25th Annual INCOSE International Symposium (IS2015)
Seattle, July 13-16, 2015

Exploring the Role of Design in Systems Engineering

Ali Shafaat
Lyles School of Civil Engineering
Purdue University
ashafaat@purdue.edu

C. Robert Kenley
School of Industrial Engineering
Purdue University
kenley@purdue.edu

Abstract. The paper explores the role of design in systems engineering. It reviews the treatment of design in systems engineering in general and in the Systems Engineering Handbook in particular, and concludes that design is the missing dimension in systems engineering. It provides a review of salient viewpoints from research in design that can enhance the understanding of the design dimension in system engineering.

1. Introduction

System design is one of the key activities in system engineering. For example, Ryschkewitsch et al. (2009) state “Systems engineering is first and foremost about getting the right design.” The main goal of system design is satisfying operational requirements by creating a technical solution; however, translating all of the information to accurate technical specifications can be challenging. This translation is essential when constructing well-structured problems from the large-scale ill-structured problems. During the second half of the twentieth century, numerous attempts were made to develop best practices for tackling complex problems systematically and holistically. System Engineering tried to embrace the holistic concept of "systems" and interdisciplinary engineering processes through the life cycle of the product.

Just as the definition of systems engineering has been elusive, design has also been a difficult activity to define, but in general, it is about specifying an artefact that satisfies the objectives and associated constraints (Visser, 2004). Typical systems engineering problem definitions have many degrees of freedom in the initial state. Typically, the requirements given at the start are never complete or clear nor are they sufficient to define achievable and measurable goals, so a progressive definition of new requirements is necessary (Detienne , 2006). Designers develop heuristics to tackle complex problems systematically and holistically over time as they learn more about the problem, and this often requires collaboration among participants with multiple competencies. Bucciarelli (1988) and Schön (1988) developed social aspect of design process, which includes interactions not just between designers but also between designers and users. As Detienne (2006) pointed out, the role of users may be changing from informative and evaluative (verification) to more generative (solution elaboration) and sometimes decisional, which approaches a situation in which users are co-designing.

This paper generally reviews design-related activities in systems engineering and specifically evaluates the treatment of design in the INCOSE System Engineering Handbook (Haskins 2011). It presents findings from research on design theory for complex problems and discusses how these theories might be incorporated into the handbook and other systems engineering guidance.

2. Design-Related Concepts in Systems Engineering

Systems engineering practitioners and scholars chose different terms that cover overlapping concepts based on their background and perception. This is especially true for design-related vocabulary. This section discusses the more common terms related to design: systems engineering, systems analysis, systems architecting, and systems design.

Systems Engineering (SE): Brill (1994) described the long debate on the definition of systems engineering as a "semantics jungle." One problem is that systems engineering has both a 'systematic' and 'systemic' nature (Chestnut 1967). The systematic nature of systems engineering focuses on management processes, and the systemic nature focuses on design methods.

Stupples (2006) believes that the systematic nature of systems engineering is the current face of the discipline. He states, "SE has sadly lost its science foundation and is being practiced widely as the application of management processes to systems design and hence information required for decision making under uncertainty is not being generated." Comparing earlier definitions of systems engineering with recent ones supports Stupples' assertion. One of the earlier definitions (Chase 1974) states that SE is "the process of selecting and synthesizing the application of the appropriate scientific and technical knowledge to translate system requirements into system design and subsequently to produce the composite of equipment, skills, and techniques that can be effectively employed as a coherent whole to achieve some stated goal or purpose." Here, as Rhodes and Hastings (2004) pointed out, the emphasis was on the translation of requirements for the design process.

The coordinative and managerial view to SE has emerged in the past two decades as can be seen in two definitions of SE. First, INCOSE (Haskins 2011) defined SE as "an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs." The intent of INCOSE definition may have been to capture the full context within which a system is designed; however, in doing so, it increased the emphasis management processes. Second, NASA (2007) defined SE as "the art and science of developing an operable system that can meet requirements within imposed constraints. It is holistic and integrative, and incorporates and balances the contributions of structural, electrical mechanism-design, and power engineers, plus many other disciplines, including systems safety, to produce a coherent whole that no single discipline dominates. Systems engineering is about trade-offs and compromises, about generalists rather than specialists." Again, the stress is on the process that coordinates the contributions of multiple disciplines of rather than on systemic design methods.

System Analysis (SA): "Analysis" decomposes a problem into components, while "design" generally means "synthesis" or combining elements into a functional new whole (Gibson et al. 2007). System analysis looks at cost, schedule, risk, and other performance measures as the critical aspects of the deliverables for each designed alternative. In large problems (with mostly aleatoric uncertainty) system analysis can use mathematical modelling (similar to operations research) to find the optimum solution. Analysis can be seen as an evaluation process and helpful for defining the ill-structured problems. System analysts also confirm that the designed system will meet requirements (Sheard 1996), and the systems analysis process in ANSI/EIA 632 follows this approach.

Systems Architecting: Systems architecting is concerned with creating and building systems. IEEE 610.12:1990 states, "Architecture is the organization of the system components, their relations to each other, and to the environment, and the principles guiding its design and evolution." Maier and Rechtin (2000) indicate that architecting strives for fit, balance, and compromise among the tensions of client needs and interest (functions) and resources, and technology (physical) Systems architecting is built by four methodologies (Maier and Rechtin 2000) and they all include science and art:

1. Normative (solution-based), such as building codes and communication standards.
2. Rational (method-based), such as systems analysis and engineering.
3. Participative (stakeholder-based), such as concurrent engineering and brainstorming.
4. Heuristic (lessons-learned), such as "Simplify. Simplify. Simplify."

System architectures are developed at multiple levels; the higher-level architecture of a system is created and then decomposed to define lower levels. At all levels, developing architectures starts with definition of a problem and conception of solution (Cole 2008). Levis (1993) as part of decomposition process defined three architectures called, functional, physical, and allocated as shown in Figure 2. Buede (2009) indicates in his textbook that these architectures are developed in parallel with close interaction.

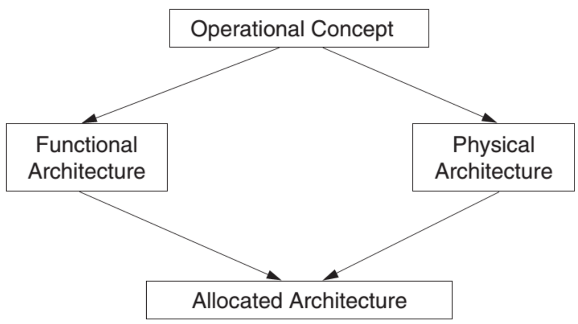


Figure 1. Architecture development (Levis 1993)

The physical architecture includes interfaces, activity allocations, data and information flow, communications links, geographical distribution, and other attributes (Cole 2008).

Systems Design (SD): There are two main perspectives on systems design. The first group looks at SD as “system design architecture” which is very close to high-level system architecture. The culmination of systems design from this perspective occurs when the components, modules, interfaces, and data for the system requirements satisfaction are defined. This viewpoint can be seen in DOD-STD-2167A. The second approach is more general and looks at design as the process of finding solutions for the user requirements. ANSI/EIA-632 (2003) defined some system design processes that “are used to convert agreed-upon requirements of the acquirer into a set of realizable products that satisfy acquirer and other stakeholder requirements.”

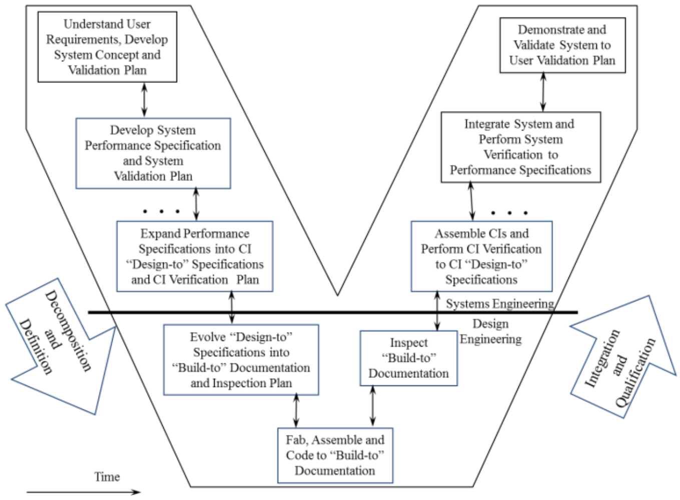
3. Treatment of Design in the INCOSE System Engineering Handbook and the Lost Dimension

The INCOSE System Engineering Handbook (Haskins 20101) is organized around processes inherited from the ISO 15288 standard for systems and software engineering. We reviewed the handbook for its statements on design and complied the strengths and weaknesses of its treatment of the topic of design in Table 1.

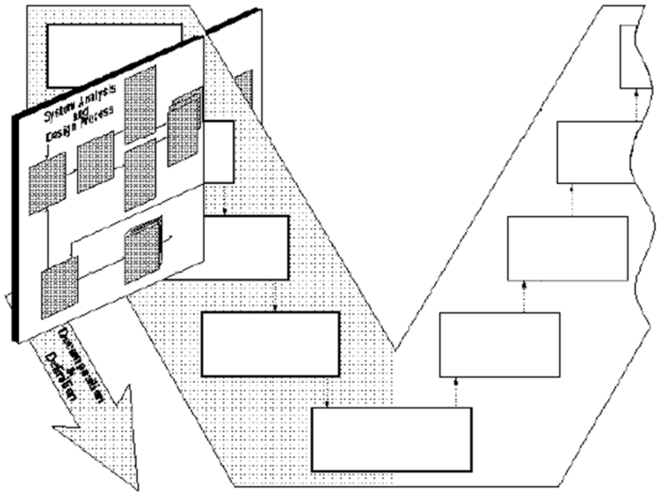
Table1: Strengths and Weaknesses of INCOSE SE Handbook

Strengths	Weaknesses
1. Presents design in multiple phases 2. “Design” appears more than 500 times 3. Differentiates between design engineering at lower levels and architectural design at the system level	1. Inconsistency in use of terminology: design engineering, design space, design drivers, architecture, and so on. 2. Sometimes problem definition is part of the design, and sometimes it is not 3. Lack of human, cognitive perspective 4. No indication of the means to empower designers to thrive within the management processes

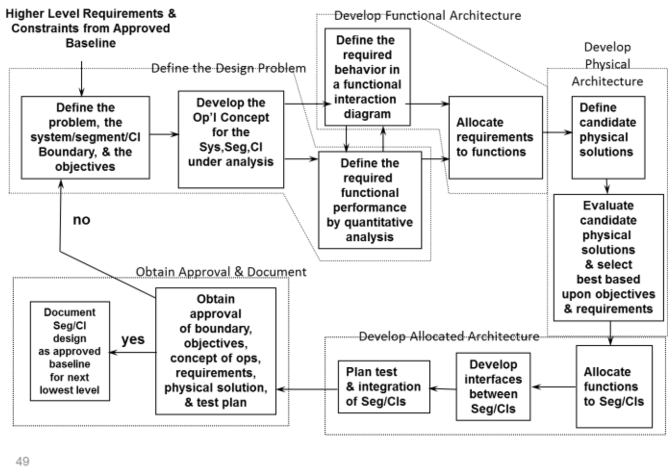
Forsberg and Mooz (1991) introduced the Vee Model (Figure 2(a)) to address the technical aspects of a project’s life cycle and to represent the time-ordered sequence of project events. In the same paper, they described the relationship between the Vee Model and an orthogonal iterative systems analysis and design process as shown in Figure 2(b). Buede (2009) developed his own version of the orthogonal iterative process (Figure 2(c)), which he refers to as systems engineering design. Unfortunately, very few have adopted this two-part orthogonal model and have focused instead on the sequential Vee Model as their sole project management and systems engineering model and have lost the other dimension of the iterative systems engineering design process that is executed at each step of decomposition and definition coming down the left-hand side of the Vee. Specifically, the organization of the INCOSE System Engineering Handbook (Haskins 2011) around management processes is in line with adopting a Vee-Model-only way of thinking about systems engineering.



(a) Systems Engineering Vee (after Forsberg and Mooz) (Buede 2009)



(b) Application of System Analysis and Design Process to the Technical Aspect of the Project Cycle (Forsberg and Mooz 1991)



(c) Detailed Functions of Systems Engineering Design (Buede 2009)

Figure 2. The Vee model and the orthogonal dimension of design

One consequence of losing the design dimension is a reduction in ability to deal with complexity. Buede articulates the need to embed the iterative systems engineering design process within the top-down Vee management process to deal with complexity in evaluating trade offs and making design decisions: “These processes are iterative. Design starts as a top-down process and is analogous to peeling an onion to uncover the specifications associated with increasingly detailed components of the system. However, the trade offs and decisions associated with the design process are so complex and intertwined that there is significant movement between low-level and high-level design issues.” (Buede 2009, 45)

In addition to complexity in decision-making, systems engineers must cope with exogenous and endogenous complexity that relates to the system’s architecture (Frenken 2006). Exogenous complexity is characterized by a topology of interactions between system elements, which remains unchanged during modelling so the modeler can establish the complexity and use this information in ex-ante analysis. On the other hand, in endogenous complexity, the topology of interactions changes over time, which means that the degree of complexity is no longer constant and the modeler cannot control it. However, during system design, including architecture and component design, interactions are dynamic in nature so generally they change as design evolves. New connections need to be added, and some eliminated, and remaining interactions may have different characteristics at different phases of design. Therefore, even though the system itself is exogenously complex, the design is endogenously complex.

The complex nature of system engineering problems makes it difficult for the end user (acquirer) to define the problem precisely in the initial stages of design due to the “wicked” (Rittel and Webber 1973) nature of the problem or users’ familiarity with the system. The systems engineering process is initiated by studying the user’s need and is followed by user requirement definition (Haskins 2011). At this stage the systems engineer works to understand the end user’s needs, while the end user’s definition of the problem evolves because of the dialogue between system engineer and end user. The system engineers’ feedback even in that early phase helps the end users with their problem definition and possible solution space. Therefore, an effective interaction between the system engineering team and end users (acquirer) supports a transformation from an ill-defined, unstructured problem to an ill-structured problem. This transformation becomes possible only when the parties start learning more about the problem, environment, and the context.

Furthermore, in most systems engineering approaches, user requirements have to be documented before the conclusion of the exploratory research stage; however, for complex systems the lack of clarity, uncertainty, and dynamic nature may make it difficult to achieve the quality and stability of requirements that one might deem acceptable. Thus, it may be unclear when to stop the requirements engineering and transition to designing a solution. More importantly, the design process often gives new insight into the problem, so the problem definition and its solution space are changing evolving together (Rittel and Webber 1973). Such wicked problems are essentially unique, and can be explained in numerous ways (Dickerson and Mavris 2010), and as such, this type of problem and the proposed solution are highly dependent on both context and the situation.

4. Three Views of the Systems Engineering Design Process

The following sections will discuss the systems engineering design process from three perspectives:

(1) the role of learning in systems engineering design,
(2) systems engineering design as a situative process, and
(3) systems engineering design as a social process.

These three lenses of system engineering design provide insights into how to best approach a system engineering design problems.

Systems Engineering Design as a Reflective Process

Design as a learning process has received significant attention in both the educational and organizational domains. Students and employees learn about the context, tools, and techniques by experiencing them during design. Design-based learning or learning by design (Kolodner et al. 1998) and similar frameworks have been developed based on this justification.

While designers are acquiring design skills and context knowledge, they also are gradually gathering knowledge about the problem they are working on. Designers evaluate routes for reaching better solutions by experimentation. An experiment can be as simple as sketching and evaluating the sketches with a critical eye. Design can be seen as an iterative process of these learning cycles (propose, experiment, and reflect) until a satisfactory solution will be reached (Lawson and Dorst 2009). The cycle is based on the framework of Argyris and Schön (1978) that treats learning as error detection and correction.

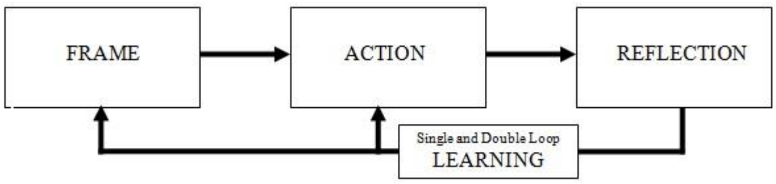


Figure 3: Schön reflection model (adapted from Schön (1983))

Designers look at a problem from different angles and propose ideas, experiment with the ideas, and learn about the problem from the results. The results come from the self-reflection or feedback from other team members, other disciplines, or stakeholders (Valkenburg and Dorst 1998). The reflections ask for a small change or even a change to the governing variables. Schön first described this model in his book (1983), as shown in Figure 3.

These processes are not only experienced by the design team during system engineering, but the acquirer and other stakeholders will go through the similar iterations depending on their degree of contribution in design process. However, since the acquirer and other stakeholders are mainly dealing with system level problems, the number of iterations, the nature of action, and the type of reflection can be different from the way design teams approach sub-system and component level problems. Despite the differences, both designers and stakeholders are learning and this learning helps them in reframing the problem and solution.

As mentioned before, the problem's initial state has many degrees of freedom. On the other hand, the design requirements, in terms of sub-goals and constraints, are changing because of the learning process. Therefore, it is necessary for system engineering standards to accommodate learning explicitly as part of almost every aspect of the system engineering process. This should consider that the design process helps expedite the designer's learning and better learning enhances the design quality.

Therefore, problem and solution co-evolution should have enough time and reflection has to be facilitated purposefully.

Systems Engineering Design as Situative Cognition

This evolution of the problem definition and the solution space is based on the information that is received and being processed during the learning process. This is a cognitive process, and cognitive processes are different for each social entity. If two teams try to design a solution for the same problem, the way they design will most likely be different. One of the main reasons is the different interactions (human and non-human) are emerging and being experienced during the design. Social entities have different pattern of interactions, internally and externally, and these patterns are changing over time shaping different overall perspectives for the designers. Therefore, their learning process and their developed schema depend on the situation they are in at the time. This is known as the "situative" perspective on human knowledge, which has been applied to design processes. Clancy (2009) stated that situative perspective looks at "human knowledge as arising conceptually through dynamic construction and/or reinterpretation within a specific social context." Figure 4 schematically depicted this notion.

Design processes also have similar characteristics; a design process starts with a plan in mind that re-forms as it is being worked on in (Ullman et al 1988). This is especially true for ill-structured problems (Zimring and Craig 2001). Greeno et al., (1998) assert that situative cognition focuses more on analyzing larger system behavior than on individuals. In this larger system, knowledge is created as a by-product of unique relationship between individuals and the environment (including both human and non-human objects) within which knowledge is embedded in (Brown et al. 1989).

To shape the large interactive social system, Bucciarelli (1994) advocated direct face-to-face exchange to understand differences and to construct meaning. He sees "no overriding perspective, method, science, or technique that can control or manage the design process in object-world terms. The process is necessarily social and requires participants to negotiate the differences and construct meaning." From situated design perspective, knowing is distributed in the system so active participation of the designer is the focal point of the design process rather than the designer's technical skill set. However, beyond design itself, sometimes the solution is situated as well. This is to say that if the solution was detached from its context and applied in a new context, the solution would no longer work unless one could find enough analogous cues in the new situation (Greeno, Collins, and Resnick, 1996). Creativity also is seen as situated endeavor as it occurs in the relationship between an individual and a society, and between an individual and the technical environment (Fischer 1999). So textbooks should emphasize on the facts that design is a situated process, involves situated knowing, and consequently the solution is situated. Handbooks should describe systematic ways to build a system of interactions for designers and design teams.

Systems Engineering Design as Social Process

In addition to a situated perspective on distributed knowing and the need for active participation of the design team in the evolution of the problem definition, design has another social aspect. Coupling of work and organization looks at the collaboration between co-designers from the task interdependency lens (Detienne 2006). The group participation and communication portion of design process is present in many design descriptions (i.e. Gennari and Reddy 2000). One of the reasons is the exogenous complexity of the design process as mentioned before. This complexity produces dynamic interdependencies (Detienne, 2006), and as a result, design can be characterized as a tightly coupled workflow (Olson and Olson 2000).

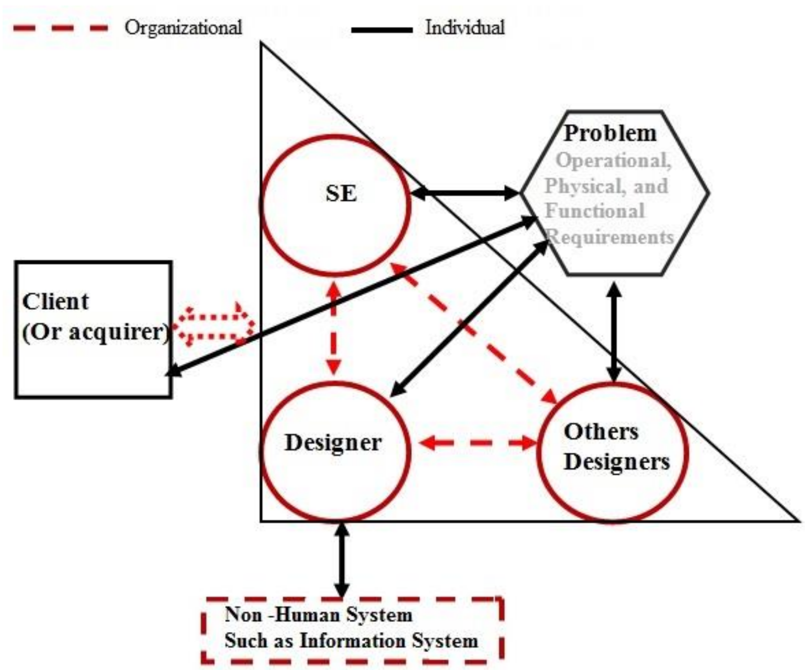


Figure 4: Designers' interactions

Researchers have observed and measured many social processes such as conflict, cooperation, competition, adaptation, and assimilation during the design process. Bucciarelli (1994) looked at the social processes during the design: "…the process of designing is a process of achieving consensus among participants with different 'interests' in the design, and that those different interests are not reconcilable in object-world terms." The misalignments among perspectives and impact on process/problem/solutions can be experienced in all design problems with more than one stakeholder, and this brings value to the design process and consequently the agreed solution. This supports the assertion in the INCOSE System Engineering Handbook (Haskins 2011) that "the best architectures, requirements, and designs emerge from self-organizing teams." It is not results from different levels of knowledge and expertise among team members, but also from their attitudes, motivation, and organizational position.

5. Discussion

Design has a significant role in system engineering. Although the System Engineering Handbook (Haskins 2011) mentions the word "design" more than 500 times in, we believe that the lost dimension of the iterative systems engineering design needs to covered more explicitly in the handbook and other systems engineering standards. Clearly, the quality of the design influences the quality of the product significantly. Specifically, we have asserted the importance of the learning, social, and cognition processes in creation of shared understanding of problem between actors from different disciplines and organizations during the design process. We believe the handbook can make descriptions of the design dimension more clear and more broadly applicable. We highlight three perspectives that can be included in the handbook:

Learning as a variable: As mentioned in the previous section, designers are learning about the problem as they are designing. The question is how does it happen? Moreover, how it can be enriched? Part of the learning is based on the information gathering. However, information cannot be gathered meaningfully before problem understanding, on the other hand problem cannot be understood without information Designers fill the gap by iteration which is described by Adams et al. (2003) as reflection. According to Schön, reflection is critical element of design. A major portion of reflection a designer receives is self-reflection. During reflection, a designer (or design team) uses lenses to assign meaning(s) to experiment that can help future actions (Turns et al. 2014). The quality of reflection and consequent learning can be enhanced by considering iteration, explanation, and reasoning as part of the designers' responsibilities. The more they learn about the problem, more suitable alternatives will be generated, progress will be faster, and results will be more legitimate. System engineering handbooks should see the learning processes, such as ideating, experimenting, and reflecting, for all stakeholders and help the organizations to empower them.

Moving from processes to individuals and their roles: Human are the real drivers of the processes so their empowerment is crucial for system design success. The current INCOSE System Engineering Handbook's (Haskins 2011) focus more on the phases and the processes. It is helpful for the managers and outside observers to understand the general procedure. However, for the individuals who are performing systems engineering, it is more important to be aware of their role and as well as other co-designers whom they can have helpful interactions. Therefore, a shift from focusing on the process to responsibility can help the human being who is performing the role to understand his position and suitable interactions better. Then rather than defining a process using a phrase such as a "component design is," it is suggested to have phrases such as "component designer is doing." This approach is supported by theories from the design, psychology, and human factors literature.

Moving from individual to social entity: Design does not happen without a lot of persuasion or even argument among individuals who have different interests and perspectives. The process is sometimes incoherent and messy and there are conflicts because it is getting further from the co-designers' idealized picture. However, after the storming stage co-designers from different disciplines can build a shared understanding about the design content. This is in accordance with the "situative" perspective as from the "situative" perspective, cognition and learning, including conceptual understanding and conceptual growth, are achievements of interaction (Greeno and Van De Sande 2007). This understanding is valuable because it increases the quality of the final solution (Valkenburg, 2000; Dong, 2005). The interactions among co-designers are not limited to one specific level such as component level. The interactions can cross the hierarchical levels, at different time as shown in Figure 5. The designers at lower levels have influence on the problem understanding of the designers and decision makers at higher levels. This interaction is not in one direction from the bottom up; system architects and designers at the higher level are changing the environment, constraints, and specifications for the lower level designers from the top down as well. These interactions, as well as interaction across the same level with co-designers, should continue during the design process. Techniques like big room meetings, creating communities of practice, building knowledge management, and information systems are some, out of many, of the ways to increase interactions among co-designers.

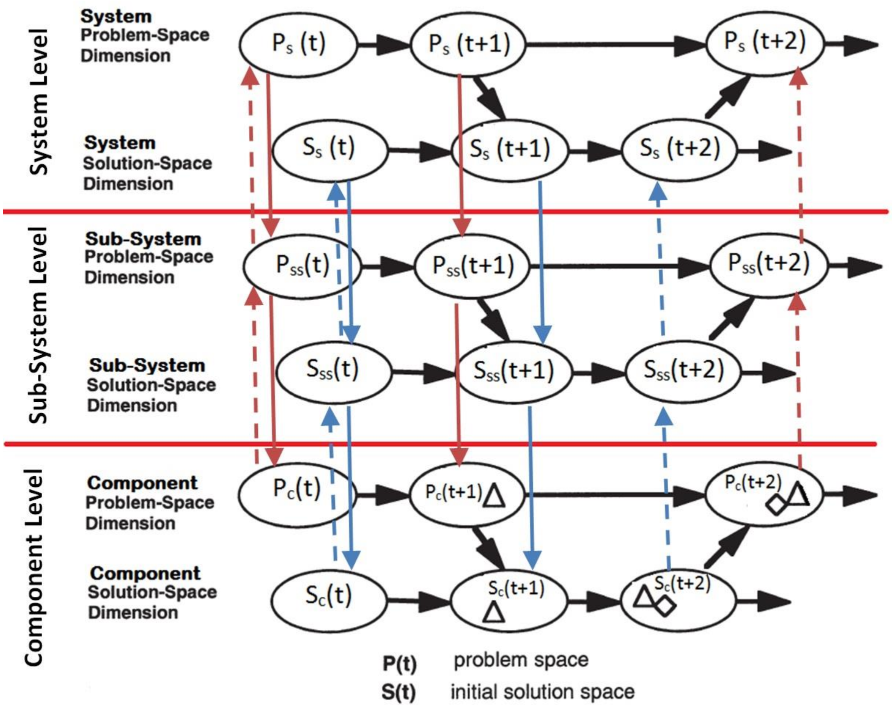


Figure 5: Problem-solution coevolution and interaction among designers across hierarchy (adapted from (Gero and Maher 1997))

When design is seen as a social process (full of ambiguity and uncertainty), it is far from the ideal image of an instrumental process depicted by handbooks. This viewpoint helps the system engineer and system designer to value interactions, reflections, and communication more and to make sure they are working well.

Conclusion

There is a need for more explicit and focused coverage of design in the INCOSE Systems Engineering Handbook (Haskin 2011). Recognizing the lost dimension of design and articulating the realities of the design process in all its messiness from the learning, situative, and social points of view can provide those responsible for managing systems engineering processes a better understanding of design. This allows managers to empower designers by creating suitable social settings for executing good design practices. There also is significant learning that could result from permitting researchers to study the design processes employed in large systems engineering problems to discover the social settings that enable success in the engineering of systems.

References

Adams, R. S., Turns, J., and Atman C. J. 2003. "Educating effective engineering designers: The role of reflective practice." *Design studies* 24(3): 275-294.

ANSI/EIA. 2003. *Processes for Engineering a System.* Philadelphia, PA, USA: American National Standards Institute (ANSI)/Electronic Industries Association (EIA)

Arnold, S. 2002. "ISO 15288 Systems engineering—System life cycle processes." *International Standards Organisation.*

Argyris, C., & Schon, D. 1978. "Organizational learning: A theory of action approach." *Reading, MA: Addison Wesley*.

Bucciarelli, L. L. 1988 "An ethnographic perspective on engineering design." *Design studies* 9(3): 159-168.

Bucciarelli, L. L. (1994). *Designing engineers*. MIT press.

Buede, D. M. 2011. "The engineering design of systems: models and methods" (Vol. 55). John Wiley & Sons.

Brill, J. 1994. "Systems Engineering–A Semantics Jungle." *Systems Engineering Journal of NCOSE*, *1*(1).

Brown, J. S., Collins, A., & Duguid, P. 1989. "Situated cognition and the culture of learning." *Educational researcher*, *18*(1), 32-42.

Chase, W. P. 19794. "Management of Systems Engineering." Malabar, FL

Cole, R. 2008. "SoS architecture." *Systems of Systems Engineering Principles and Applications*, 37.

Chestnut, H. 1967. "Systems Engineering Methods." Wiley Series on Systems Engineering and Analysis

Clancey, W. J. 2009. "Scientific antecedents of situated cognition." *Cambridge handbook of situated cognition*, 11-34.

Détienne, F. 2006. "Collaborative design: Managing task interdependencies and multiple perspectives." Interacting with computers 18, no. 1 (2006): 1-20.

Dickerson, C., & Mavris, D. N. 2010. *Architecture and principles of systems engineering*. CRC Press.

Dong, A. 2005. "The latent semantic approach to studying design team communication." *Design Studies*, 26(5), 445-461.

Fisher, G. 2000. Social Creativity, "Symmetry of Ignorance and Meta-Design." *Knowledge-Based Systems Journal*, *13*(7-8), 527-537.

Forsberg, K., and Mooz H. 1991. "The Relationship of System Engineering to the Project Cycle." In First Annual Symposium of NCOSE. Chattanooga, US-TN.

Frenken, K. 2006. "Technological innovation and complexity theory. " *Economics of Innovation and New Technology*, 15(2), 137-155.

Gal, I. 2002. "Adult statistical literacy: Meanings, components, responsibilities*."* *International Statistical Review*, 70 (1), 1-25.

Gennari, J. H., & Reddy, M. 2000. "Participatory design and an eligibility screening tool." In *Proceedings of the AMIA Symposium* (p. 290). American Medical Informatics Association.

Gero, John S., and Mary Lou Maher. 1997. A Framework for Research in Design Computing. Paper read at Challenges of the Future: 15th eCAADe Conference Proceedings, at Vienna, AT.

Gibson, J. E., Scherer, W. T., & Gibson, W. F. 2007. *How to do systems analysis* (Vol. 47). John Wiley & Sons.

Greeno, J. G., Collins, A. M., & Resnick, L. B. 1996. “Cognition and Learning. Handbook of Educational Psychology.” 15-46.
Greeno, J. G., Benke, G., Engle, R. A., Lachapelle, C., & Wiebe, M. 1998. “Considering conceptual growth as change in discourse practices.” In M. A. Gernsbacher & S. J. Derry (Eds.), Proceedings of the Twentieth Annual Conference of the Cognitive Science Society (pp. 442–447). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
Greeno, J. G., and van de Sande, C. 2007. "Perspectival understanding of conceptions and conceptual growth in interaction." *Educational Psychologist*, 42(1): 9-23.
Haskins, C., ed. 2011. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Version 3.2.2. Revised by M. Krueger, D. Walden, and R. D. Hamelin. San Diego, CA (US): INCOSE.
Kolodner, J. L., Crismond, D., Gray, J., Holbrook, J., & Puntambekar, S. 1998. “Learning by design from theory to practice.” *In Proceedings of the International Conference of the Learning Sciences* (Vol. 98, pp. 16-22).
Lawson, B., & Dorst, K. 2013. *Design expertise*. Routledge.
Levis, A., 1993.*National Missile Defense (NMD) Command and Control Methodology Development*, Contract Data Requirements List A005 report for U.S. Army Contract MDA 903–88-0019, Delivery Order 0042. George Mason University, Center of Excellence in Command, Control, Communications, and Intelligence, Fairfax, VA.
Maier, M. W., & Rechtin, E. 2000. *The art of systems architecting* (Vol. 2). Boca Raton: CRC press.
NASA. 2007. *NASA Systems Engineering Handbook.* Washington, DC (US): National Aeronautics and Space Administration.
New York State Office for Technology. 2003. “The New York State Project Management Guidebook. 2nd Edition.” New York. http://www.cio.ny.gov/pmmp/guidebook2/index.htm
Olson, G. M., & Olson, J. S. 2000. “Distance matters.” *Human-computer interaction*, *15*(2), 139-178.
Rhodes, D., & Hastings, D. 2004. “The case for evolving systems engineering as a field within engineering systems.” In 1st MIT Engineering Symposium.
Rittel, H. W., & Webber, M. M. 1973. “Dilemmas in a general theory of planning.” *Policy sciences*, *4*(2), 155-169.
Ryschkewitsch, M., Schaible, D., & Larson, W. 2009, December. “The art and science of systems engineering.” *In Systems Research Forum* 3(2): 81-100. World Scientific Publishing Company.
Sage, A. P. 1992. Systems engineering (Vol. 6). John Wiley & Sons.
Schön, D.A. 1983. “The reflective practitioner: How professionals think in action.” *Basic books*, Vol. 5126.
Schön, D. A. 1988. “Designing: Rules, types and words.” *Design studies*, 9(3), 181-190.
Sheard, S. A. 2006. “Definition of the Sciences of Complex Systems.” *INSIGHT* 9 (1): 25.
Simon, H. A. (1969). *The sciences of the artificial* (Vol. 136). MIT press.
Turns, J. A., Sattler, B. and Borgford-Parnell, J. L... 2014 "Integrating Reflection into Engineering Education." *121st ASEE Annual Conference and Exposition*. Indianapolis, US-IN.

Ullman, D. G., Dietterich, T. G., & Stauffer, L. A. 1988. “A model of the mechanical design process based on empirical data: a summary.” *Artificial intelligence in engineering design*, 193-215.
US Department of Defense. 2003. Department of Defense Directive 5000.1. *The Defense Acquisition System.* Washington, DC (US): Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics.
Valkenburg, Adriana C. 2000 "The reflective practice in product design teams." Ph.D. diss., Delft University of Technology (Netherland)
Valkenburg, R., & Dorst, K. 1998. “The reflective practice of design teams.” *Design studies*, 19(3), 249-271.
Visser, W. 2004. Dynamic aspects of design cognition: Elements for a cognitive model of design. *Rapport de recherche de l'INRIA-Rocquencourt*, Equipe EIFFEL.
Zimring, C., & Craig, D. L. 2001. “Defining design between domains: An argument for design research á la Carte.” *Design knowing and learning: Cognition in design education*, 125-146.

Biography

Ali Shafaat is a PhD student at Lyles School of Civil Engineering at Purdue University in West Lafayette, IN (US). He received two master degrees in Industrial and Civil (Geotechnical) Engineering. He worked for more than six years in industry and academia before starting his doctoral studies. He was a lecturer at different higher education institutions where he has taught system engineering, geotechnical engineering, engineering economy, and industrial management. He is doing research in project governance systems, socio-technical systems and engineering education.

C. Robert Kenley is an Associate Professor of Engineering Practice in Purdue’s School of Industrial Engineering. He has over thirty years’ experience in industry, academia, and government as a practitioner, consultant, and researcher in systems engineering. He has published papers on systems requirements, technology readiness assessment and forecasting, Bayes nets, applied meteorology, and the impacts of nuclear power plants on employment.