APPLYING THE COLLAPSING DESIGN STRUCTURE MATRIX METHOD TO DEVELOP MILITARY ENTERPRISE SYSTEMS INTEGRATION PLANS

Michael Kretser, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, 937-255-6565, michael.kretser@afit.edu

Jeffrey Ogden, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, 937-255-6565, jeffrey.ogden@afit.edu

John Colombi, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, 937-255-6565, john.colombi@afit.edu

Paul Hartman, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, 937-255-6565, paul.hartman@afit.edu

ABSTRACT

Traditional static Design Structure Matrices (DSMs) have predominantly focused on product level problems with minimal use in enterprise-level challenges. This paper utilizes a new method of collapsing DSMs (C-DSMs) to illustrate enterprise information system (IS) redundancy in a representative U.S. Air Force logistics enterprise example. User constraints direct the C-DSM method which employs an algorithmic approach that automates the creation of a systems integration plan and rudimentary initial cost estimates.

INTRODUCTION

Government organizations such as the United States Air Force (USAF) and the Department of Homeland Security (DHS) have struggled with implementing ERPs that cover a large number of systems. The USAF tried to set a world record with 250 system integration ERP with Expeditionary Combat Support System (ECSS), and DHS has been charged with reducing the number of financial systems down from 500 [1]. After ten years and $1B spent the USAF was forced to cancel ECSS due to lack of progress and lack of a foreseeable conclusion to the integration effort [2]. Prior to cancellation of ECSS, the USAF commissioned the RAND Corporation to research ERPs to develop a better understanding on how to properly understand and implement an ERP. This report was conducted from 2011 to 2012, but was published in 2013 after the cancellation of ECSS. In RAND report "Improving Air Force Enterprise Resource Planning-Enabled Business Transformation" [3] the authors investigate the conditions for successful ERP implementation: the Business Case, Governance, Business Process Reengineering (BPR), Organization Change Management, and IT Acquisition.

The RAND report discusses five conditions, and two of these conditions can be addressed using the C-DSM method presented in this paper. The first condition, the Business Case, should be the "framework for cross-functional decision making" and the point at which functional areas connect. The Business Case is contingent upon a clear current "AS-IS" environment and the target "TO-BE" environment which achieves enterprise goals to include cost and performance [3]. The authors conclude that the USAF has struggled with enterprise business strategy and creating the "AS-IS" and "TO-BE" environments. The second condition, the "Governance" factor, is the decision making method that is employed to achieve a corporation's enterprise-focused goals. To employ this correctly the authors recommend a single responsible authority or small group which is a problem as the USAF does not leave officers in place long enough to oversee large programs from inception to completion. Officers have 2-3 years at best which is barely enough time to even plan for an ERP, much less oversee its

implementation. These conditions are further described as critical success factors for ERP implementation in other published research [4-5].

In an attempt to satisfy the Business Case and Governance conditions in the RAND report, the C-DSM method methodically drives integration independent of frequent leadership change. It produces transition plans that can be moderately modified (or left alone) by the current leadership, and will continue long after the leadership change has occurred; planning is no longer dependent on one or two key individuals. Developing the source matrices needed to employ the C-DSM method provides the current "AS-IS" enterprise structure and the output of the algorithm will provide iterations of decreasing size and complexity of the "TO-BE" states of the enterprise.

LITERATURE REVIEW

Given the problem of hidden, but systematically growing complexity in large enterprises, it was necessary to review current systems engineering methods. Furthermore, to develop an algorithm that would visually represent an enterprise through various stages of change, graphical methods were researched. Many methods were reviewed to include N2 diagrams, functional decomposition, block definition diagrams (SysML), and object/class diagrams (UML), but in the end design structure matrices (DSMs) were chosen [6]. DSM methodology was the tool of choice as it is a concise tool that can represent a large breadth of data points, with sufficient depth of detail, in a small graphical depiction that could also be mathematically manipulated.

Once DSMs were chosen to represent an enterprise's collection of software systems, a literature review of DSMs was conducted to examine the current state of the literature covering the cross-section of enterprise problems with DSM methods. It was found that there was significant use of DSMs in the product realm (697 works in one keyword search using "design structure matrix AND product"), several uses in the process and organizational architecture realm, some new methods covering the cross section of these realms (multi-domain matrices (MDMs))[7], but a significant dearth in the employment of DSMs in the enterprise realm exists (only 10 in one review). Furthermore, there was a gap in the literature where DSMs are used to prescribe deletion or removal of systems providing a path to enterprise reduction through integration or innovation.

The traditional static DSM models systems for analysis and integration and was first presented by Steward [8]. The matrix illustrates the relationship between components of a system, or systems-of-systems, under study by indicating where relationships exist and/or to what strength a relationship exists in a concise format (see Figure 1). DSMs allow for mathematical manipulation of the relationships and are ideal for algorithmic clustering and reduction of matrix size (number of systems).

System	1	2	3	4
1	1	1	0	1
2	1	1	0	0
3	0	1	1	0
4	1	0	0	1

System	1	2	3	4
1	1	1	0	.25
2	1	1	0	0
3	0	.75	1	0
4	.5	0	0	1

Figure 1. Simple binary relationship showing existence of a relationship (left) and strength of relationship (right)

Rows and columns can represent any number of characteristics such as system architecture, organizational/team-based DSMs or activity/process DSMS [9], and in this research, could represent shared functionality, number of information exchanges (in and out), frequency of exchange, commonality of data elements, interoperability, commonality of programming languages, cost to integrate/modify, etc.

METHODOLOGY

Due to page constraints, the C-DSM method will be discussed here briefly, but will be more fully explained in an accompanying presentation. C-DSMs use constraints and macro-level iteration to reduce the size of traditional DSMs by integrating clustered systems and making use of the diagonal to ensure relationships are not lost through the integration process. Traditional DSMs ignore the diagonal (e.g. matrix entry 1,1; 2,2; 3,3; etc.) as the relationship between a system and itself is irrelevant while the C-DSM method stores intermediate algorithmic results on the diagonal for error-checking (See Figure 2).

	SysA	SysB	SysC	SysD	SysE	SysF	SysG	SysH	SysI	SysJ
SysA	1	0.5	1	0	0.25	0.75	0	1	1	1
SysB	0.5	1	0	1	1	0.25	0.75	0	0	0.5
SysC	1	0	1	0.5	0.25	0.25	0.25	0.75	1	1
SysD	0	1	0.5	1	0	0	0	1	0.25	0.5
SysE	0.25	1	0.25	0	1	0.5	0.25	1	1	0
SysF	0.75	0.25	0.25	0	0.5	1	0.1	0.3	0.6	1
SysG	0	0.75	0.25	0	0.25	0.1	1	0	0	0
SysH	1	0	0.75	1	1	0.3	0	1	0	1
SysI	1	0	1	0.25	1	0.6	0	0	1	0.7
SysJ	1	0.5	1	0.5	0	1	0	1	0.7	1

→

	Cls1	Cls2	Cls3	Cls4	Cls5	Cls6	Cls7	Cls8
Cls1	0.5	0.5	0.6	0.5	0.4	0.5	0.5	0.8
Cls2	0.5	0.5	0.1	0.1	0.1	0.9	0.6	0.8
Cls3	0.6	0.1	0	0.5	0.3	1	1	0
Cls4	0.5	0.1	0.5	0	0.1	0.3	0.6	1
Cls5	0.4	0.1	0.3	0.1	0	0	0	0
Cls6	0.5	0.9	1	0.3	0	0	0	1
Cls7	0.5	0.6	1	0.6	0	0	0	0.7
Cls8	0.8	0.8	0	1	0	1	0.7	0

Figure 2. Clustered DSM (left) is calculated without the diagonal while the C-DSM (right) is calculated with the diagonal. This example uses simple averages in the integration calculation for clarity.

The objective of the C-DSM algorithm is to minimize the off-diagonal elements through clustering similar systems, integrating into single systems with the goal of providing a less complex DSM. The process repeats until the process owner has obtained their desired level of integration, or until the DSM has reduced to a single system. One repetition represents an arbitrary time period that is defined by the process owner and the speed of integration is controlled through user constraints and controls. Repeating the reduction algorithm and recording the clustering of systems produced a series of slowly shrinking matrices (see Figure 3) and with an accompanying integration roadmap.

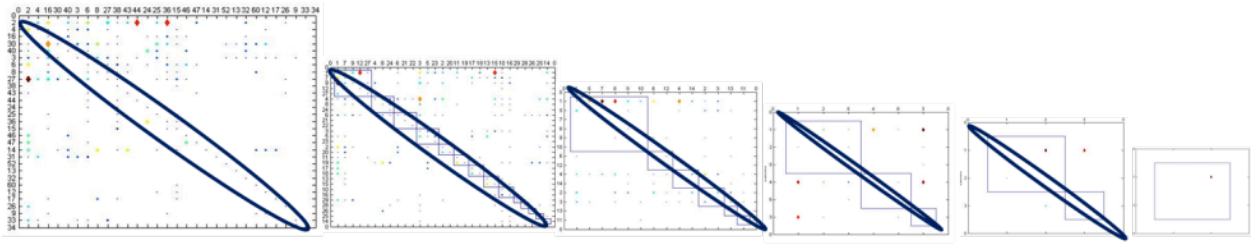

Figure 3, Graphical depiction of a collapsing DSM through iterative integration

To prepare for the C-DSM algorithm, the original DSM must be populated with values that represent the relationships between systems. Several potential factors for measuring the relationships were listed previously (such as frequency of communication), but another method that could be useful would be to have a board of experts (or even employ the Delphi method) to produce a table of attributes that can be used to calculate value measurements [10-11].

RESULTS

ERP implementation frequently employs a many-to-one or a many-to-few approach where an ERP is selected based on how similar it is to the current processes [12], and then a cutover strategy is enacted where data is moved from the old systems to the new. This process has not been successful for some large companies, but the larger the effort, the lower the chance of success. As such, the converse approach was taken in this experiment which used a slow iterative process of reducing the number of systems in a many-to-many approach (10% each iteration). In this case, 100 systems were randomly generated to represent an actual government integration project and initial cost estimates were used to generate the calculations below.

Simple cost calculations were adopted during this early method development work and provide insight on how process owners could use this method to make decisions (see Figure 4). In this particular case, it only took three iterations of clustering and integration, even at the slow constant rate of 10% per iteration, to realize cost savings as the total cost is lower after three iterations than the annual maintenance (O&M Costs) costs were before integration (at iteration zero).

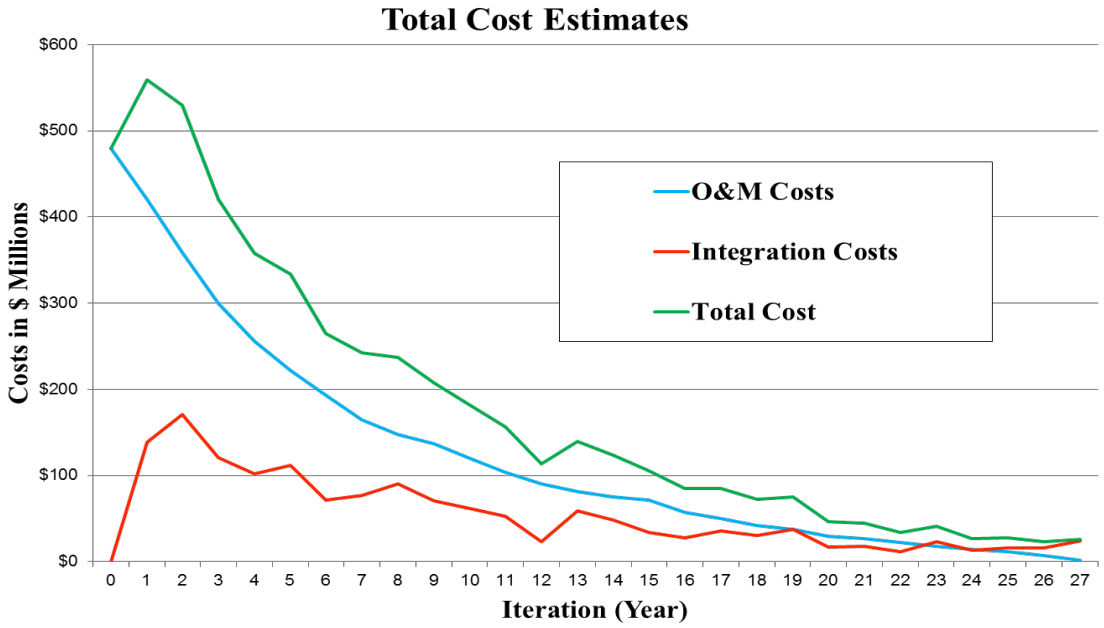

Figure 4. Initial cost estimates of solution by iteration

CONCLUSION

The C-DSM method, as with any other method, has its strengths and weaknesses. One common method for corporate sector integration is the selection of an ERP and the development of a cutover strategy. The ERP option uses a many-to-one strategy which is less onerous when the number of systems is small. In the government and military sector, the number of systems is substantial and there are frequently "untouchable" systems disallowing a complete integration to a single system. The C-DSM system allows for a user-customized integration experience that continues from many-to-fewer (possibly one) until the user decides to stop. However, when the number of systems is large, the amount of resources required to map the relationships between systems will be rather substantial. Additionally, the taxonomy used by system design planners will also drive how the C-DSM method will cluster similar systems. If there are hidden or unaccounted for relationships, then some of the suggestions made by the C-DSM algorithm will not make sense to integration planners. Fortunately, planners can choose to follow the suggestions or not and make adjustments before the next iteration of the collapsing matrix. Frequent interaction and evaluation allows for frequent course-correction, which is perhaps one of the greatest strengths of this method.

There is more work to be done to provide more robust cost estimates, but the C-DSM method presented in this paper provides a means to assist systems planners in better preparation for ERP-like integration planning by providing system designers with potential future states based on clustering parameters provided by the system designers. The C-DSM method also produces the integration plan on how to get to the future state, and provides initial cost estimates of those solutions.

REFERENCES

[1] U.S. Government Accountability Office. (2013). *Department of homeland security: Progress made and work remaining after nearly 10 years in operation.* (GAO Report No. GAO-13-370T).

[2] Reilly, S. (2012) How the air force blew $1B on a dud system. Retrieved from http://www.federaltimes.com/article/20121126/DEPARTMENTS01/311260009/How-Air-Force-blew-1-billion-dud-system

[3] Riposo, J., Weichenberg, G., Duran, C. K., & Fox, B. (2013). Improving air force enterprise resource planning-enabled business transformation. (RAND Report No. RAND_RR250).

[4] Ahmed, M. M. (2012). Critical Success Factors for ERP Implementation in SMEs. Robotics and Computer-Integrated Manufacturing.

[5] Motwani, J. S. (2005). Critical factors for successful ERP implementation: Exploratory findings from four case studies. Computers in Industry.

[6] Pineda, R.L. and Smith, E.D., Functional Analysis and Architecture, in: A.K. Kamrani and M. Azimi (Eds.), Systems Engineering Tools and Methods, 1st ed., CRC Press, Boca Raton, FL, 2011, pp. 35-79.

[7] Eppinger, S. D. and Browning, T. R., *Design Structure Matrix Methods and Applications.* Cambridge, MA: The MIT Press, 2012.

[8] Steward, D.V., "The design structure system: a method for managing the design of complex systems ," IEEE Transactions on Engineering Management, vol. 28, pp. 71-74, 1981.

[9] Browning, T. R. (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.

[10] Keeney, R., Value-Focused Thinking: A Path to Creative Decision Making. Cambridge, MA: Harvard University Press, 1992.

[11] Kirkwood, C., Strategic Decision Making: Multiobjective Decision Analysis with Spreadsheets. Belmont, CA: Wadsworth Publishing Company, 1992.

[12] E. J. Umble. Enterprise resource planning: Implementation procedures and critical success factors. *Eur. J. Oper. Res. 146(2),* pp. 241-257. 2003.