# Simplifying Solving Complex Problems

Joseph Kasser
Temasek Defence Systems Institute
National University of Singapore
Block E1, #05-05
1 Engineering Drive 2, Singapore 117576
Email tdskj@nus.edu.sg

Yang-Yang Zhao
Norwegian Institute of Systems Engineering
University College of Southeast Norway
#5369, Hasbergs vei 36
3616 Kongsberg, Norway
Email yangyang.zhao@hbv.no

*Abstract*-**Perceptions of complexity from the Holistic Thinking Perspectives (HTP) [3] indicate that there is a dichotomy on the subject of how to solve the problems associated with complex systems. While some authors opine the need for new tools and techniques to solve the problems, others show the same problems being remedied successfully. This paper examines the situation, discusses and resolves the dichotomy with reference to the Hitchins-Kasser-Massie Framework (HKMF) and the HTPs. Building on prior work the paper then maps the management of complex problems that seems to work in the real world into a notional process. The paper concludes that (1) complexity can be, and is being, managed successfully if the correct paradigms are applied and (2) the various single-pass processes for solving non-complex systems are subsets of the existing Multiple-Iteration Problem-Solving Process.**

***Keywords-complex systems, systems engineering; systems thinking; human factors; Multiple-Iteration Problem-Solving Process; Nine Systems Model.***

## I. *INTRODUCTION*

This paper documents the findings from research on the subject of how to solve the problems associated with complex systems that began in 2004. The paper, discusses and resolves the dichotomy on the subject of how to solve the problems associated with complex systems with reference to the Hitchins-Kasser-Massie Framework (HKMF) [1] and the Holistic Thinking Perspectives (HTP) [2]. Building on prior work the paper then discusses aspects of complexity and outlines a process for solving the problems associated with complex systems which appears to map into the manner that works in the real world in an alternative paradigm; the Multiple-Iteration Problem-Solving Process. Due to the limitations of space, the paper references additional examples in the literature. The paper asserts that:

1. Complexity can be managed in the appropriate paradigm.
2. The various single pass processes for solving non-complex systems are subsets of the Multiple-Iteration Problem-Solving Process for solving complex systems.

## II. *THE DICHOTOMY FOR SOLVING COMPLEX PROBLEMS*

Perceptions of complexity from the HTPs [2] indicate that there is a dichotomy on the subject of how to solve the problems associated with complex systems. On one hand there is literature on the need to develop new tools and techniques to solve the problems and on the other hand, there is literature that shows that the same problems are being remedied successfully.

Examples from the 'need to develop new tools and techniques' side of the dichotomy found in a literature review of complexity in the systems engineering domain include:

- Shinner who stated that the problems posed by complexity seem to be unmanageable [3].
- Bar-Yam [4] who:
  - Quoted the Chaos study [5] suggesting that the systemic reason for the challenged projects in the study was their inherent complexity.
  - Stated "*for all practical purposes adequate testing of complex engineered systems is impossible*" and suggested an evolutionary process for engineering large complex systems.
- The Complexity Primer for Systems Engineers [6].

Examples from the other side of the dichotomy include:

- Jenkins who defined systems engineering as, "*the science of designing complex systems in their totality to ensure that the component subsystems making up the system are designed, fitted together, checked and operated in the most efficient way*" [7].
- Systems engineering was the successful management methodology that created the Semiautomatic Ground Environment (SAGE) project, a computer and radar-based air defence systems in the United States of America in the 1950s [8]. SAGE was a massive networked system of radars, anti-aircraft guns, and computers.
- Industry being able to manage complexity without too many issues in such diverse domains as fleets of cruise ships, airlines, international air freight forwarding companies, automated rapid transit systems, banking via the Internet and Automatic Teller Machines (ATM), hospitals, oil rigs, etc.

## III. *RESOLVING THE DICHOTOMY*

The dichotomy summarized in TABLE 1 may be resolved by observing that each side is focused on one or more different non-contradictory aspects of the situation as follows:

1. One side may be talking about the need to develop new tools and techniques to solve the problems associated with producing a single correct optimal solution that satisfies the problem, while the other side consists of those who are willing to settle for an acceptable solution that satisfices the problem.

TABLE 1 SUMMARY OF REASONS FOR THE DICHOTEMY FROM VARIOUS PERSPECTIVES

| | Perspective | Need new tools and techniques | What's the problem? |
|---|---|---|---|
| **1** | **Solution paradigm** | Looks for a single correct solution | Looks for acceptable solutions |
| **2** | **HKMF Column** | A-F | G |
| **3** | **HKMF Layer** | Layer 2 moving up to Layer 3 | In Layer 3 |
| **4** | **Subjective complexity** | Very | Not at all |
| **5** | **Structure of the problem** | Confusing ill-structured with complexity | No confusion |

2. One side may be talking about developing complex systems (HKMF Columns A-F) and the other side may be taking about managing complex systems in operation (HKMF Column G).
3. One side may be positioned in the HKMF Layer 2 while the other side is positioned in the HKMF Layer 3-5. The theory of integrative levels [9] cited by Wilson [10] recognizes that system behaviour is different in the different levels of the hierarchy so that tools and techniques that work at one level may not work in others. Moreover, the Layer 2 side are used to dealing with their system in Layer 2, the metasystem in Layer 3 and the subsystem in Layer1. When they move up into Layer 3 they add Layer 4 to their area of concern but do not drop Layer 1 increasing the artificial complexity. Those in the other side of the dichotomy already in Layer 3 have dropped Layer 1 simplifying their area of concern.
4. One side may perceive the situation from a different level of subjective complexity than the other.
5. One side may be confusing ill-structured problems with complexity while the other does not.

## IV. *DEFINITIONS*

Before documenting approaches to solve the problems associated with complex systems, it is useful to have a working definition of complexity and of a complex problem.

### A. *A definition of complexity*

There is no single definition of complexity. The literature on complexity contains different definitions of the term which cloud the situation. For example:

- *"A complex system usually consists of a large number of members, elements or agents, which interact with one another and with the environment"* [11]. According to this definition the only difference between a system and a complex system is in the interpretation of the meaning of the word 'large'.
- ElMaraghy et al. wrote, *"Colwell [12] defined thirty-two complexity types in twelve different disciplines and domains such as projects, structural, technical, computational, functional, and operational complexity"* [11].
- Tomiyama et al. introduced two different types of complexity: (i) complexity by design and (ii) the intrinsic complexity of multidisciplinary, from the viewpoint of knowledge structure [13].
- Suh defined complexity as, *"the measure of uncertainty in achieving the functional requirements (FRs) of a system within their specified design range."* [14]. Suh also stated the need to abstract out things that were not pertinent to the issues at hand.

In addition, from the various definitions of a system in the literature:

- Jackson defines a system as "*A complex whole the functioning of which depends on its parts and the interactions between those parts*" [15].
- "*The classification of a system as complex or simple will depend upon the observer of the system and upon the purpose he has for constructing the system*" [16].
- "*A simple system will be perceived to consist of a small number of elements, and the interaction between these elements will be few, or at least regular. A complex system will, on the other hand, be seen as being composed of a large number of elements, and these will be highly interrelated*" [16].
- "*A complex system is an assembly of interacting members that is difficult to understand as a whole*" [17].

It appears that complexity is in the eye of the beholder [16], yet there are no specific numbers that can be used to distinguish complex systems from non-complex systems.

The attributes associated with the different definitions of objective complexity include:

1. Number of issues, functions, or variables involved in the problem
2. Degree of connectivity among those variables.
3. Type of relationships among those variables.
4. Stability of the properties of the variables over time.

### B. *A definition of a complex problem*

The scientific community cannot agree on a single definition of a complex problem [18] cited by [19]. Accordingly, let a complex problem be one of set of problems posed to remedy the causes of undesirability in a situation in which the solution to one problem affects another aspect of the undesirable situation.

## V. *THE DIFFERENT TYPES OF COMPLEXITY*

Complexity can be partitioned in various ways. Consider the benefits of partitioning complexity as either Subjective or Objective complexity.

### A. *Subjective complexity*

Sillitto distinguished between subjective and objective complexity [20] as:

- ***Subjective complexity***[1] – which means that people don't understand it and can't get their heads round it.
- ***Objective complexity*** – which means that the problem situation or the solution has an intrinsic and measurable degree of complexity.

---

[1] Which can be quantified into four levels of difficulty [21].

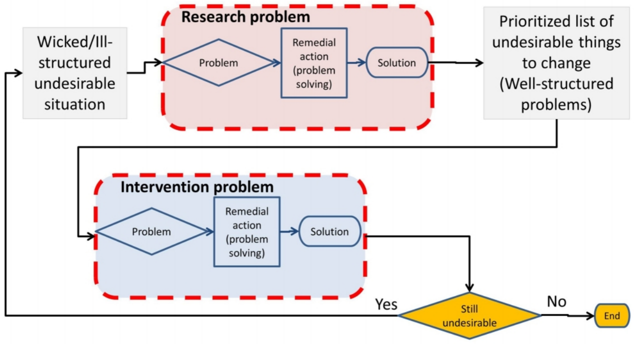

Figure 1 The Two-Part Multiple-Iteration Problem-Solving Process

There do not appear to be unique words that uniquely define the concepts of 'subjective complexity' and 'objective complexity' in the English language. Hence the literature accordingly uses the words 'complicated' and 'complex' both as synonyms to mean both subjective and objective complexity and to distinguish between subjective and objective complexity. To further muddy the situation some authors use the word complex to mean subjective complexity while others use the word complicated to mean subjective complexity and vice versa.

### B. *Objective complexity*

The various definitions of objective complexity in the literature can be aggregated into two types, as follows [22]:

1. ***Real world complexity:*** in which elements of the real world are related in some fashion, and made up of components. This complexity is not reduced by appropriate abstraction it is only hidden.
2. ***Artificial complexity:*** arising from either poor aggregation or failure to abstract out elements of the real world that, in most instances, should have been abstracted out when drawing the internal and external system boundaries, since they are not relevant to the purpose for which the system was created. For example, in today's paradigm, complex drawings are generated that contain lots of information[2] and the observer is supposed to abstract information as necessary from the drawings. The natural complexity of the area of interest is included in the drawings; hence the system is thought to be complex.

Using the analogy to complex numbers in mathematics [21], objective complexity may be considered as the real part of complexity and subjective complexity may be considered as the imaginary part since it can be reduced by education and experience.

## VI. *MANAGING COMPLEXITY VIA A PARADIGM SHIFT*

From the perspective of the problem-solving paradigm of systems engineering [23], the standard systems engineering approach can be reworded to become "*an evolutionary approach to remedying the undesirability in a situation by turning an ill-structured problem into a number of well-structured problems, remedying the well-structured problems and then integrating the partial remedies into a whole remedy*". One can then add that "*The various problem-solving processes in the literature are parts of a meta-problem-solving process that starts with ill-defined problems, converts them to well-defined problems and evolves a remedy to the set of well-defined problems recognizing that the problem may change while the remedy is being developed*" [21].

By observing the management of complex systems in industry from the HTPs it was possible to infer a meta-process for solving the problems associated with complex systems based on a modified version of the existing holistic extended problem-solving process [2]. Let this meta-process shown in Figure 1 be called the Multiple-Iteration Problem-Solving Process. The Multiple-Iteration Problem-Solving Process consists of two sequential problem-solving processes embedded in an iterative loop. The first problem-solving process converts the ill-structured problem [21] posed by the complex situation into one or more well-structured problems. Since one problem solving approach does not fit all problems [21], the second problem-solving process is tailored to remedy specific type of problems. Choice of which of the problems identified by the first problem-solving process to tackle in the second problem-solving process will depend on a number of factors including urgency, impact on undesirable situation, the need to show early results and available resources.

### A. *The first problem-solving processes*

The first problem-solving processes in the Multiple-Iteration Problem-Solving Process remedies a research problem using an adaptation of the Scientific Method [21]. This is the process that figures out the nature of the problematic situation and what needs to be done about it. The output of the process is a prioritized list of things to change (based on the causes of the undesirability and their importance). The problem solvers create the prioritized list of things to change by:

1. Gaining a thorough understanding of situation.
2. Creating the FCFDS and performing feasibility studies.
3. Identifying probable causes of undesirability.
4. Estimating the approximate contribution of each cause to the undesirable situation.
5. Developing priorities for remedying the causes.

It is often possible to achieve consensuses on what is undesirable and on the FCFDS even when consensus on the causes of the undesirability cannot be achieved.

Feasibility studies on the FCFDS are performed on the FCFDS because there is no point in creating a FCFDS if it is not feasible. Examples include:

- ***Operational feasibility:*** A single solution or combination of partial solutions is achievable.
- ***Structural (technical) feasibility:*** suitable technologies exist at the appropriate technology readiness levels.
- ***Quantitative feasibility:*** Cost, risk and uncertainty.
- ***Temporal feasibility:*** Schedule (the solution will be ready when needed).

As an example, the complex problem may be associated with undesirable traffic congestion in an urban area. The mayor, feeling under pressure to do something about the

---

[2] The DODAF Operational View (OV) diagrams can be wonderful examples of artificial complexity.

growing traffic congestion in her city, provides the trigger to initiate the problem-solving process which begins with the ill-structured problem of how to remedy the undesirable effects of traffic congestion.

An understanding of the situation might produce a number of causes. The analysis would also provide quantitative information such as an estimate of the degree of the contribution by the cause to the undesirable situation.

The need to remedy each cause would then be prioritized according to selection criteria that might include cost, schedule, political constraints, performance and robustness.

The often forgotten domain knowledge needed to gain consensus on, and prioritize the cause is 'human nature'. Each of the stakeholders needs to know 'what's in it for them' in implementing the change. So it is the task of the systems engineer to identify and communicate that information.

Tools developed for gaining an understanding of the system (situation) and the nature of its undesirability include:

- Checkland's Soft Systems Methodology (SSM) [24].
- Avison and Fitzgerald's interventionist methodology [25].
- The Nine System Model and the HTPs [26-28].

A Case Study describing the first problem-solving process is the MSOCC data switch replacement project [29]. The Case Study discusses a situation in which a SSM similar to Avison and Fitzgerald's interventionist methodology [25] coupled with an object-oriented approach for viewing requirements was used in a tailored version of the system engineering problem-solving process in a complex environment by a systems engineering team to solve the problem posed by the need to illicit, elucidate and achieve consensus on two sets of requirements[3]. Both an optimal systems architecture and optimal System Development Process (SDP) design were achieved in a relatively short period of time compared to using the standard systems engineering approach. Moreover, the customer deemed the Systems Requirements Review (SRR) and the System Requirements Document [30] complete and comprehensive.

### B. *The second problem-solving process*

The second problem-solving process in the Multiple-Iteration Problem-Solving Process is an intervention problem [21] and remedies the aspect(s) of the undesirable situation identified by the first problem-solving process by converting an undesirable situation to a situation without the undesirable aspects. Since one problem solving approach does not fit all problems [21], the second problem-solving process is tailored to remedy the specific type of problems. Once the second problem-solving process is completed the process may iterate back to the beginning for a new cycle as shown in Figure 1 because the second problem-solving process:

- May only partially remedy the original undesirable or problematic situation.
- May contain unanticipated undesirable emergent properties from the *solution system* and its interactions with its adjacent systems.
- May only partially remedy new undesirable aspects that have shown up in the situation during the time taken to develop the solution system.
- May produce new unanticipated undesired emergent properties of the solution system and its interactions with its adjacent systems which in turn produce new undesirable outcomes.

The process used by NASA in the 1960s when faced with the well-structured complex problem of landing a man on the moon and returning him safely to earth within a decade maps into the Multiple-Iteration Problem-Solving Process.

## VII. *UNDERSTANDING THE SITUATION*

The key to providing an acceptable remedy to the problem is a true understanding of the situation. The principle of hierarchies in systems [31] cited by [10] is one of the ways humanity has managed complexity for most of its recorded history. The observations of the way that industry manages complexity maps into the use of the principle of hierarchies. The process for gaining an understanding of the situation is:

1. Distinguishing between subjective and objective complexity.
2. Minimizing artificial objective complexity by abstracting out non-pertinent aspects of the situation.
3. Partitioning the situation to optimize the situation (system).
4. Understanding the relationships between the parts of the system and the emergent properties.
5. Applying the tools and techniques suited for the appropriate layer in the HKMF.
6. Using multiple partial views of the system instead of complex and complicated single views.

### A. *Distinguishing between subjective and objective complexity*

In gaining an understanding of a situation, the understanding is inferred from the observations and often stated as 'the cause is …' or 'the problem is ...'. The understanding is based on decisions and one of the decision traps, or factors that lead to bad decisions [32] is lack of domain knowledge which increases the subjective complexity. Domain knowledge is critical in all three domains: the problem, solution and implementation domains. The traditional way of acquiring short-term domain knowledge to reduce the subjective complexity is to use consultants.

### B. *Minimizing artificial objective complexity*

Artificial complexity can be minimized by optimal situation (system) partitioning. Once the situation is partitioned then each system engineer manages their area of concern which is limited to the meta-system, the SOI and its subsystems [26].

The personnel in each system in the hierarchy are operating within the 'A' paradigm of systems engineering [33] which begins in column 'A' of the HKMF. Consequently they have a vision of what their system is doing (or supposed to do in the future if it is still being developed) in the context of the entire situation. They just get on with doing it. As an example

---

[3] The MSOCC switch upgrade took place in 1989. Avison and Fitzgerald didn't publish their methodology until 2003.

consider an allied naval convoy crossing the North Atlantic Ocean in 1942. The convoy is a system[4]. Each ship in the convoy can be considered as both a subsystem of the convoy, or as a system[5]. There was a CONOPS for the convoy. There were separate CONOPS for the naval escort ships and the merchant vessels describing the actions and interactions of these subsystems of the convoy in various scenarios.

. Consider the example of the complex problem of docking two spacecraft [22]. Once the spacecraft are close, the problem is simplified by minimizing artificial complexity and creating a closed system view to only consider the:

- Relative positions of the spacecraft.
- Relative velocity of the spacecraft.
- Relative alignment in X, Y and Z orientation of the spacecraft.

The problem is then set up in the context of a closed system to produce a relative docking velocity close to zero with the docking collars on both spacecraft properly aligned.

### C. *Partitioning the situation to optimize the system*

Partitioning the situation to optimize the system represents a challenge for reasons that include:

- The systems optimization paradox which was stated by Machol and Miles who wrote, "*the principle of suboptimization states that optimization of each subsystem independently will not lead in general to a system optimum, and that improvement of a particular subsystem actually may worsen the overall system. Since every system is merely a subsystem of some larger system, this principle presents a difficult if not insoluble problem, - one that is always present in any major systems design*" [34].

- There will usually be different viewpoints on what should be optimized.

Wymore stated, "*Conventional systems engineering wisdom has it that if subsystems are optimized, then the system cannot be optimum*" [35] and then used a mathematical approach to show that conventional wisdom was mistaken and how it was possible for systems engineering to ensure that optimum design of the subsystems can result in optimum design of the system. System optimization at one level is always a subsystem optimization of the metasystem. If any system is a subsystem of the containing or metasystem, then where does the optimization take place? The answer is that system optimization at any level optimizes the interactions between the subsystems of that system level within the constraints imposed by the systems engineer of the metasystem.

A useful template for partitioning the situation is the Nine Systems Model [27, 36] which comprises nine situations, processes and socio-technical systems in a clearly defined interdependent manner across multi-system-levels. The *Functional* perspective of the Nine System Model shown in Figure 2 shows the relationships between the situations, systems and processes. For an example of applying the Nine System Model to manage complexity in the MSOCC data switch replacement project, see Kasser, Zhao and Mirchandani [26].

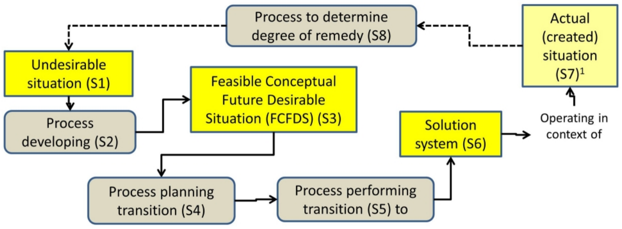

Figure 2 The Nine-System Model (*Functional* perspective)

1. One set of guidelines for partitioning the situation to optimize the system with reference to the Nine System Model is:
2. Examine the undesirable situation (S1) from several different perspectives.
3. Develop an understanding of the situation (S1).
4. Create the FCFDS containing the SOI (S3).
5. Use the principle of hierarchies to abstract out the artificial complexity.
6. Abstract out the parts of the situations (S1 and S3) that are not pertinent to the problem.
7. Partition the FCFDS (S3) into the SOI (S6) and adjacent systems.
8. Optimize the interfaces.
9. Partition the SOI into subsystems.

### D. *Understanding the relationships between the parts of the system and the emergent properties*

The understanding is achieved by:

1. Creating models of the system behaviour generally from the *Operational* and *Functional* HTPs.
2. Using mathematical tools such as Queuing Theory commonly used in Operations Research in HKMF Area 3G.

### E. *Applying the tools and techniques suited for the appropriate layer in the HKMF*

If the problem is in HKMF Layer 3-5 use tools and techniques for those layers rather than attempting to adapt tools and techniques from HKMF Layer 2.

### F. *Using multiple partial views of the system instead of complex and complicated single views*

Multiple partial views minimize artificial complexity and allow non-pertinent attributes to be hidden facilitating understanding particular aspects of the complex situation. Examples are:

1. The various views of a building; e.g. separate structural drawings, plumbing views for the water supply and sewage pipes, and electrical drawings.
2. Separating open system and closed system views such as in the spacecraft docking problem situation discussed above.

---

[4] Some people might call it a System of Systems.

[5] Alternatively, the naval ships could be one subsystem and the merchant marine ships a second subsystem of the convoy. Each ship is then a subsystem within the naval or civilian subsystem of the convoy. If there are ships from the navies of more than one allied country in the convoy, then the ships of each country could constitute a subsystem within the naval subsystem. The choice of subsystem partitioning depends on the issues being considered.

## VIII. *SUMMARY*

This paper discussed the subject of how to solve the problems associated with complex systems, and resolved the dichotomy with reference to the HKMF [1] and the HTPs [2]. Building on prior work the paper then discussed aspects of complexity and outlined aspects of a process for solving the problems associated with complex systems which maps into the manner that appears to work in the real world; the Multiple-Iteration Problem-Solving Process. Due to the limitations of space, the paper referenced additional examples in the literature.

## IX. *CONCLUSIONS*

The conclusions from the research are:

1. Complexity can be, and is being, managed successfully if the correct paradigms are applied, namely:

   Apply the tools and techniques suited for the appropriate layer in the HKMF.

   Develop acceptable solutions that satisfice the problem rather than single correct solutions that satisfy the problem.

2. The various single pass processes for solving non-complex problems are subsets of the Multiple-Iteration Problem-Solving Process for solving complex systems.

## X. *REFERENCES*

[1] J. E. Kasser, “The Hitchins-Kasser-Massie (HKM) Framework for Systems Engineering,” proceedings of the the 17th International Symposium of the INCOSE, San Diego, CA., 2007.

[2] J. E. Kasser, Holistic Thinking: creating innovative solutions to complex problems, 2 ed.: Createspace Ltd., 2015.

[3] S. M. Shinner, A Guide to Systems Engineering Management, Lexington Massachusetts: Lexington Books, 1976.

[4] Y. Bar-Yam, “When Systems Engineering Fails --- Toward Complex Systems Engineering,” proceedings of the Systems, Man and Cybernetics, 2003. IEEE International Conference on, 2003.

[5] CHAOS. "The Chaos study," March 19, 1998; http://www.standishgroup.com/chaos.html.

[6] S. Sheard et al., "A Complexity Primer for Systems Engineers," INCOSE Complex Systems Working Group, 2013, p. 10.

[7] G. M. Jenkins, "The Systems Approach," Systems Behaviour, J. Beishon and G. Peters, eds., p. 82, London: Harper and Row, 1969.

[8] T. P. Hughes, Rescuing Prometheus, New York: Random House Inc., 1998.

[9] J. Needham, Integrative levels: a revaluation of the idea of progress, Oxford: Clarendon Press, 1937.

[10] T. D. Wilson, “Philosophical foundations and research relevance: issues for information research (Keynote address),” proceedings of the Fourth International Conference on Conceptions of Library and Information Science: Emerging Frameworks and Method, University of Washington, Seattle, USA, 2002.

[11] W. ElMaraghy et al., “Complexity in engineering design and manufacturing,” CIRP Annals - Manufacturing Technology, vol. 61, no. 2, pp. 793-814, 2012.

[12] B. Colwell, “ Complexity in Design,” IEEE Computer, vol. 38, no. 10, pp. 10-12, 2005.

[13] T. Tomiyama et al., “Complexity of Multi-Disciplinary Design,” Annals of the CIRP, vol. 56, no. 1, pp. 185-188, 2007.

[14] N. P. Suh, “Complexity in Engineering.,” Annals of CIRP, vol. 54, no. 2, pp. 46-63, 2005.

[15] M. C. Jackson, Systems thinking: Creative Holism for Managers, Chicheste, UK: John Wiley & Sons Inc., 2003.

[16] M. C. Jackson, and P. Keys, “Towards a System of Systems Methodologies,” Journal of the Operations Research Society, vol. 35, no. 6, pp. 473-486, 1984.

[17] J. T. Allison, “Complex System Optimization: A Review of Analytical Target Cascading, Collaborative Optimization, and Other Formulations,” The University of Michigan, 2004.

[18] J. Quesada, W. Kintsch, and E. Gomez, “Complex Problem Solving: A field in search of a definition? ,” Theoretical Issues in Ergonomic Science, vol. 6, no. 1, pp. 5-33, 2005.

[19] A. Fischer, S. Greiff, and J. Funke, “The Process of Solving Complex Problems,” The Journal of Problem Solving, vol. 4, no. 1, pp. 19-42, 2012.

[20] H. Sillitto, “On Systems Architects and Systems Architecting: some thoughts on explaining and improving the art and science of systems architecting,” proceedings of the the 19th International Symposium of the International Council on Systems Engineering, Singapore, 2009.

[21] J. E. Kasser, and Y.-Y. Zhao, "The myths and the reality of problem-solving," Systemic and systematic Integrated Logistics Support: Course notes, J. E. Kasser, ed., Singapore: TDSI, 2016.

[22] J. E. Kasser, and K. Palmer, “Reducing and Managing Complexity by Changing the Boundaries of the System,” proceedings of the the Conference on Systems Engineering Research, Hoboken NJ, 2005.

[23] J. E. Kasser, and D. K. Hitchins, “Yes systems engineering, you are a discipline,” proceedings of the the 22nd Annual International Symposium of the International Council on Systems Engineering, Rome, Italy, 2012.

[24] P. Checkland, Systems Thinking, Systems Practice: John Wiley & Sons, 1993.

[25] D. Avison, and G. Fitzgerald, Information Systems Development: Methodologies, Techniques and Tools, Maidenhead: McGraw-Hill Education (UK), 2003.

[26] J. E. Kasser, Y.-Y. Zhao, and C. J. Mirchandani, “Simplifying Managing Stakeholder Expectations using the Nine-System Model and the Holistic Thinking Perspectives,” proceedings of the the 24th International Symposium of the International Council on Systems Engineering (INCOSE), Las Vegas, NV, 2014.

[27] J. E. Kasser, and Y.-Y. Zhao, “Managing Complexity via the Nine Systems in Systems Engineering,” proceedings of the the 24th International Symposium of the International Council on Systems Engineering (INCOSE), Las Vegas, NV, 2014.

[28] J. E. Kasser, and Y.-Y. Zhao, “Simplifying Solving Complex Problems,” proceedings of the the 11th International Conference on System of Systems Engineering, 2016.

[29] J. E. Kasser, and C. J. Mirchandani, “The MSOCC Data Switch Replacement: A Case Study in Elicitating and Elucidating Requirements,” proceedings of the the 15th International Symposium of the International Council on Systems Engineering (INCOSE), Rochester, NY, 2005.

[30] NASA/GSFC, Multi-Satellite Operations Control Center (MSOCC) Communications Switching System (MCSS) System Requirements Review, December 1989, 1989.

[31] H. Spencer, "First principles," A System of Synthetic Philosophy, Williams and Norgate, ed., London, 1862.

[32] J. E. Russo, and P. H. Schoemaker, Decision Traps, New York: Simon and Schuster, 1989.

[33] J. E. Kasser, “Getting the Right Requirements Right,” proceedings of the the 22nd Annual International Symposium of the International Council on Systems Engineering, Rome, Italy, 2012.

[34] R. E. Machol, and R. F. Miles Jnr, "The Engineering of Large Scale Systems," Systems Concepts, Wiley Series on Systems Engineering and Analysis R. F. Miles Jnr, ed., pp. 33-50: John Wiley & Son, Inc., 1973.

[35] A. W. Wymore. "Subsystem Optimization Implies System Suboptimization: Not!," 30 September 2013, 2013; http://www.sie.arizona.edu/sysengr/wymore/optimal.html.

[36] J. E. Kasser, Perceptions of Systems Engineering: Createspace, 2015.