122nd ASEE Annual Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Making Value for Society
Paper ID #12537

Integrating Core Systems Engineering Design Concepts into Traditional Engineering Disciplines

Rama N Reddy
Prof. Kamran Iqbal, University of Arkansas, Little Rock

Kamran Iqbal obtained his MS and Ph.D. degrees in Electrical Engineering, and MBA degree from the Ohio State University. He has held teaching and/or research appointments at the Ohio State University, Northwestern University, University of California, Riverside, University of California, Irvine, California State University at Fullerton, and University of Arkansas at Little Rock (UALR), where he currently serves as Professor of Systems Engineering. He earlier served as Assistant Chair of the Systems Engineering department and helped launch the Masters of Systems Engineering and PhD in Engineering Science and Systems programs at UALR. He has taught a variety of courses in systems and electrical engineering program and written a monograph titled 'Fundamental Engineering Optimization Methods (published by BookBoon.com).' His research interests include linear systems, biomedical engineering, biomechatronics, and computational intelligence. He is a member of IEEE, IET (UK), ASEE, IASTED, and Sigma Xi (former president of the Sigma Xi Central Arkansas Chapter). More information on Dr. Iqbal is available at syen.ualr.edu/kxiqbal/.

©American Society for Engineering Education, 2015

Integrating Systems Engineering Concepts in all Design Oriented Courses in the Engineering Curriculum

University of Arkansas at Little Rock, 2801 S. University Ave., Little Rock, AR 72204

Systems engineering is an interdisciplinary field of engineering that focuses on how to design and manage complex engineering systems over their life cycles. This study makes a case for introducing core systems engineering concepts in undergraduate courses across engineering disciplines. We argue that rapid advances in technology, increasing complexity of engineering projects, lack of protection in a globalized world, and the pressures of reduced time to market have all created a need for a greater understanding of systems engineering concepts, which must now be introduced at early stages of engineering curriculum. These practices would enhance preparedness and marketability of engineering graduates into the twenty-first century work force.

Systems engineering is a relatively new approach to the design of large and complex engineering systems. The engineering community and the aerospace industry initiated this new approach to the design of large systems in 1960s. As the systems became large and complex the conventional engineering design approach for such systems was not adequate. There were questions about the cost, schedule, and performance issues for the entire life cycle of such large systems. The primary issues addressed in this change of focus were to reduce cost, develop reasonable delivery schedules, and increase performance. The systems engineering approach not only considered the design phase of the system but also all other phases of the systems life cycle including development, deployment, upgrades, maintenance, disposal, and training. Thus, the systems engineering approach is considered as the concurrent engineering or the life cycle engineering approach. The industries involved in the design and deployment of large-scale systems estimate that the systems engineering approach contributes to a saving of 6% to 8% of the total cost as well as increased performance and reduced scheduled delivery time in the overall life cycle of the system.

The authors feel that the systems engineering approach needs to be introduced in engineering courses that involve the design of projects. Engineering systems design and analysis mostly requires the use of Newtonian mechanics including: solid mechanics, fluid mechanics, and laws of thermodynamics. The systems engineering approach prepares the students in design and development of engineering systems from the perspective of concurrent engineering or life cycle engineering. The design of complex systems such as automobiles, airplanes, space vehicles, power plants, and large ships may be used to give a feel to the students for the application of systems engineering design approach.

The systems engineering approach to design is to consider the entire life cycle phases of the system under consideration. The life cycle phases must be considered concurrently during the design process. The design of any system starts with the conceptual design where top level requirement or requirements must be specified. For software systems the top level requirements must include all the required inputs, controls, and outputs. For hardware systems the top level requirements must include all the inputs, controls, outputs, and mechanisms. These requirements must be thoroughly understood and iterated to make sure they are completely specified. This requires a thorough understanding of the system and its life cycle phases.

The next step is the preliminary design where the design and decomposition process is repeatedly iterated to make sure that the design is complete and thorough. In the systems engineering approach, there is abstraction in concepts at the top level; then, as the design and decomposition process continues to the next level the abstraction decreases and the details emerge. As the process of design and decomposition continues to further levels, the abstraction gradually disappears and details emerge. Once the design and decomposition process is complete the engineering team can see a clear picture of the system with all of its details. In the next design stage, the discipline engineering teams start designing the system configuration items, elements, components, small segments, large segments, top-level major segments, and the system.

Once the design process at configuration item level is complete, the integration process begins. During integration process the configuration items are integrated to build elements, these must be tested for verification to validate the elements. This process continues until all the segments and major components have been completely built and integrated to define the system. The verification process at each step of integration leads to the validation. The verification and validation process is necessary to make sure that the system is built right. The last step in the design process is to formulate an acceptance test to make sure the right system is built that satisfies the stakeholder requirements.

The system to be designed under consideration must be clearly and completely defined with stakeholder's requirements in mind. The stakeholders, the management team, and the systems engineering team together must consider the cost, schedule, and performance constraints. The feasibility studies will be conducted to make sure that the stakeholders, management, and the systems engineering team agree on the feasibility of developing the system under consideration. Based on the discussion of all the parties involved a very important document must be written; the document is called the stakeholder's requirements document (SRD).

Based on the SRD the systems engineering team translates its contents in engineering and technical terms. The system under consideration may require scientific and mathematical models. The systems engineering team design the system starting with the conceptual design, preliminary design, and the final design that satisfies the stakeholder's requirements. A design document called Engineering Requirement Document (ERD) is prepared. This document is prepared in consultation with the stakeholders, management, and the systems engineering team. This document must be approved by all parties involved in developing the system.

The decomposition, design, integration, and verification process is shown in Fig. 1. This represents the famous "Vee" model [1], where the decomposition and the design are shown by downward arrows, and the integration and verification process are shown by upward arrows. The decomposition, design and the integration and verification process is shown in the figure.

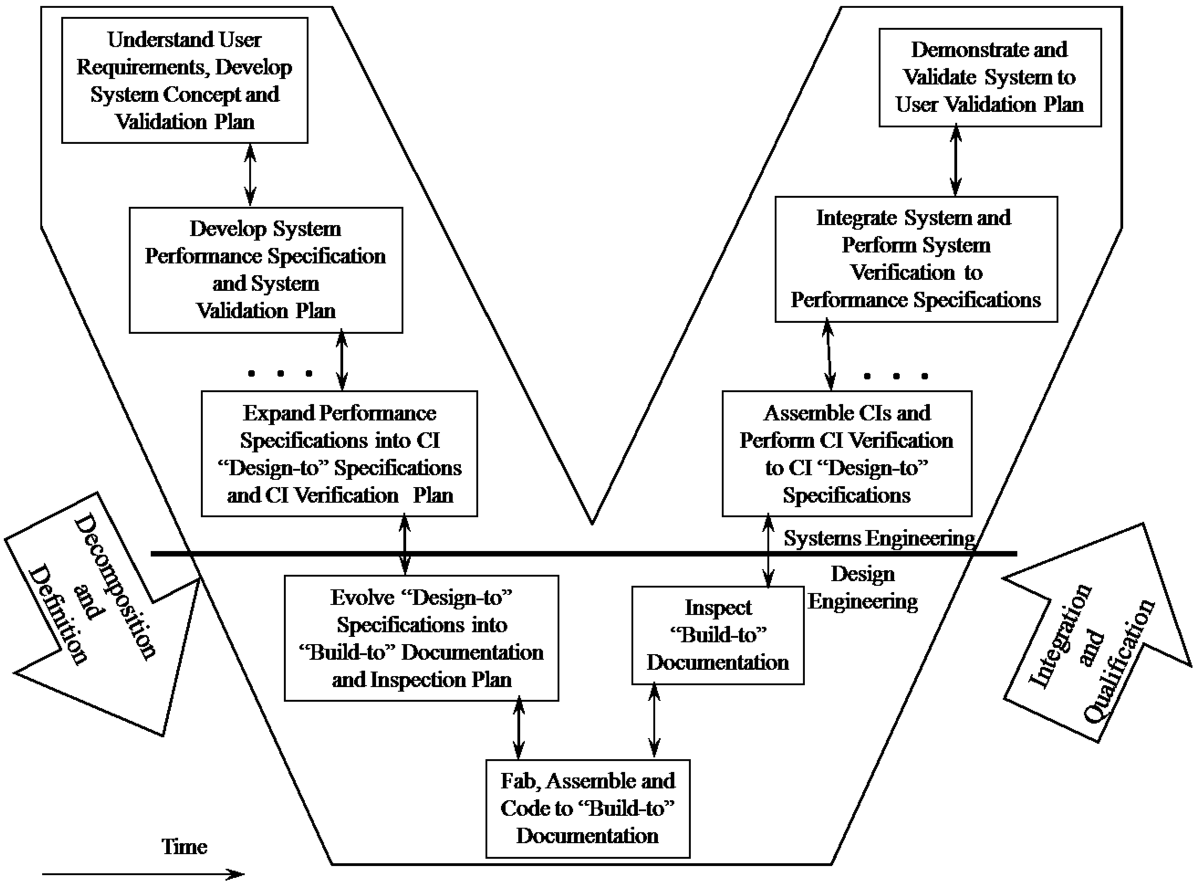

Fig. 1: The systems Engineering 'Vee' model [1]

The design process starts with the specification of all inputs, controls, outputs, and mechanisms at the top level to satisfy the system requirements. During the design process all of the "life cycle" phases must be considered; this is also referred to as "concurrent engineering" process. The design process starts with the development of functional architecture followed by the physical architecture, and then the allocated architecture. These architectures look as though they are separate and distinct, but they are not. All of these architectures must be considered in parallel to see how they complement each other in the design of an engineering system. Further, each of the three architectures must be iterated to develop a successful design. We will now consider individual architectures and how it contributes to the overall design process.

As stated above, the first step in the design process is to develop a functional architecture. At the top level of the functional architecture we state the need and the requirements for the system with the consultation of the stakeholders. Based on the need and the system requirements, we then figure out all the necessary inputs, controls, outputs, and mechanisms to achieve the stated need, purpose, and the requirements. This may require several iterations to make sure all of the possible inputs, controls, outputs, and mechanics are in place to design the system. At present the systems engineering uses the TTDSE (Traditional Top-Down Systems Engineering), a concept that originated from software engineering design and development.

In the functional architecture, we divide the top level engineering system into its major subsystems. The subsystems at this level are abstract except for the functionalities void of any details. This process of decomposition continues until all the abstractions gradually disappear and details emerge. The decomposition process results in major segments at the top level and these segments are further divided in sub-segments. The segments and sub-segments are further divided in the next level to elements and further into sub-elements. The elements are divided into components and sub-components. This process continues until the decomposition is complete and the components at the bottom level cannot be divided any further. These bottom level components called Configuration Items (CI's). These CI's cannot be divided any further. After the functional architecture has been completed, systems engineering design process is complete from the perspective of functional design. This is the first step in the design process from the perspective of application of system engineering design principles.

A generic template for representing the physical system architecture is presented in Fig. 2 [1]. This template suggests the creation of a generic partition of six system functions that are [2]:

- User interface: those functions associated with requesting and obtaining inputs from users, providing feedback that the inputs were received, providing outputs to users, and responding to the queries of those users
- Input Processing: those functions needed to receive inputs from external interfaces (nonhumans), and other nonhuman system components and to process those inputs to put them into a format needed by the system's processing functions
- The process model: transform inputs into outputs: the major functions of the system
- Control processing: those functions needed to control the processing resources or the order in which these processing functions should be conducted
- Output processing: those functions needed to convert the system's outputs into the format needed by the external interfaces or other nonhuman system components and then place those outputs onto the appropriate interface
- Provide structural support, enable maintenance, conduct self-test, and manage redundancy processing: those functions needed to perform internal support activities, respond to external diagnostic tests, monitor the system's functionality, detect errors, and enable the activation of standby resources

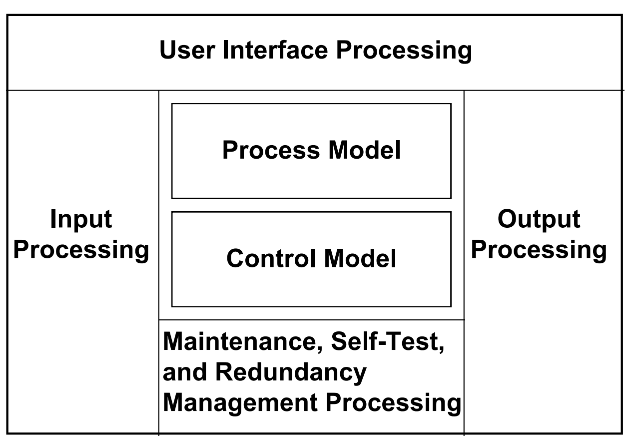

Fig. 2: The functional system architecture development model [1, 2].

Design Case Study

This case study is to illustrate the concept of functional architecture, which is the first step in the design of engineering systems. The case presented in this paper is the design of a "***lawn mower.***" The functional decomposition process begins with the specification of the top level requirement which is "***cut the grass***". This requirement be written in the top level box of the functional architecture. This top level box must contain the information about all the possible inputs, controls, outputs, and mechanisms. These requirements are derived from the complete and thorough analysis of all the input, controls, outputs, and mechanisms requirements. This document must be prepared by the systems engineering team together with the stakeholders involved in this design of this system. Figure 3 shows the top level box indicating the design requirement.

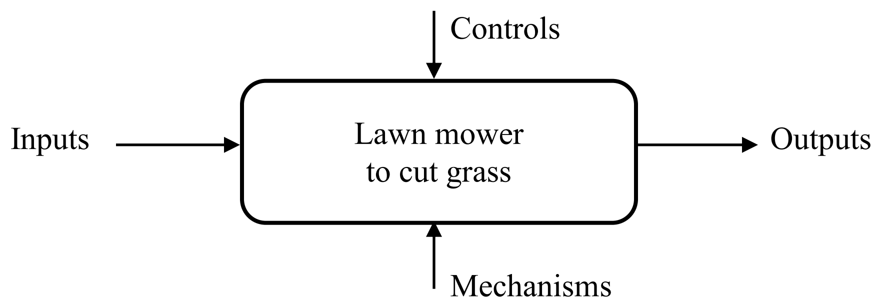

Figure 2: The top level requirement diagram for the design of a lawn mower

The top level functionality of the system is divided into the following functionalities that are further divided into sub-functionalities (Fig. 4):

1. Cutting functionality
2. The power requirements functionality
3. The moving functionality with the chasee and the wheels
4. The disposal functionality
5. The other function like the paint and so on

The next step in the design process is the physical architecture. In this step the functional architecture is translated or mapped into physical architecture. The first step in developing the physical architecture is to map the functions into physical entities that perform those functions. The best mapping is a one-to-one mapping. The physical architecture is a two-step process: the first step is to develop the generic physical architecture from the functional architecture; the second step is to develop the instantiated physical architecture.

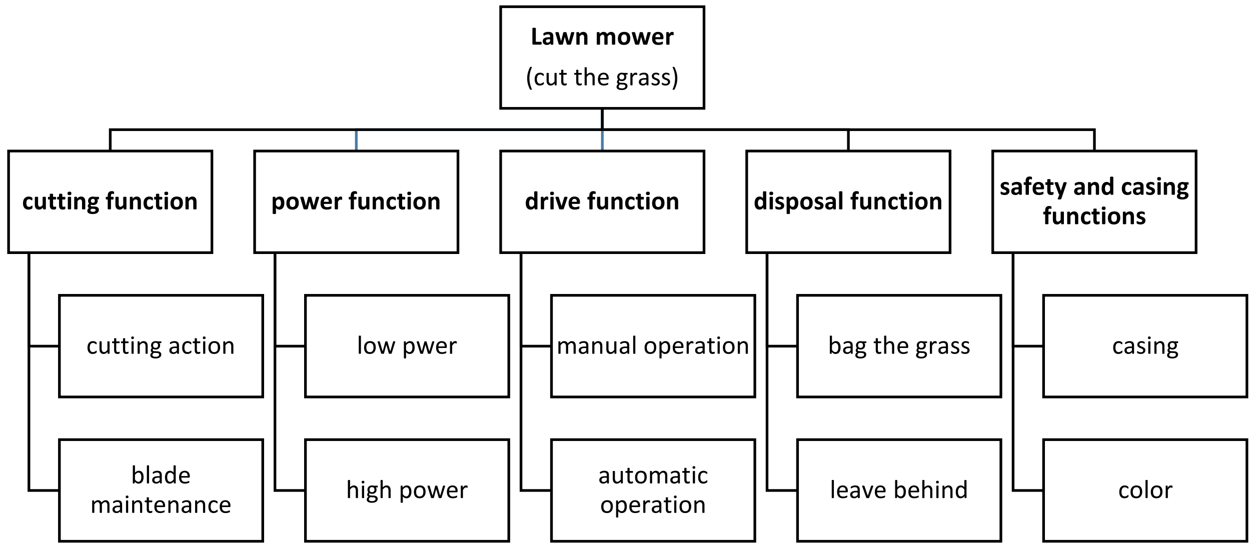

Figure 4: Functional architecture of the lawn mower

Each generic component or components, generic element or elements, and so on, are selected from several choices for each of the generic components based on the design criterion. This is accomplished by using morphological box. A morphological box is a two-dimensional matrix where the generic physical entities are represented in vertical columns and each of the instantiations for each generic entity in each column are represented in horizontal rows. This is a very convenient method for mapping the functions to physical entities.

The functionalities developed in the functional architecture (Fig. 4) for a lawn mower must be mapped into the physical entities. The functional architecture give the generic description of the physical entities corresponding to the functions. In this design of a lawn mower the functional design results in a generic physical architecture. There are different blade sizes, different engine powers to power the lawn mower, different chases, different moving mechanisms, different depth of cutting adjustments, different paint colors. The generic lawn mower system with different choices is represented by a morphological box as shown in Table 1.

Table 1: A morphological box for a lawn mower

Mover Type	**Blade Size (inches)**	**Engine Power (hp)**	**Depth of Cut (inches)**	**Paint Color**
Motorized mower	22	2.5	2	red
Manual mower	24	3.0	3	green
	28	3.5	4	black
	36	8.5		
	48	10.5		
		20.0		
#Choices: 2	5	6	3	3

These choices give us 540 combinations, but when you are looking for a lawn mower you only see a few choices. The engineers who design this lawn mower have to make decisions about the viable choices keeping in mind the end users, manufacturers, maintenance, and the weight of the mover in mind. For example, a viable choice for a manual lawn mower may be 22 inches to 24 inches blade size, a horse power of 3.0 to 3.5, a depth cutting adjustment of 2 to 3 inches, and a color choice of red, green, or black.

These two architectures must be developed in parallel to keep the checks and balances between the two architectures to eliminate the problems in the design of the system. The next step in the design process is to develop the allocated architecture where the physical entities are allocated to functions. This step must also be considered during the functional and physical architecture development to eliminate the problems that may show up during the allocation process of the allocated architecture. During the allocation process one must consider the interfaces between the hardware segments, elements, components, configuration elements. The design of interfaces is very critical to entire design process.

The interface is the physical or logical connection between the segments, elements, components. There are external interfaces and internal interfaces. In the design of engineering systems the system engineers are primarily concerned about the design of internal interfaces. The interfaces must be considered during the development of the functional architecture. The functions necessary for the interfaces must be considered during the development of functional and physical architectures. The interface is not only the physical connection, but also communication and transportation connection among subsystems. Through the interface connection there is material flow, energy flow, information flow, and data flow. Interfaces are critical for the system to function properly, effectively, and economically. Interfaces must be loosely coupled so that the components, element, and segments can be easily replaceable when they malfunction, or get damaged during the operations. During the transportation of energy, materials, information, and data the interfaces must deliver them in the specified time without modification.

To recapitulate, there are four major steps in developing a system. These are: functional architecture, physical architecture, allocated architecture, and interface architecture. The first step is to develop the functional architecture. During the development of functional architecture the top level requirement must be specified based on the need of the stakeholder or the customer.

There are two major steps in developing the physical architecture. First, one should develop the generic physical architecture in general terms, and then develop the instantiated physical architecture with details for each of the generic physical entities. This is accomplished through the morphological box where each generic component is selected from several instantiated hardware elements, For example aircraft wing is generic component and the instantiated wing is selected based in the options available and requirements of the aircraft under wing. The options available in the morphological box may be selected based on the requirements such as the size, lift, and other characteristics. This is done in the allocated architecture. The interfaces must be considered by doing the interface architecture.

Integrating Systems Engineering Methodology in Engineering Education

The systems engineering design process offers a methodology that is broadly applicable to all engineering disciplines. Thus, the authors' view is to introduce system engineering design methods and principles in engineering courses offered to mechanical, electrical, civil, chemical, and petroleum engineering students. This is particularly important in senior design projects and other design oriented courses. These include courses such as machine design, design of thermal and fluid systems, design of boilers, design of gas and water turbines, electrical generators, aircraft engines, rocket engines, engines for large ships and submarines, the design of large heating, air condition systems.

The purpose of engineering education is to graduate engineering students who can design [4]. The systems engineering design and analysis need to be integrated in all engineering courses. Further, students must be able to apply the system engineering design concepts and methods in all design related courses as well as in their senior design projects. The capstone design experience is extremely useful for the students as they prepare to enter the workforce and work in the industry. The senior design project is a two semester requirement for undergraduate students in most engineering programs. During the first semester they must be exposed to the design process to design their projects by using the system engineering design principles. They should get the experience in developing the functional, physical, allocated, and interface architectures. The design as well as the system architecture must be completed during the first semester. In the second semester they should be able to implement the design and develop the system they have designed.

Over the past several years, the authors have made consistent efforts to integrate systems engineering concepts and practices into the design oriented courses in our undergraduate engineering curriculum. Our university offers an undergraduate systems engineering program that has four main components: a university core (humanities and social sciences, 21 hours), a college core with additional science and mathematics courses (30 hours), a systems engineering core component (32 hours), and an option core (32 hours). The systems engineering core includes courses on probability and random signals, discrete event simulation, engineering optimization, decision and risk analysis, engineering economy, and systems engineering design and analysis. The core also includes a two course capstone design sequence taken in the senior year. The option core is geared towards depth in the students' chosen area of emphasis, i.e., electrical systems, mechanical systems, or computer systems.

One of the authors teaches courses in the mechanical systems option, including Introduction to Mechanical Engineering, Engineering Thermodynamics, Machine Design, and Fluid Mechanics. He also teaches a required graduate course on Systems Design and Analysis and Systems Architecture and Design that focuses on the systems engineering design and system architecture process. He regularly uses examples from industry to teach these concepts. Students carry out team projects that reinforce their understanding of systems engineering architecture and design process. He also introduces systems engineering design examples in the mechanical engineering courses he teaches.

The other author teaches courses in the electrical systems option, including Introduction to Electrical Engineering, Circuits and Systems, and Control Systems Design. He also teaches core systems engineering courses on Probability Models and Optimization Methods in Systems Engineering. He introduces systems engineering design concepts with the help of design examples in these courses. Those concepts are later reinforced in the capstone design projects in their senior year. The capstone design teams undertake industry sponsored projects that are reviewed and judged by our systems engineering industrial advisory board members. These authors have many years of teaching experience in the areas of mechanical and electrical engineering disciplines. The authors think that in designing large mechanical and electrical systems the concepts of design of large systems must be introduced in courses such as thermodynamics, fluid mechanics, and electrical machines. A simple design of a thermal power plant or a hydroelectric unit can be introduced in the basic courses to drive home the significance of systems engineering principles in design.

In conclusion, we believe that knowledge of the systems engineering design process is applicable and useful to all engineering disciplines. The technological developments of the last few decades have confirmed that this knowledge is essential to prepare engineering graduates to succeed in the twenty-first century workplace. These concepts can be taught and reinforced through examples in successive discipline-related courses using a variety of methods. Our experience has shown that engineering courses at all levels should include design examples that illustrate these concepts. Our efforts to impart this knowledge to our engineering graduates have been duly appreciated and commended by our employers. We submit that engineering instructors at other institutions can likewise benefit from our experience.

References:

[1] Dennis M. Buede. The Engineering Design of Systems: Models and Methods, 2nd Ed. Wiley: Hoboken, New Jersey, 2009.
[2] D.J. Hatley, I.A. Pirbhai. Strategies for real-time system specification. Dorset House, New York, 1988.
[3] C.L. Dym, A.M. Agogino, O Eris, D.D. Frey, L.J. Leifer. Engineering design thinking, teaching, and learning, Journal of Engineering Education, pp. 103-120, Jan. 2005.