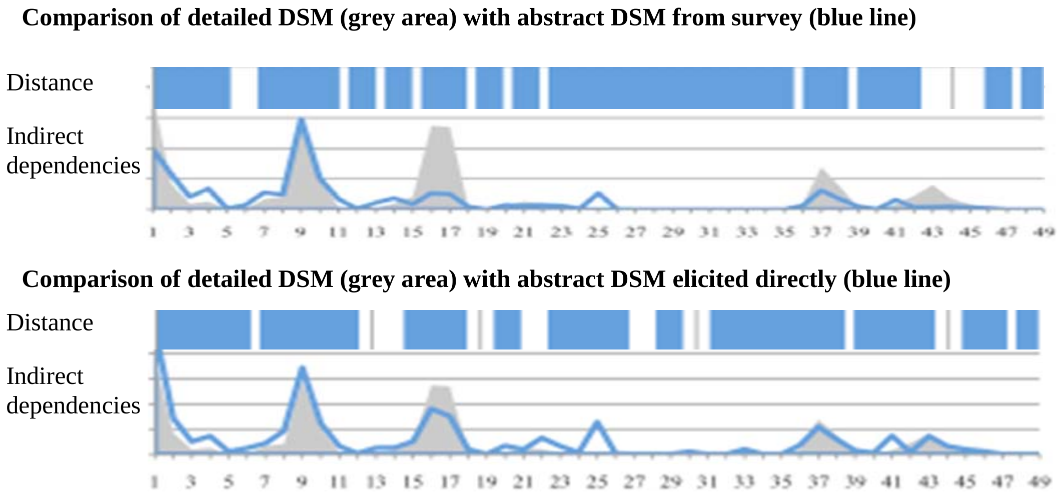

Figure 9: Comparison of detailed DSM to abstract-survey DSM (top) and SRA-DSM (below)

Comparison of the two matrices using distance metrics also highlights quite a number of mismatches. This suggests that the data on the two levels do not perfectly fit together. This suggests that there are still discrepancies that need revisiting if the two matrices are to agree with regards to this metric.

Considering the indirect dependencies, the graph suggests the two matrices match reasonably well. There are several issues that might be worth looking at, specifically cell numbers 1, 16, 17, 25, 37 and 43 as the difference is quite high in these.

In summary, the match between the data elicited through the EDC online survey at different levels of abstraction does not completely match, even though the same people have elicited it. Some aspects are quite good and suggest a strong basic match. Nevertheless, there are some inconsistencies especially with the minimum distance and the amount of indirect dependencies. A possible suggestion could thus be: certain relations need to be reconsidered on both levels of abstraction. To localise these, the metric *high amount of indirect dependencies* might need to be broken down into several areas of the whole detailed DSM to spot where exactly mismatches might be.

Comparison of detailed DSM to directly-elicited abstract DSM

The abstract DSM filled directly by the SRAs is not symmetric either; just as the detailed one. The reason is the same as with the detailed DSM and has been described earlier in this Subsection. The degree of connectivity seems of equal quality just as the distance matrix; the according graph shows less but larger inconsistencies. The matrix of indirect dependencies, however, shows a better matching abstract DSM. Especially cells number 1, 16, 17, 37 and 43 have improved. Cell number 25 turns out to be a worse match.

5.3 Summary of case studies

Product DSM. A product DSM is less complicated and more straightforward than an organisational DSM as less subjectivity and personal perception play a role. In this paper, the product DSM illustrates how the structural comparison metrics can differentiate between high- and low-quality data.

Organisation DSM: The metrics clearly show that, while the models have major consistencies, they are not totally alike. This is clearly shown by the presentation of the comparison along a single axis, which facilitates interpretation of the differences between values for particular cells. The data itself is highly subjective, as the symmetry constraint showed through indicating that the different participants had different views on the meaning of a strong or weak dependency. During the processing of survey data to create the DSMs, the data turned out to be affected by transcription errors. Some relations were missed, others which should not have existed were marked. Running the metrics resulted in an unusual looking set of graphs. Especially the metric *matrix of indirect dependencies* was able to locate where errors had been made; each error was considered to determine whether it could be easily explained and corrected. The case study seems to illustrate how comparison of models elicited using different means and from different perspectives, using structural metrics, can add value to the modeller and can supply her/him with additional information and insights. These insights can be used to highlight subjectivity as well as potential mistakes in transcription and other sources of error.

6 DISCUSSION

The results suggest that it is possible to raise the quality of data during the elicitation process by taking different views and perspectives of the same system. The comparison of consistency between the two or more levels, using structural metrics, allows insights about the data quality that has been elicited. Interestingly, it seems possible to gain insights that can help improve data quality, even without considering the real-world implications of the metrics that are discussed.

The approach outlined in this paper is only a starting point which aims to highlight the potential for using structural comparisons to assist in data acquisition for the DSM. Clearly, there are many opportunities to improve the analysis which has been outlined here, and systematise it as a paper-based method or even as a process embedded in a DSM software tool. In terms of the theory, a key aspect that needs attention is the multitude of different structural aspects that could be considered. Even if one structural aspect suggests an inconsistency between the matrices, the elicitation could still be correct because data is lost in comparing abstract and detailed models. Likewise, different metrics would most likely suggest different disagreements or different levels of importance for particular disagreements. The approach discussed in this paper only considers binary DSMs. Many of the methods could be adapted relatively easily for binary MDMs; however additional issues arise when considering DSMs or MDMs containing information about the dependencies, such as their strength.

Finally, it is important to highlight that the approach can only pick up inconsistencies between different perspectives. In the event of the same error occurring on both levels of abstraction the metrics will not work. The approach also cannot help distinguish which is the ‘correct’ value, when multiple models are in disagreement. However, by pointing out the potential discrepancy, we propose that structural comparisons may help modellers focus their efforts and result in better-quality models.

7 CONCLUSION

The DSM is a useful technique to gain insights into a complex system, such as a product, process or organisation. However, the quality of data in the DSM can significantly affect the quality and believability of insights gained through study of that model. Various methods have been proposed to assist with DSM knowledge elicitation, aiming to reduce effort of acquisition or improving quality of models by considering and comparing data acquired from different levels of abstraction. This paper has proposed and illustrated an extended approach in which the structural importance of disagreements between perspectives of a system is considered. We argue that highlighting the structural importance of disagreements can help focus the modeller's attention on those potential errors which may have most impact on the structurally-oriented analyses for which DSMs are often used – such as modularisation and simulation. Initial application of the ideas to two realistic DSM-modelling case studies seem promising, but much further work is required to systematise the proposed method.

REFERENCES

[1] Lindemann, U.; Maurer, M.; Braun, T.: Structural Complexity Management – An Approach for the Field of Product Design. Berlin: Springer 2009.

[2] Biedermann, W.; Strelkow, B.; Karl, F.; Lindemann, U.; Zaeh, M.: Reducing data acquisition effort by hierarchical modelling. In: Proceedings of the 12th International Dependency and Structure Modelling Conference, Cambridge, UK, July 2010. Munich: Hanser 2010.

[3] Wynn, D.; Kreimeyer, M.; Eben, K.; Maurer, M.; Clarkson, J.; Lindemann, U.: Proceedings of the 12th International Dependency and Structure Modelling Conference, Cambridge, UK, July 2010. Munich: Hanser 2010.

[4] Ariyo.: 'Change propagation in complex design: predicting detailed change cases with multi-levelled product models', PhD-thesis, Cambridge University Engineering Department, 2007.

[5] Jarrett, T.: 'A model-based approach to support the management of engineering change', PhD-thesis, Cambridge University Engineering Department. 2004.

[6] Eichinger, M.; Maurer, M.; Pulm, U.; Lindemann, U.: Extending Design Structure Matrices and Domain Mapping Matrices by Multiple Design Structure Matrices. In: Proceedings of the 8th Biennial Conference on Engineering Systems Design and Analysis (ASME-ESDA06), Torino. Torino, Italy: ASME 2006.

[7] Kreimeyer, M.: A Structural Measurement System for Engineering Design Processes. Dissertation, Technische Universität München, 2010. München: Dr-Hut 2010.

Contact:
Dr. David C. Wynn
University of Cambridge
Engineering Design Centre
Trumpington Street
Cambridge, CB2 1PZ, UK
Phone +44-1223-748565
Fax +44-1223-332662
dcw24@cam.ac.uk
http://www-edc.eng.cam.ac.uk/people/dcw24.html

Steffen Schmitz was a student at the Technische Universität München, Germany, and graduated in 2010. He focused his studies on product development and structural complexity management. He is now working as a management consultant for companies with complex technical products.

David C. Wynn is a Senior Research Associate at the Cambridge University Engineering Department. His research focuses on computer-based modelling of complex collaborative processes, such as engineering design.

Wieland Biedermann is a scientific assistant at the Technische Universität München, Germany, and has been working at the Institute of Product Development since 2007. He has published several papers in the area of structural complexity management.

P. John Clarkson received his PhD from Cambridge University and worked at PA Consulting before returning to Cambridge. He was appointed Director of the Engineering Design Centre in 1997 and Professor of Engineering Design in 2004. His interests are in the general area of engineering design.

Udo Lindemann is a full professor at the Technische Universität München, Germany, and has been the head of the Institute of Product Development since 1995, having published several books and papers on engineering design. He is committed in multiple institutions, among others as Vice President of the Design Society and as an active member of the German Academy of Science and Engineering.