Supporting Documentation for the Comment Response Record
Entropy Measures for System Identification and Analysis

Joseph J. Simpson, Mary J Simpson
System Concepts LLC

Reviewer comments associated with this paper to date focus on two primary areas:

1. General comments regarding design structure matrix (DSM) methods, and
2. Specific comments related to the entropy measures calculations and interpretations.

The comment response record is based on this supporting document, as the depth and nature of the reviewer comments necessitates a more detailed written response.

General reviewer comment (1) associated with DSM

"Although Steward coined the term DSM, his work deals exclusively with temporal (process) models; he never does clustering."

To address this comment, a definition of clustering is provided, and portions of Stewards work are addressed. The authors limit their comments to the parts of Steward's work that provide substantial motivation for the authors approach. The authors believe that an expansive discussion of issues in this area is important to fully express the methods, objectives and purpose. The definition of clustering will be addressed first.

The definition of clustering is taken from "Clustering Algorithms", J. A. Hartigan, 1975, p.1. "Clustering is the grouping of similar objects." Hartigan considers only two clustering structures in his text: partitions and trees. Hartigan (p. 11) also defines cluster: "a cluster is defined as a subset of a set of objects." These two general definitions provide the foundation for a wide and varied application of clustering activities, methods, and approaches.

The motivation for Stewards work, described in his book, *Systems Analysis and Management: Structure, Strategy and Design*, 1981 [p. Preface – first page in book], is quoted below:

> *"Some years ago, while working on the design of a nuclear power plant, we realized that by rearranging an iterative computing scheme we could obtain a better design in a few computer minutes than we had previously been able to obtain after many computer hours. This was a significant insight at the time. It occurred to me that we might study the structure of systems and flow of information during their design so that we could arrive at such insights in a more systematic way."*

The citation above describes the system selection metrics, and operational objects, that motivated Steward to create the DSM approach.

There are three distinct phases in Steward's development of DSM.

- **Phase One, pre-DSM** - contained no structured search for systematic improvement of the model design.
- **Phase Two, DSM development** - included detailed systems analysis and measurement to identify techniques that created acceptable computer performance as well as acceptable system (reactor) design characteristics.
- **Phase Three, Enhanced-DSM** - wherein well-defined DSM methods, practices, and approaches are integrated with other system analysis techniques to create systems (total systems) that achieve the required global, system-level metrics.

Each phase of this DSM development will be briefly discussed, with an emphasis on the authors' view of global, system-level activities and local, system-component level activities.

Phase One, pre-DSM

During Phase One of Steward’s DSM development, the system (reactor) design process was created by engineers using standard engineering practices that allowed a wide variation in the system-level metrics of (1) computer execution time and (2) design acceptability. These two global system metrics provided for the creation of four distinct groups.

Group One has bad computer execution time, and bad total system design.

Group Two has good computer execution time, and bad total system design.

Group Three has bad computer execution time, and good total system design.

Group Four has good computer execution time, and good total system design.

The general process and grouping outline in shown in Figure 1.

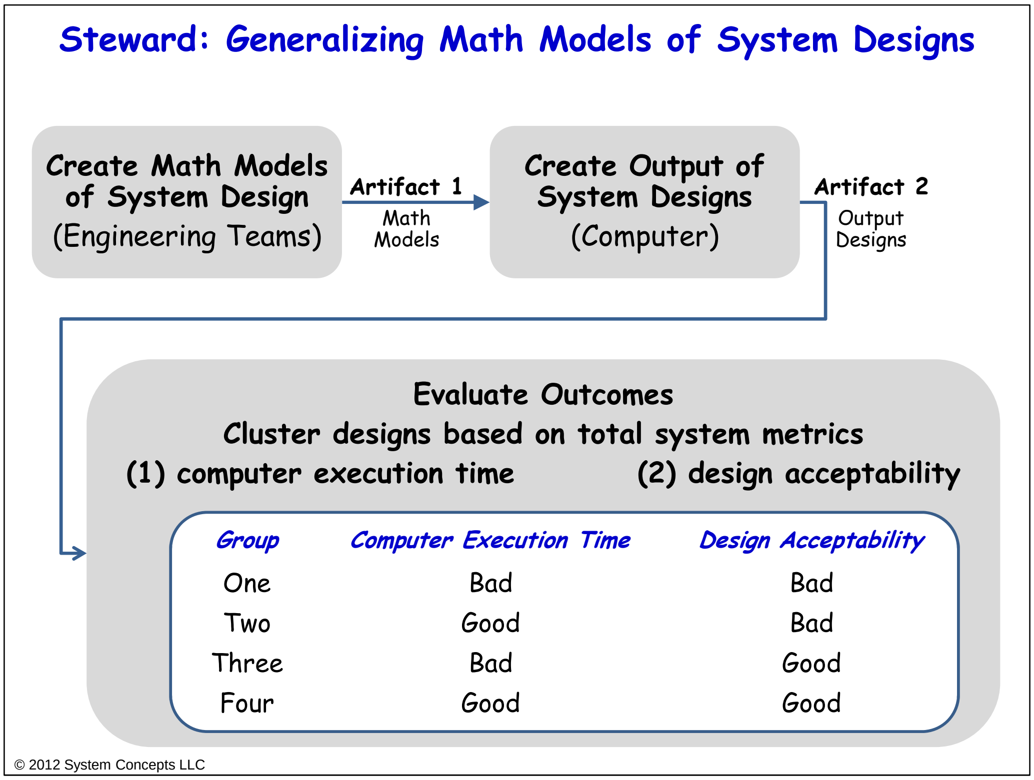

Figure 1. Steward’s Phase One Clustering

During Phase One of DSM development, Steward was clustering the prepared model designs based on the two criteria of (1) computer execution time and (2) total system design acceptability. Both of these metrics are global, system-level metrics. The authors believe that Steward was using clustering techniques to create groupings of model and design types. While the mathematical model and computer output design artifacts may describe temporal processes, they are themselves static items with a specific static computer execution time, and a specific design acceptability rating.

Phase Two, DSM development

During Phase Two of Steward’s DSM development, the total system (reactor) design model development process was evaluated to identify characteristics of the specific, component-level models and processes that impacted the global, total system metrics. The component-level, or local, knowledge and information required for Phase Two was centered in two areas: knowledge of the total system (reactors), and knowledge about the model development process. The global-level metrics for total system design for Phase Two were the same as those developed for Phase One: computer execution time and design acceptability.

Steward's DSM local model evaluation and analysis techniques, including partitioning and tearing, were combined with other local-level, graphical analysis techniques to create a body of knowledge used to impact the total-system-level metrics in a positive manner. Steward pointed out that there were many similar analysis techniques associated with DSM, including Interpretive Structural Modeling by John N. Warfield. In Phase Two, the system (reactor) design process could be accomplished by engineers in the same manner that was employed in Phase One. If the Phase One design processes were used exclusively, then the total system (reactor) design acceptability could be low, the cost of acceptable system designs could be high, and there was a very small chance of a highly capable system design. DSM techniques could be used in Phase Two to improve the total system (reactor) design acceptability and reduce the cost of the system design development. In Phase Two, it is clear that there were two levels of system design measurement and analysis: local techniques and global metrics. The local techniques were strongly associated with the DSM partitioning and tearing activities. The global metrics were directly tied to the total system characteristics of computer time and design acceptability. The Phase Two process is shown in Figure 2.

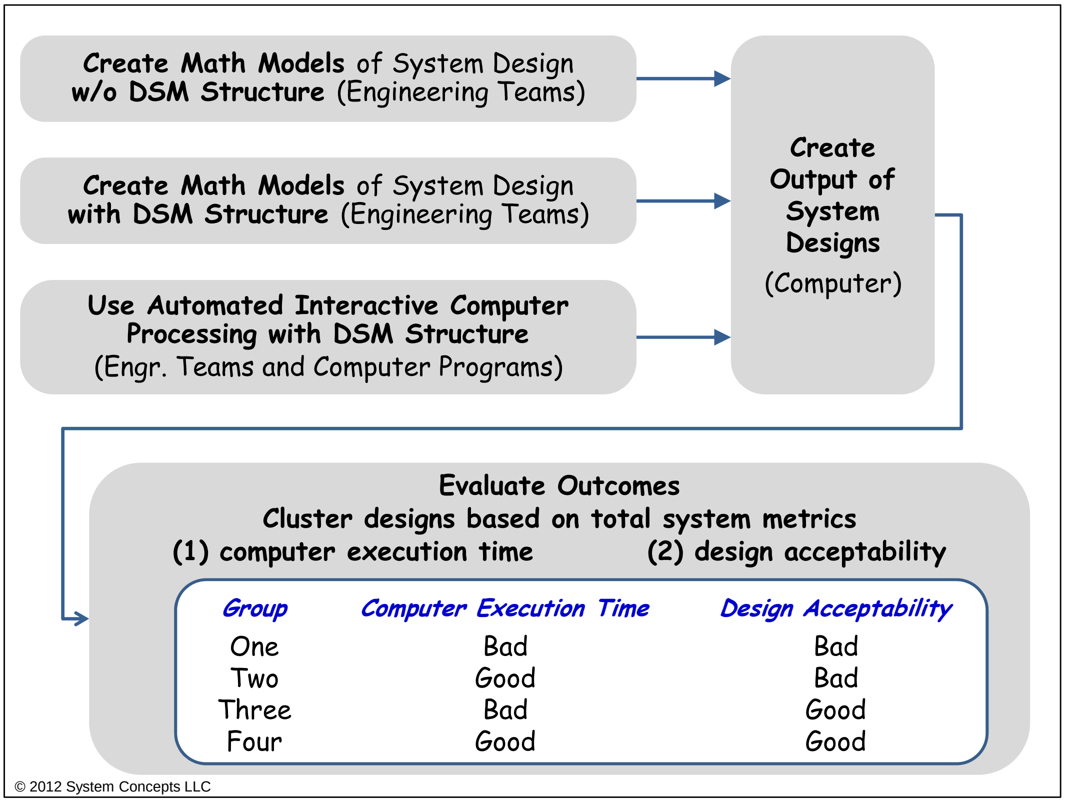

Figure 2. Phase Two, DSM Development Process

Phase Three, Enhanced-DSM

Phase Three of DSM development includes the incorporation of a number of new and unique approaches. Many of these new approaches are organized around domain application areas and general processes. In this response, the authors focus on the application of artificial intelligence and evolutionary computation techniques. Figure 3 shows a general outline of the types of solution processes and approaches that are available in Phase Three.

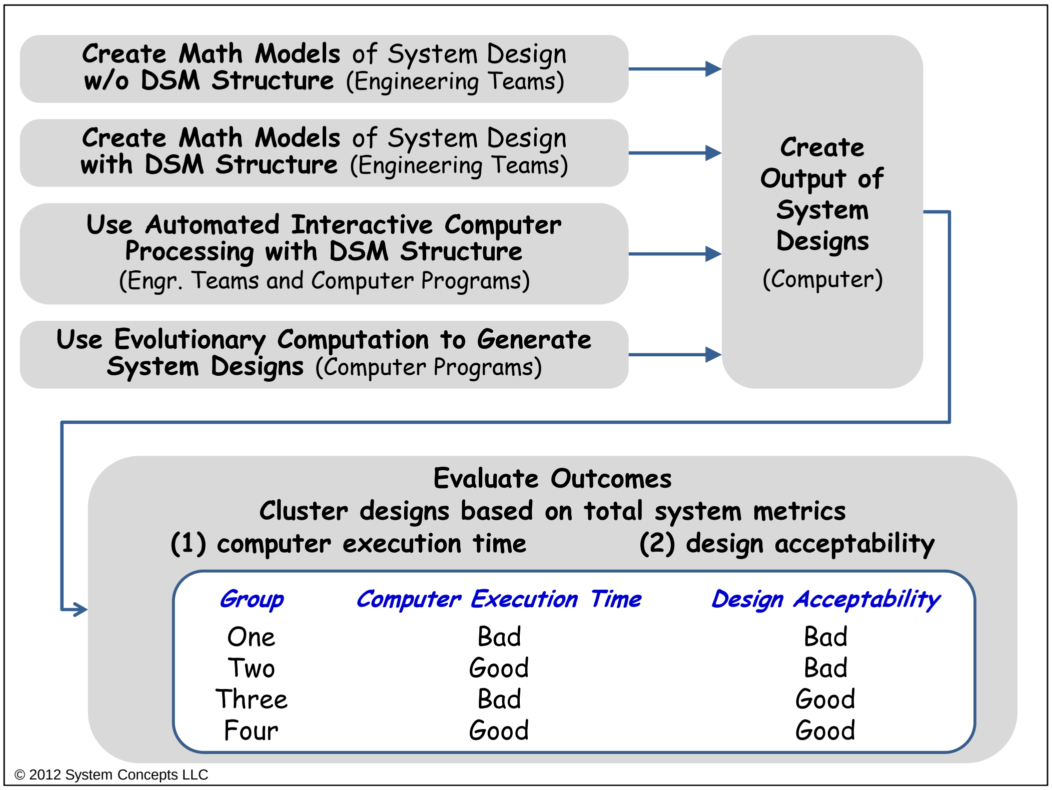

Figure 3. Phase Three, Enhanced DSM Process

Evolutionary computational techniques can be applied at the global system level, at the local process development level, or a combination of both local and global system levels. This definition of two levels of model development provides for a more detailed discussion of the DSM process application. The authors believe that Steward applied clustering techniques at the global level. The goal of the abstract relation type (ART) evaluation approach is to establish a computational-based system that can produce the same local system-component level results produced by manual and/or computer-assisted application of classical DSM techniques. These local-level (system component) results then impact the global (total system) metrics in the same manner. In using an automated computational approach, the authors believe that the cost of acceptable, advanced total system design can be reduced.

The authors' research, definition, development, and presentation of the abstract relation type (ART) evaluation approach are not limited to DSM. ART has also been applied to N-Squared Chart and automated N-Squared Chart techniques. In the fifty plus years since Steward developed the DSM approach, vast computing power has become a ubiquitous, readily-available resource. The authors' goal is to use this vast pool of computational capability to generate improved *total system designs* with minimum or no human involvement at the local system-component analysis and evaluation level to maximize global metrics and minimize engineering cost.

General reviewer comment (2) associated with DSM

"While the term DSM has grown to encompass static models (which are usually clustered), this is not due to Steward."

This comment is addressed from two perspectives: (1) the growth and types of DSM models, and (2) the domain and model characteristics considered key by the authors to support the successful development of fully-automated, total system, and local level (system-component) designs.

The concepts, ideas and methods associated with the term DSM are dynamic and growing. Many defining characteristics of a DSM approach are domain and application specific. As outlined in "Structural Complexity Management", by Udo Lindemann, Maik Maurer, and Thomas Braun, matrix methods have grown to include intra-domain, inter-domain, combined intra-domain and inter-domain as well as multiple-domain matrix approaches. More detail about DSM development is provided in Design Structure Matrix Methods and Applications", by Steven D. Eppinger and Tyson R. Browning. This work groups the current DSM applications into models for product architecture, organizational architecture, process architecture, and multi-domain architectures. Further, DSM models have been classified as static or time-based.

The abstract relation type (ART) was developed, in part, to address the agile growth and application of the DSM methods enumerated above. The ART approach directly uses domain and model characteristics considered key in system development: that is the organizing system relation that creates and structures the system of interest over a set of objects. Each ART description also has a prose description (formal and informal), a graphical representation, and a mathematical component. The authors believe that the ART approach identifies and emphasizes the type of system information needed to support the total automation of a given problem solution process.

Steward evaluated, extended and applied the Interpretive Structural Modeling (ISM) techniques of Warfield. ISM employs a number of mathematical evaluation and analysis techniques to produce a static structural representation of a problem space. The exact types of methods that were employed by Steward in the modification and extension of the ISM approach would be an interesting area to explore. The authors generally agree with the concepts of static, time-based, temporal flow, and multi-domain DSM categories, but believe that there are other categories of model, domain and operational information that can be more effectively used to support the automation of the problem-solving process.

General reviewer comment (3) associated with DSM

"You should compare your approach with the published clustering approaches."

This comment is addressed from three perspectives: (1) authors' publications regarding published clustering approaches, (2) need for standardized matrix examples, and (3) proposed set of matrix examples.

Authors' Publications regarding Published Clustering Approaches

The authors have published papers using examples from the published literature, while observing that some of the previously published work by other authors contained apparent errors. There are many steps in the publication process where errors can be injected into design information that is published for review. These errors can prevent the verification and validation of the reported results. Though there are other examples, one paper is cited as an example of the difficulty encountered in evaluating and comparing detailed system information using a hardcopy paper format: "Characterizing Complex Product Architectures," by D.M. Sharman and A.A. Yassine, in *Systems Engineering Journal*, Vol 7, No. 1, pp. 35-60, 2004.

Figure 1 from their paper showed a partitioned DSM model. The base DSM shown in Figure 1 of their paper presented the same system in two different structural forms: a base model and a partitioned model. Upon close examination of the graphical representations for the base and the partitioned models, it became clear that the base model and the partitioned model contained a different number of system dependencies and/or interfaces. Before a comparison could be performed, questions related to which graphical representation was correct needed answering. As a result, an evaluation and detailed analysis of both representations was needed prior to any comparison of technique.

Their Figure 2 showed the outcome from their DSM clustering technique. That outcome was exactly reproduced and verified by the ART evolutionary computational approach. The authors' publication, "Systems of Systems Complexity Identification and Control" presented at the 2009 IEEE System of Systems Conference provides additional details.

Need for Standardized Matrix Examples

To address the lack of standard matrix examples, the authors have developed a range of standard matrix examples. These standard examples are used to support the analysis of N-Squared Charts (N2C), Automated N-Squared Charts (AN2C), and Design Structure Matrices (DSM) – all of which are standardized system analysis and evaluation methods used in the discovery, design and evaluation of systems.

A common theme among these three methods (N2C, AN2C and DSM) is the concept of system object and interface (connection) clustering. The general concept of arranging system objects along the matrix diagonal, in a manner that concentrates the system interfaces (or connections) around the matrix diagonal, is one of these common themes. While each method has varying specific interpretation rules and definitions of the clustering activity, the general clustering activity is common to all of these methods. Clustering is defined as the grouping of similar objects.

There are manual and automated clustering methods. The standard matrix examples present a set of base models in unstructured form. Select individual base models will also be provided with an example in clustered or partitioned form. The matrix pairs (base and structured) are used to verify a given automated analysis approach. These pairs of matrices are designed to be standard examples for use in benchmarking and comparing differing clustering or partitioning techniques. The single base model examples are provided for testing and verification of methods. A process for checking the best known values for these base model matrices will be established. These standard examples are organized around basic matrix attributes and basic matrix evaluation approaches.

Proposed Set of Matrix Examples

At the most basic level, matrix clustering or object grouping can be divided into three types. The first type of matrix clustering focuses the interfaces along the matrix diagonal. The second type of matrix clustering focuses the interfaces in the lower triangular section of the matrix. The third type of matrix clustering focuses the interfaces in the upper triangular section of the matrix. These three basic types are the first to be addressed by the standard matrix example set.

Each standard matrix has a size N, establishing an N by N square matrix. Each standard matrix has a specific number of connections. The “percent-full” gives an indication of the matrix interface density. “Percent-full” is found by dividing the number of specific connections in the matrix, by the number of possible connections that can be made. A matrix that is 100 percent full will not respond well to many types of analysis and evaluation. As described in the working paper, a matrix with a minimum number of interfaces (N-1) is also expected to perform poorly under many types of analysis and evaluation. The best results for automated analysis and evaluation are expected from matrices that have a percent-full rating of around 20 to 50 percent. One use being made of the set of standard matrix examples is testing of these expectations.

Assuming a matrix that has an object (or named entity) along the diagonal, the remaining matrix cells can have an entry in each row or there can be rows in the matrix with no entries. This basic difference in matrix structure (empty rows or no empty rows) also impacts matrix methods and evaluation approaches. The first set of standard examples addresses matrices with an object along the diagonal, and rows that have at least one entry in each row.

A naming convention has been developed to clearly identify each type of standard matrix. Each matrix name has the following components:

Name base for standard matrix example, SME,

> ***Analysis type***, for example: clustering-diagonal:*CD*, clustering-upper-triangular:*CU*, clustering-lower-triangular:*CL*
>
> ***Matrix size N***, for example: 16,
>
> ***Interface percent-full***, for example: 20 percent, 20P
>
> ***Empty rows or no empty rows***: no empty rows, NER
>
> ***Configuration type, ordered or disordered***, for example disordered, DO

This results in a complete name of SME_CD_16_20P_NER_DO

The first set of standard matrix examples will be developed for the following sizes:

9 x 9
16 x 16
25 x 25
36 x 36
49 x 49
64 x 64
81 x 81
100 x 100
121 x 121
144 x 144

The minimum interface percent-full is:

9 x 9 = 8/72 = 11.1 percent
16 x 16 = 15/240 = 6.25 percent
25 x 25 = 24/600 = 4.0 percent
36 x 36 = 35/1260 = 2.8 percent
49 x 49 = 48/2352 = 2.0 percent
64 x 64 = 63/4032 = 1.6 percent
81 x 81 = 80/6480 = 1.3 percent
100 x 100 = 99/9900 = 1.0 percent
121 x 121 = 120/14520 = 0.8 percent
144 x 144 = 143/20592 = 0.7 percent

The complete set of standard matrix examples is composed of the following percent-full interface values for each matrix size:

5 percent (except for 9 x 9 and 16 x 16 based on the previous “minimum interface full values”)
10 percent (except for 9 x 9 based on the previous “minimum interface full values”)
15 percent
20 percent
25 percent
30 percent
35 percent
40 percent
45 percent
50 percent

Each size of matrix will have an example for each of the percent-full interface values. The complete set of standard examples for the described diagonal clustering is then *approximately* 100 matrix pairs (one base and one clustered or partitioned), for a total of 200 matrices. In a similar fashion the upper triangular will have 200 matrices and the lower triangular clustering example sets will have 200 matrices. This comprises a total of around 600 matrices for the complete basic standard matrix set with no empty rows.

When matrices with empty rows are added to the above basic standard set, the expanded standard matrix set contains 1200 matrices. If the static and time-based matrix representations are addressed, that would create a standard matrix example set of 2400 matrices.

These matrices will be developed, published and discussed in an incremental fashion to encourage the development, comparison and discussion of common, verifiable matrix clustering algorithms and methods. The smaller-sized matrices are expected to perform well at all listed percent-full values. The larger matrices may not perform well at the higher percent-full values; and smaller percent values may need to be generated.

General reviewer comment (4) associated with DSM

"… compare your approach with the published clustering approaches. One that uses evolutionary computation is attached."

Note: The reviewer attached a copy of "An information theoretic method for developing modular architectures using genetic algorithms", by Tian-Li Yu, Ali A. Yassine and David E. Goldberg. This paper was published in *Res Eng Design* (2007) 18:91-109 on August 10th 2007.

The provided paper contains three figures that present original (disordered) and clustered (ordered) system representations. These three figures are: (1) Figure 1, page 94, (2) Figure 9, page 99, and 3) Figure 11, page 101. The paper contains other examples of DSM clustering, but none of the other examples appear to provide the required "disordered – ordered" pair to support evaluation and comparison of different evolutionary computational techniques. The methods, procedures and algorithms presented in the provided paper are significantly different than the techniques used by the authors. The system representations provided in each of these three figures are discussed in the following sections. The authors have limited their discussion to the graphical representations of these figures, rather than including comments related to the text of the paper.

Figure 1, Page 94

Figure 1 on page 94 of the published work presents a 7 x 7 DSM matrix in three different configurations, labeled a, b, and c. Configuration (a) is the original DSM. Configuration (b) is a clustered DSM. Configuration (c) is an alternative clustering of the original DSM.

The authors' general evaluation procedure has a set of initial screening activities. The first screening activity is to check the graphical representation of the given system. Using the construction-rule definition of a system, "a system is a relationship mapped over a set of objects," the authors check the system configurations for "system equality." In this case, each of the system's graphic representations of the clustered DSMs in Figures 1(b) and 1(c) should contain the same number of objects (represented along the diagonal), and the same number of entries (or marks) showing interfaces or connections, as the original DSM in Figure 1(a). The system presented in Figure 1 fails this first screening validation test. The original DSM presented in Figure 1(a) has *7* objects and *19* marks. Each of the clustered DSM configurations presented in Figure 1(b) and 1(c) has 7 objects and *18* marks – which represent different systems than the one presented in Figure 1(a). The authors assume that an error was made in Figure 1(b) and 1(c), when the mark in "row A – column B" was omitted. If the mark in "row A – column B" is inserted into each of the clustered DSM configurations in Figure 1(b) and 1(c), the authors evolutionary computation technique generates the same configuration as the corrected configurations. Due to the general nature of these configurations, all clustering is considered notional.

Four of the authors' clustered matrix alternatives, generated by an evolutionary computing approach, are shown in Figure 4.

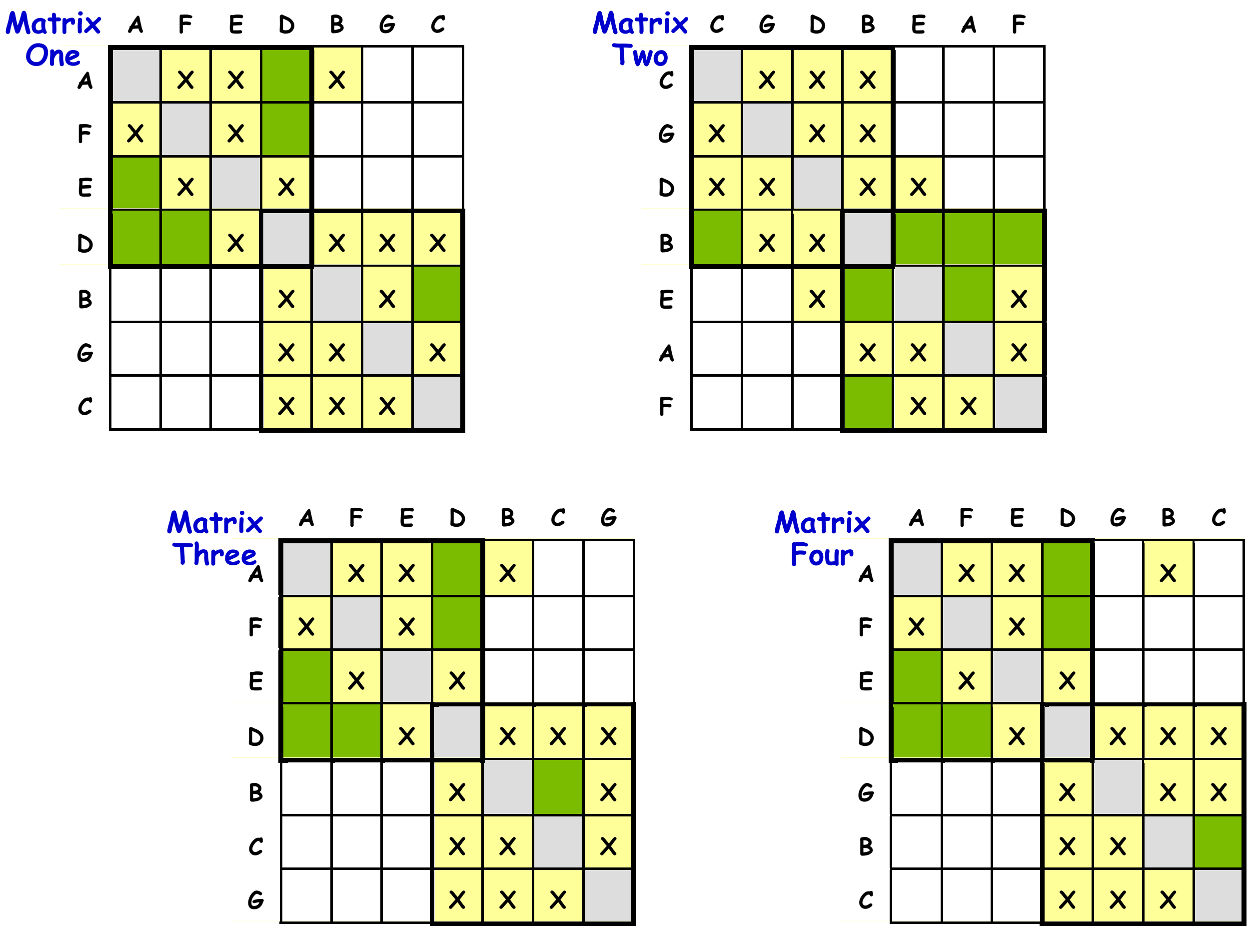

Figure 4. Four Valid Alternatives for Further Analysis of Figure 1(a)

Figure 9, Page 99

Figure 9 on page 99 of the published work presents a 7 x 7 DSM matrix in three different configurations. Configuration 9(a) is the original (disordered) DSM. Configuration 9(b) is a clustered DSM. Configuration 9(c) is an alternative clustering of the original DSM. The system presented in Figure 9 passes the initial system equality test, since each system configuration (9(a), 9(b), and 9(c)) contains 7 objects and 14 marks. The authors' evolutionary computation technique generates the same system configurations as those presented in Figure 9. . Due to the general nature of these configurations, all clustering is considered notional.

Four of the authors' clustered matrix alternatives, generated by an evolutionary computing approach, are shown in Figure 5.

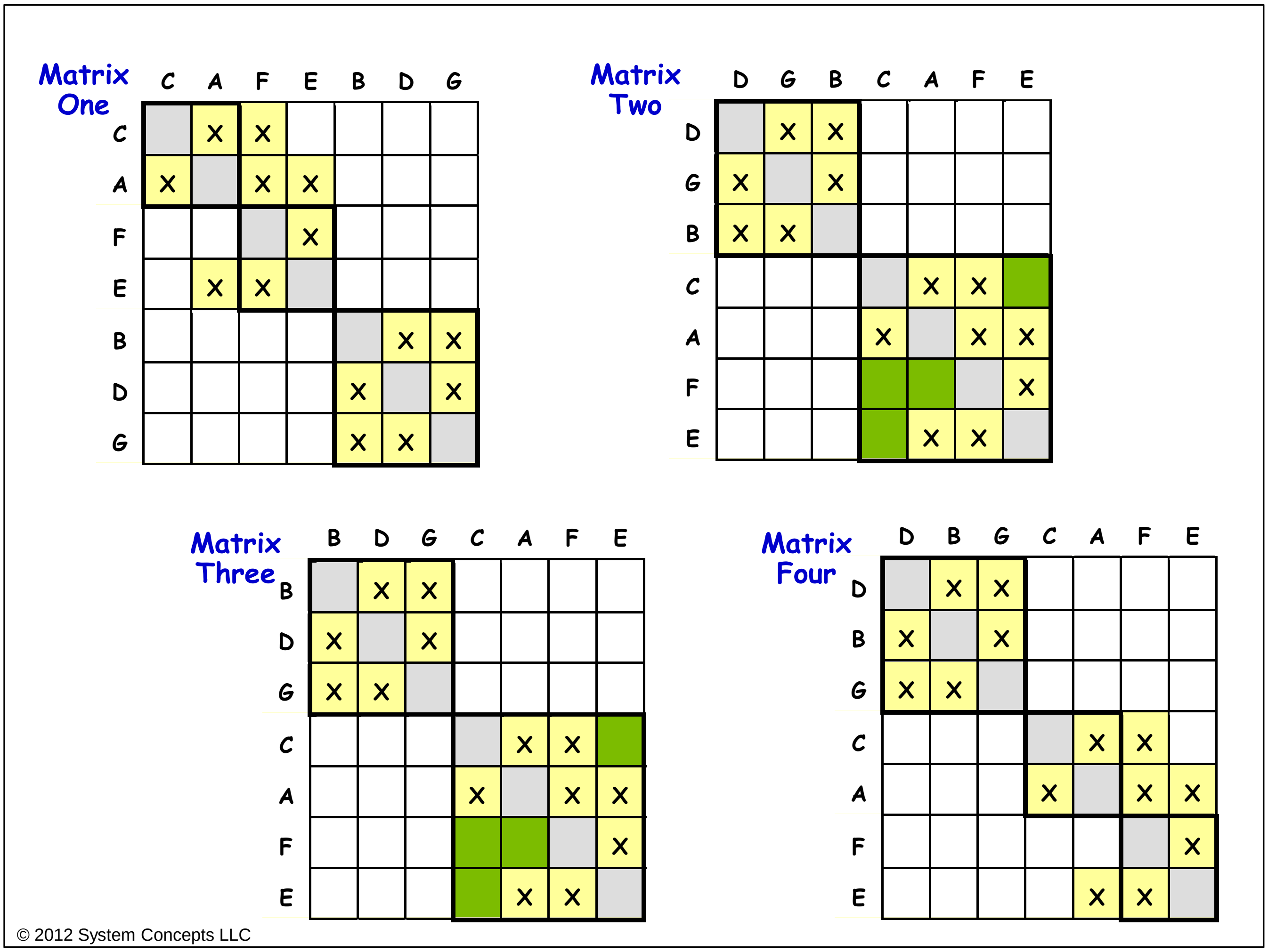

Figure 5. Four Valid Alternatives for Further Analysis of Figure 9(a)

Figure 11, Page 101

Figure 11 on page 101 of the referenced publication presents four different system configurations, each with an input (disordered) configuration and an output (ordered) configuration. Each of the matrices in these four system representations are a 9 by 9 matrix. The examples are numbered Figures 11(a), 11(b), 11(c), and 11(d).

Figure 11(a)

Again using the construction-rule definition of a system, “a system is a relationship mapped over a set of objects,” the authors check the system configurations for “system equality.” In this case, although Figure 11(a) Input and Figure 11(a) Output, both have nine (9) objects and 18 marks each, there are two (2) of the objects have no marks associated with them. Two objects that lack any relationship or mapping marks are not considered part of the system. Thus, Figure 11(a) fails the first system screening test.

Within the authors’ general evaluation procedure, the second system screening test is an evaluation of information content associated with the matrix and model components. Using Shannon's Information Theory, as explained in the authors’ original paper, *Row D, Row I, Column D* and *Column I* ***contain no information***.

At this point in the authors’ evaluation process, the matrix components are clustered into two groups; group one are the matrix components that contain no information, and group two are the matrix components that contain information. *Only the 'information-containing' components* of group two matrix components are used by the authors in their evaluation process. The elimination of system and matrix components that do not contain any information reduces the computational complexity of the system clustering activity. This is one method the authors use to reduce complexity. The remaining 7 by 7 matrix, whose components contain information, is then evaluated. Four clustered (ordered) matrices were selected for presentation as examples of the authors' technique. The automated generation of these alternatives for system clustering provides substantial advantages by 1) reducing the cognitive complexity associated with the original system, and 2) presenting a set of valid alternatives for further engineering analysis. . Due to the general nature of these configurations, all clustering is considered notional.

Four of the authors' clustered matrix alternatives, generated by an evolutionary computing approach, are shown in Figure 6. Each of these alternatives has a different label set than the original paper because of the removal of the "empty rows and columns." These matrix alternatives may be combined with empty rows and columns in a vast number of ways without impacting the final system information content.

Matrix One

	E	G	B	C	A	F	D
E		X	X	X			
G	X		X	X			
B	X	X		X			
C	X	X	X				
A						X	X
F					X		X
D					X	X	

Matrix Two

	A	D	F	G	E	C	B
A		X	X				
D	X		X				
F	X	X					
G					X	X	X
E				X		X	X
C				X	X		X
B				X	X	X	

Matrix Three

	A	D	F	G	E	B	C
A		X	X				
D	X		X				
F	X	X					
G					X	X	X
E				X		X	X
B				X	X		X
C				X	X	X	

Matrix Four

	F	A	D	C	E	G	B
F		X	X				
A	X		X				
D	X	X					
C					X	X	X
E				X		X	X
G				X	X		X
B				X	X	X	

Figure 6. Four Valid Alternatives for Further Analysis of Figure 11(a)

Figure 11(b)

Using the construction-rule definition of a system, "a system is a relationship mapped over a set of objects," the authors check the system configurations for "system equality." In this case, although Figure 11(b) Input and Figure 11(b) Output, both have nine (9) objects and 24 marks each, there is one (1) object that has no marks associated with it. An object that lacks any relationship or mapping mark is not considered part of the system. Thus, Figure 11(b) fails the first system screening test.

Within the authors' general evaluation procedure, the second system screening test is an evaluation of information content associated with the matrix and model components. Using Shannon's Information Theory, as explained in the authors' original paper, *Row I* and *Column I* ***contain no information***.

At this point in the authors' evaluation process, the matrix components are clustered into two groups; group one are the matrix components that contain no information, and group two are the matrix components that contain information. *Only the 'information-containing' components* of the matrix are used by the authors in their evaluation process. The elimination of system and matrix components that do not contain any information reduces the computational complexity of the system clustering activity. This is one method the authors use to reduce complexity. The remaining 8 by 8 matrix, whose components contain information, is then evaluated. Due to the general nature of these configurations, all clustering is considered notional.

Four of the authors' clustered matrix alternatives, generated by an evolutionary computing approach, are shown in Figure 7. Each of these alternatives has a different label set than the original paper because of the removal of the "empty row and column." These matrix alternatives may be combined with empty rows and columns in a vast number of ways without impacting the final system information content.

Matrix One

	D	B	G	C	E	A	F	H
D		X	X					
B	X		X					
G	X	X		X	X	X		
C			X		X	X		
E			X	X		X		
A			X	X	X		X	X
F						X		X
H						X	X	

Matrix Two

	F	H	A	C	E	G	D	B
F		X	X					
H	X		X					
A	X	X		X	X	X		
C			X		X	X		
E			X	X		X		
G			X	X	X		X	X
D						X		X
B						X	X	

Matrix Three

	B	D	G	E	C	A	F	H
B		X	X					
D	X		X					
G	X	X		X	X	X		
E			X		X	X		
C			X	X		X		
A			X	X	X		X	X
F						X		X
H						X	X	

Matrix Four

	D	B	G	C	E	A	H	F
D		X	X					
B	X		X					
G	X	X		X	X	X		
C			X		X	X		
E			X	X		X		
A			X	X	X		X	X
H						X		X
F						X	X	

Figure 7. Four Valid Alternatives for Further Analysis of Figure 11(b)

Figure 11(c)

Using the construction-rule definition of a system, “a system is a relationship mapped over a set of objects,” the authors check the system configurations for “system equality.” In this case, both Figure 11(c) Input and Figure 11(c) Output have nine (9) objects and 40 marks each, and all objects have marks associated with them. Figure 11(c) passes the first system screening test.

The second system screening test is an evaluation of information content associated with the matrix and model components. Using Shannon's Information Theory, as explained in the authors’ original paper, *Row G* and *Column G* ***contain no information***.

At this point in the authors’ evaluation process, the matrix components are clustered into two groups; group one are the matrix components that contain no information, and group two are the matrix components that contain information. *Only the ‘information-containing’ components* of the matrix are used by the authors in their evaluation process. The elimination of system and matrix components that do not contain any information reduces the computational complexity of the system clustering activity. This is one method the authors use to reduce complexity. The remaining 8 by 8 matrix, whose components contain information, is then evaluated. Due to the general nature of these configurations, all clustering is considered notional.

Four of the authors’ clustered matrix alternatives, generated by an evolutionary computing approach, are shown in Figure 8. Each of these alternatives has a different label set than the original paper because of the removal of the “non-information row and column.” These matrix alternatives may be combined with the non-information row and column in two basic ways without impacting the final system information content. A final matrix could be constructed from the group one and group two components.

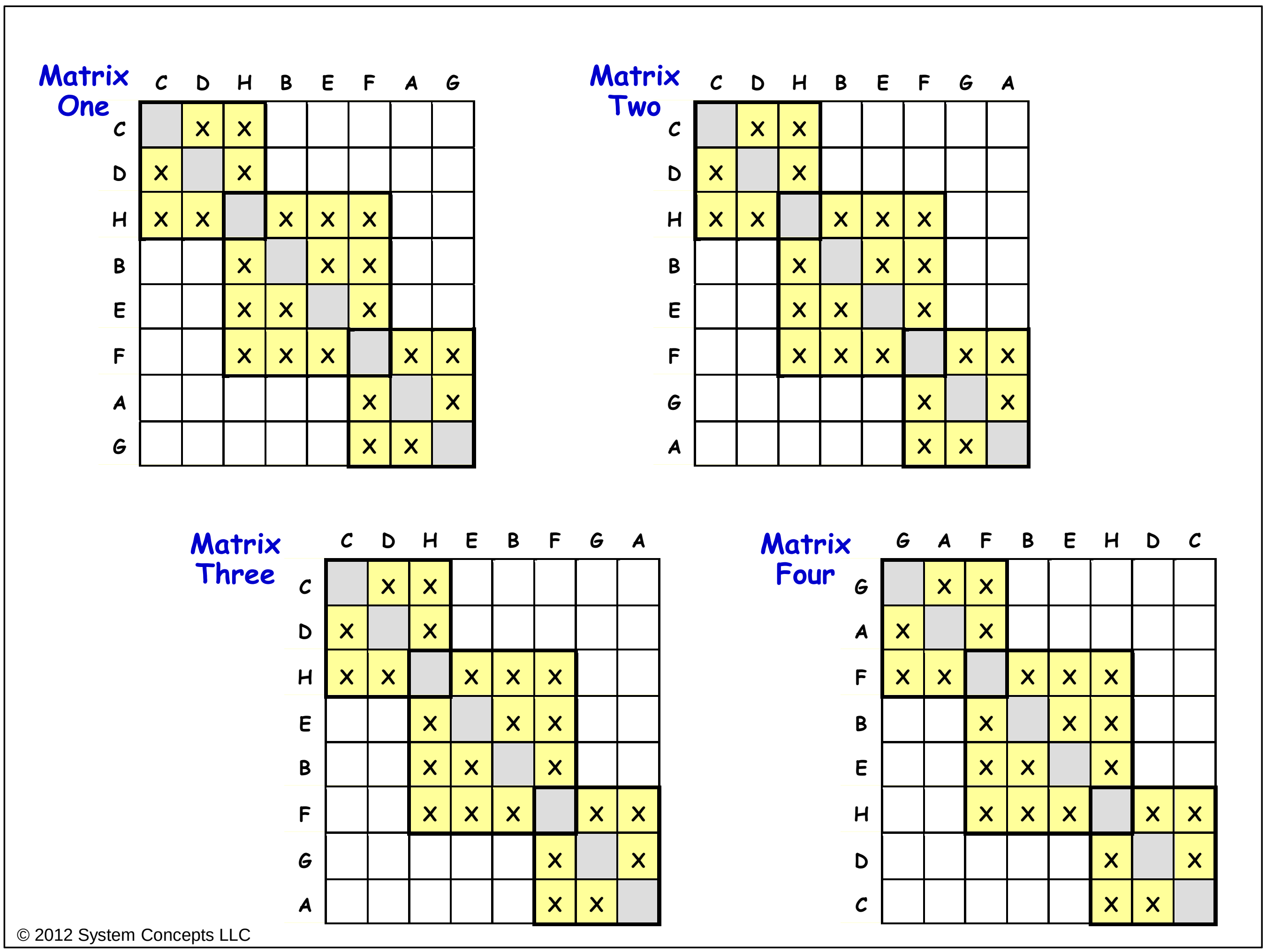

Figure 8. Four Valid Alternatives for Further Analysis of Figure 11(c)

Figure 11(d)

Using the construction-rule definition of a system, “a system is a relationship mapped over a set of objects,” the authors check the system configurations for “system equality.” In this case, both Figure 11(d) Input and Figure 11(d) Output have nine (9) objects and 36 marks each, and all objects have marks associated with them. Figure 11(d) passes the first system screening test.

The second system screening test the authors use is the evaluation of the information content associated with the matrix and model components. Using Shannon's Information Theory, as explained in the authors’ original paper, all rows and columns contain information. The authors classify this matrix as a special case of the “central clustering approach. While the authors have a range of operational central clustering routines they do not have a prepared operational approach for this special case. Therefore no outcome matrices will be shown at this time.

This analysis again emphasizes the need for a standard set of matrix examples that can be used to evaluate different processes and components. The authors believe that their evolutionary computation evaluation process offers substantive improvements for DSM practitioners as a result of its initial screening process and additional process clarity - attributes that reduce the computational complexity associated with automated evolutionary computation.

General reviewer comment (5) associated with DSM

“Be careful about putting too much faith in any result from an NxN model when the same system could also be modeled in greater detail (e.g., as a 2Nx2N model).”

The authors have used the “square matrix” approach because of the high rate of appearance in classical systems engineering, operations research, system design and systems science literature. The graphical representation of system information is well defined in many cases. The authors have processes and procedures for creating a square matrix from a combination of other different shaped matrices. Further, the authors have methods for decomposing a square matrix into other shaped matrix and graphical representations. Steward points out the equivalence between a structural matrix and a bipartite graph. Steward’s use and extension of the ISM developed by Warfield conveys a number of matrix methods and transforms that are used by the authors. Friedman's ‘Multidimensional Mathematical Model Management’ (M4) approach uses bipartite graphs and other matrix notation. The authors have shaped some of the M4 processes into a square matrix, and then decomposed these back to other matrix forms.

Issues related to the level of abstraction, level of detail and level of analysis are addressed by the ART approach. The authors are creating more examples of specific techniques and processes. The development of a standard matrix example set will greatly support this goal.

General reviewer comment (6) associated with DSM

“One of the big desires in this area of research is a “scale-free” measure that holds no matter what level of analysis is chosen by the modeler.”

The authors believe that the connection score and the object score provide the basis for “scale-free” metric development. System definition is independent of system size. Information definition is independent of system level and/or size.

General reviewer comment (7) associated with DSM

“The same system should not have different levels of entropy depending on the level of abstraction at which it is viewed (an observer-dependency).”

Since the authors included two very different types of entropy measures, physical and information, it is difficult to determine what the reviewer means by this comment. The authors will not respond to this comment at this time. This may be a concern that the reviewer would like to reframe.

General reviewer comment (8) associated with DSM

"A step in the right direction might be to normalize your metric as a ratio to N (matrix size), but even this is not enough, since the number of combinations does not grow linearly with N, so perhaps you will have to normalize against the maximum number of combinations in a matrix of size N."

Since the authors included two very different types of entropy measures, physical and information, it is difficult to determine what the reviewer means by this comment. Each of the metrics apply to a distinctly different part of the matrix. The authors will not respond to this comment at this time. This may be a concern that the reviewer would like to reframe.

General reviewer comment (9) associated with DSM

"However, the challenge then becomes that as a system is modeled at deeper levels, the matrix tends to get sparser."

The authors believe that the two metrics presented in the paper have the capability – when combined with the proper procedure – to guide the analytical process, and to help optimize the information content in these types of models.

General reviewer comment (10) associated with DSM

"Therefore, I'm not sanguine about any metric based solely on a binary DSM."

The authors agree. Indeed, the authors have developed two metrics, each based on a different part of a system (and associated with a different part of a matrix) to support the evaluation of large-scale systems.