4th Asia-Pacific Conference on Systems Engineering (APCOSE 2010)
Keelung, Taiwan October 4-6, 2010

Functions: The teaching language for systems engineering

Joseph Kasser D.Sc, CM, CMALT
Visiting Associate Professor
Temasek Defence Systems Institute
National University of Singapore
Block E1, #05-05
1 Engineering Drive 2, Singapore 117576
Email: joseph.kasser@incose.org

ABSTRACT

Early stage systems engineering operates in the conceptual and functional domain. This paper describes how thinking 'functions' for as long as possible in the system lifecycle can produce systems that are innovative and more complete than thinking 'implementation'. The paper discusses developing a standard functional template for a system from which it should be possible to develop a set of reference functions for any class of system. Then, by using the inheritance concept from the object-oriented paradigm, it should be possible to inherit functions from the set of reference functions for the class of system being developed.

The paper discusses examples of the advantages of using functions to determine the completeness of a system

1 INTRODUCTION

People tend to use implementation language to describe functions. For example, we often use the phrase "need a car" when we should be using "need transportation". Using implementation language in the early stages of a system lifecycle tends to produce results that may not be the best solution to the problem even if it is a complete solution, as well as generally not being innovative. This is because implementation language tends to turn examples into solutions with little exploration of alternative solutions. This was one of the defects identified in the Operations Concept Harbinger (Kasser, et al., 2002). We need to train systems engineers to think abstractly. One way to achieve that goal is to use the language of functions instead of implementation or solution language. Problems and solutions should be described in terms of functions (Hall, 1989). For example a problem might be stated as "provide a transportation function to move N people with B (pounds/cubic feet) of baggage M miles in H hours over terrain of type T with an operational availability of O.

2 SYSTEMS THINKING PERSPECTIVE OF SYSTEM DEVELOPMENT LIFECYCLE

Consider the big picture perspective of the context for a system development life cycle shown in Figure 2. This figure takes the form of a causal loop linking the set of systems in operation and under development. One starting point is the gap analysis where a new need is considered in the light of existing capability and capability currently under development. In the early stage systems engineering performed in column A of the Hitchins-Kasser-Massie framework (HKMF) for understanding systems engineering (Kasser, 2007a, b), an interdisciplinary team perfroms a gap anaylsys between the need and the existing capability and capability currently under development examining the issues and creating functional and operation representations of both the problem and alternative solutions.

When the gap analysis is complete and the needed functionality has been identified, a number of alternative choices for implementation are developed, evaluated, perhaps adjusted and a preferred implementation is selected. The preferred implementation choice can range from a brand new system (the currently taught systems engineering systems development lifecycle paradigm) to an arrangement or rearrangement of existing capability (a system adaption lifecycle[1]). Other implementation choices include a combination of new (to be developed) and existing capability or non technical solutions such as changes in doctrine and staffing. If the implementation choice is a brand new system, then the acquisition can be mapped into columns B to F of the HKMF; the traditional linear system development lifecycle shown shaded in Figure 2. This linear representation can now be seen as a portion of the whole loop shown in the figure. The figure also shows other systems under development which should lead to an understanding of the need to coordinate the development of the set of systems under development to ensure interoperability as and when they come into service. This view of the system acquisition lifecycle is non-linear and seems to be a better representation of reality than the current linear models.

Creating the solution concept in the form of capability or functionality is in accordance with Hall who stated that if a problem can be stated as a function, then the total solution is the needed functionality (Hall, 1989) as well as the process to produce that functionality.

3 FUNCTIONS AS COMPONENTS

A generic function is pictured in Figure 1 as a jigsaw piece because systems can be represented by a number of functions fitted together to perform the function of the system. Generically a function converts inputs to outputs using resources. These outputs include:

- Desired known outputs.
- Undesired outputs such as high frequency oscillations that can be predicted.
- Serendipitous – unknown at design time but later found to be useful, e.g. performance beyond specifications.
- Waste.
- Defects.
- Undesired excess capability that shows up later. E.g. Latent defects, unpredicted high frequency oscillations, etc.

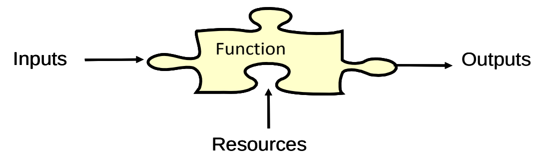

Figure 1 Generic Function

A system can be considered as being constructed of a combination of series and parallel functions. Some of the functions can be complicated and consist of a combination of sub-functions (hierarchical perspective).

4 A STANDARD TEMPLATE FOR A SYSTEM

Ensuring that a design is complete constitutes a significant problem in systems engineering. The use of a standard or reference set of functions in the form of a template that can be tailored to describe a specific system can ensure a greater probability of design completeness as described in this section. One way of grouping the complete set of functions performed by any system is into the following classes:

- **Mission** – the functions which the system is designed to perform to provide a solution to the problem as and when required.
- **Support** – the functions the system needs to perform in order to be able to perform the mission as and when required. Support functions can further be grouped in accordance with (Hitchins, 2007) pages 128-129) into:
  - **Resource management** – the functions that acquire, store, distribute, convert and discard excess resources that are

---

[1] This cycle is generally thought of an one in which there is no perceived need for the system until after the system has been created. It generally takes places in Layer 1 (the product layer) of the HKMF. Products produced via the adaption cycle are the telephone, the phonograph and the Sony Walkman.

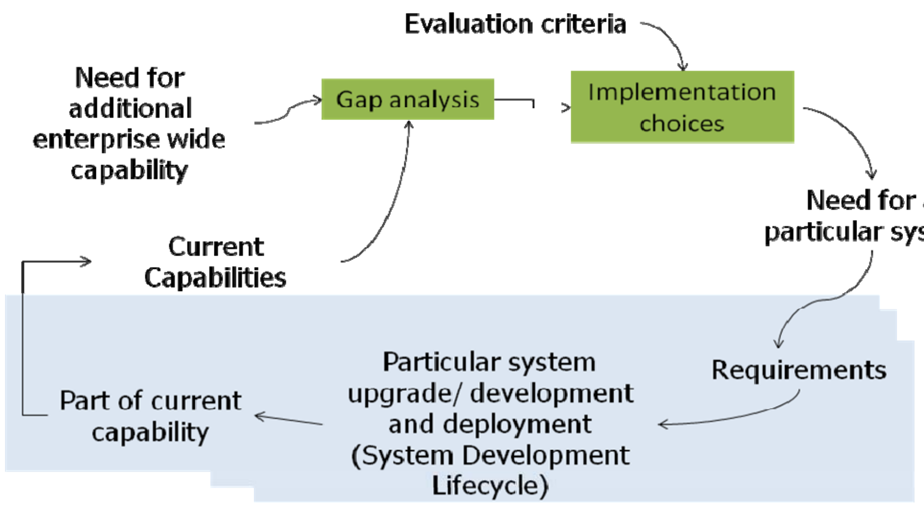

Figure 2 Systems thinking perspective of the system life cycle

utilized in performing the mission.

- **Viability management** – the functions that maintain and contribute to the survival of the system in storage, standby and in operation performing the mission.

From the temporal perspective (Kasser and Mackley, 2008), at the time the problem is being identified, the implemented solution (at sometime in the future) consists of two sets of functions;

- **the mission and support functions** to be performed by the solution system once realized (F_s) and,
- **the process functions** which have to be performed to realize the solution system (F_w).

This concept can be represented by the following relationships:

Total solution (S) = $F_s + F_w$ ....[1]

where

$F_s = F_d - F_c$ ....[2]

Which gives?

$S = (F_d - F_c) + F_w$ ....[3]

In summary,

F_s = functions performed by solution system.

F_w = functions [needed to be] performed to realize the solution system (create the desired functions that do not exist at the time the project begins. Fw constitute the unique systems engineering and non-systems engineering development processes (Biemer and Sage, 2009)

F_d = complete set of desired functions to be developed.

F_c = complete set of current functions; functionality provided in the existing situation which may range from zero (nothing exists) to some functionality in an existing system deemed as not providing a complete solution.

Moreover, the F_s and F_w can both consist of mission and support functions as discussed above.

4.1 Template examples

This template applies in all five layers of the HKMF. For example:

4.1.1 HKMF Layer 2 transportation.

The problem is to provide a transportation function to move N people with B (pounds/cubic feet) of baggage M miles in H hours over terrain of type T with an operational availability of O. The solution:

- Product functions are converted to the physical realization of the function.
- Process functions are a tailored sequence of tasks from the activities in HKMF Layer 2 columns A-G.

4.1.2 HKMF Layer 3 personnel situation.

The situation (human system) was an awareness that there was something wrong in a research group at an Australian government department (Kasser, 2000). The problem was to provide an understanding of what was wrong. The solution:

- Product functions were an understanding of what is wrong and recommendations for changing the situation (in functional language). The tangible physical representation was a PowerPoint presentation to the department.
- Process functions were a tailored set of Soft System Methodology (Checkland and Scholes, 1990) tasks in the correct sequence to provide the understanding.

4.1.3 HKMF Layer 3 coastal patrol situation.

The problem was to provide a multinational naval task force patrolling a coastal area to prevent and react to piracy. The solution:

- Product functions were converted to the physical implementation of an operational multinational naval task force with full logistics support.
- Process functions were a tailored set of the generic activities used to create and operate a naval task force comprising systems provided by the countries involved.

5 SOME ADVANTAGES OF THE USE OF FUNCTIONS

The use of functions has a number of advantages including the following as discussed below.

- Forces systems engineers to think in abstract terms in the early stages of identifying a problem and providing a solution.
- Using a standard functional template for a system can help maximize completeness of the resulting system.
- It is easier to identify missing functions in system functional descriptions than in implementation descriptions.
- A system in its functional form can be modelled at design time to determine how well the solution functionality solves or resolves the problem.

Consider the implications of these advantages.

5.1 Abstract thinking

As stated above, people tend to use implementation language to describe functions. Using the language of functions in the early stages of systems engineering will nudge the stakeholders into abstract thinking rather than fixating on an implementation.

5.2 The use of a standard functional template for a system

Based on this premise, it should be possible to develop a set of reference functions for any class of system based on the big picture systems thinking perspective (Kasser and Mackley, 2008) template shown in Figure 3. Then, by using the inheritance concept, it should be possible to inherit functions from the set of reference functions for the class of system being developed. This seems to be the concept behind Hitchin's Generic Reference Model of any system (Hitchins, 2007) pages 124-142). Systems engineers and application domain experts working together[2] would assemble the detailed functions to be performed by the solution system from the set of reference functions for the class of system being developed, tailoring the functions appropriately.

For example, if the solution is a

- **Spacecraft**, then the support functions for surviving launch and the out-of-atmosphere environment would be among those inherited.
- **Information system**, then the functions displaying information to ensure that data is not

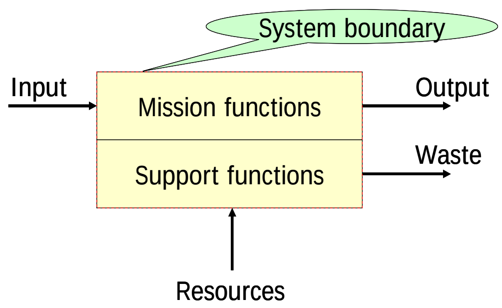

Figure 3. Functional template for system

[2] In an integrated team.

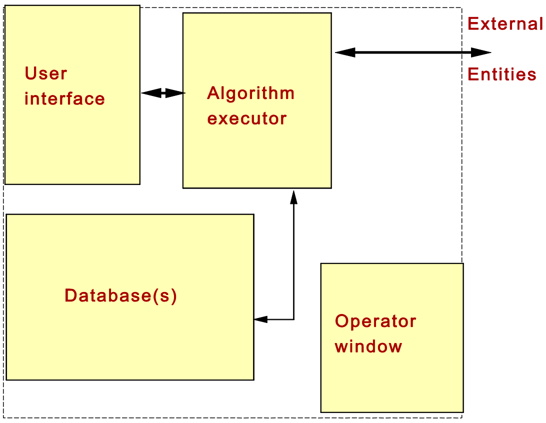

Figure 4. Functional representation of a data processing system

hidden due to colour blindness in the operators would be among those functions inherited. In addition, a typical template for data processing system might look like that shown in Figure 4 (Kasser, 1997). This method for developing a system design should decrease the number of missing functions.

An immediate advantage of the Figure 3 big picture perspective template is that one can see that there are two system outputs, the desired output labelled 'output' and another output labelled 'waste'. Process improvement and cost reductions efforts tend to focus on the effectiveness of producing the desired output. This perspective shows that there is also a need to focus on reducing the amount of waste; or exploring ways in which 'waste' can be used or sold to someone who wants the waste. For example waste heat might be sold or used; waste food in a restaurant might be used for fertiliser or animal food and so on.

5.3 Identification of missing functions

Even without a standard function template, this method for developing a system design should decrease the number of missing functions and allow innovative designs as shown in the following examples:

- An automatic bank teller machine (ATM)
- The Wright brothers' heavier-than-air flying machine
- The Singapore high school system boundary

5.3.1 An automatic bank teller machine (ATM)

An automatic bank teller machine (ATM) is a simple system to which most people can relate, having had some interaction with such machines. The mission functions for which the ATM was designed include:

- withdrawing funds,
- depositing funds,
- checking the balance in the user's accounts.

The support functions for the ATM include:

- removing and replenishing bank notes,
- deterring theft,
- countering attempted theft.

By considering the concept of operation for each of the functions of the ATM and how the functions can be implemented, alternative innovative design approaches may be identified for implementation sometime in the future. For example the physical realization of the ATM uses 1980's technology and is based on the use of a plastic card containing a magnetic memory and a user supplied password. Thirty years since the initial implementation alternative implementations have become possible based on thumb prints, laser retina scans, cell phones and other techniques to ensure security without the need for the plastic card. The functional approach allows the systems engineer to pose the following question. *"Is the financial industry trading off the costs of developing and deploying newer more secure ATMS against the cost of the losses incurred from the current generation of ATMs?"*

5.3.2 The Wright brothers' heavier-than-air flying machine.

The functional representation shown in Figure 5 elaborates the transportation function into three sub-functions. Each of the sub-functions is further elaborated into three sub-functions. The thrust function applies to horizontal motion, the lift function applies to vertical motion and the control function applies to the control of thrust and lift. When thinking about how the vehicle moves, it is easy to notice the lack of a function that moves the vehicle sideways in the figure[3]. If the transportation by air

[3] However to be fair, that function may not have

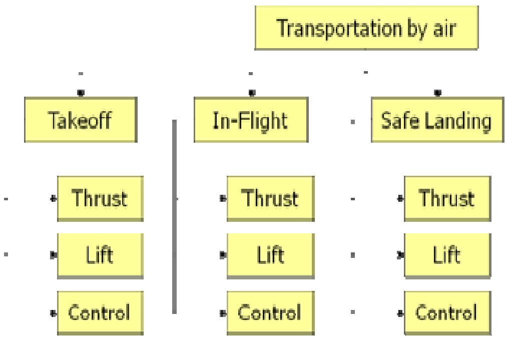

Figure 5. Wright Brothers' Heavier-than-air Flying Machine Functions

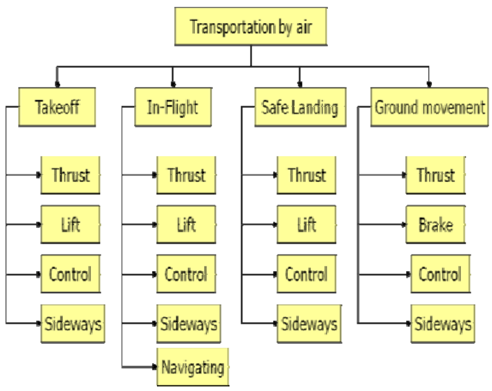

Figure 6 Additional functions to travel some distance

problem is expanded to include travel from Point A to Point B, where points A and B are separated by some distance, the functional representation might look like that shown in Figure 6. During the discussion of the concept of operations for the transportation function, the lack of a function performing communications with the ground should become apparent.

5.3.3 The Singapore high school system boundary

Consider the photograph shown in Figure 7 and answer the question "*what is missing from the picture?*" The photograph shows a portion of the system boundary and the missing function is masked by the detail in the picture. A functional view would show that the function that allows the students to cross the road safely is missing. Or is it?

The use of the figure illustrates important concepts, namely, how stating the problem influences the solution and the position of the system boundary. Is providing the safety of students who arrive and depart by bus a part of the problem for which the high school is the system providing a solution? If it is, then the pedestrian crossing is missing from the picture. If it is not, then the pedestrian crossing is still missing but is a missing part of an adjacent system; the transportation system which transports the students between their homes and the school. The location of the system boundary at the gate in the fence, or at the bus stop on the other side of the road will depend on the answer to the question. These types of issues can be seen more clearly in the functional views of the type shown in Figure 5 and Figure 6 than in the physical view shown in Figure 7.

6 INCORPORATING THE EFFECT OF CHANGE IN THE DEVELOPMENT MODEL.

The current functionality (F_c) provides a baseline from which to transition to the solution system. Once the concept of baselines is accepted, then evolutionary development can be represented as a progression that converges the solution with the problem via intermediate versions of the solution system, each of which provides a temporary baseline from which the subsequent version of the solution system is developed; namely the cataract methodology (Kasser, 2002).

Viewing the system development as a change from current functions to desired functions evolving through a number of baseline milestones allows the system lifecycle model to incorporate change and the effects of change. The same view is valid for several situations including

- upgrades during the in-service phase of the system lifecycle.
- changes in the nature of the problem that may occur while the solution system is under devel-

---

been needed. The Wright brother's heavier-than-air flying machine was a solution to the problem of flying *per se* for a short distance using a heavier-than-air flying machine. No intentional sideways movement was needed; in the air, the function was not required, and on the ground the function was provided by human effort expended in moving the machine.

Figure 7 Entrance to a High School in Singapore

opment. In such a situation, the solution development should be broken out into a number of planned baselines providing temporary solutions where the time allocated to developing each baseline should be shorter than the interval between the changes in the nature of the problem. This is an implementation of the cataract development model (Kasser, 2002) and a functional representations of what has become known as agile systems engineering.

7 MODELED AT DESIGN TIME

Since a function by definition transforms an input to an output (consuming resources and generating waste), functions can be expressed mathematically which allows them to be used in simulations. If the set of functions is complete, then calculations can be made to determine how well the proposed solution system will meet the need at design time before the functions are allocated to the components in the implemented version of the system. When the model underlying the calculations is developed, variable and parameters that are not understood or are vague become candidates for risks to be monitored during the development.

8 PARTITIONING SYSTEMS INTO SUBSYSTEMS

Partitioning systems (physical and functional) should be performed using elaboration not decomposition (Hitchins, 2007). The rules for elaboration are:

1. Subsystems may contain mission and support functions.
2. Maximize cohesion and minimize coupling between subsystems (Ward and Mellor, 1985).
3. The maximum number of subsystems at any level should generally be no more than seven (±1) to comply with Miller's Rule to facilitate understanding the system (Miller, 1956).
4. Design for self-regulating or homeostatic subsystems at the higher levels of elaboration (Kasser, 1997).
5. Abstract (hide) non-relevant information at any level of partitioning to facilitate understanding the system. Note how Figure 3 does not show details of the mission and support functions. These details belong in lower level drawings even though their inclusion might not violate Rule 3 above.

9 SUMMARY

Early stage systems engineering operates in the conceptual and functional domain. This paper has shown how systems thinking can provide a more realistic representation of the systems acquisition and development lifecycle and described some examples of how thinking 'functions' in the various phases of the system lifecycle can produce systems that are innovative and more complete than thinking 'implementation'.

10 BIOGRAPHY

Joseph Kasser combines knowledge of systems engineering, technology, management and educational pedagogy. Having been a practicing systems engineer and engineering manager since 1970 in the USA, Israel, and Australia he brought a wealth of experience and a unique perspective to academia in 1997. He has since become internationally recognised as one of the top systems engineering academics in the world. He is an INCOSE Fellow, the author of “A Framework for Understanding Systems Engineering”, “Applying Total Quality Management to Systems Engineering” and many INCOSE symposia papers. He is a recipient of NASA’s Manned Space Flight Awareness Award (Silver Snoopy) for quality and technical excellence for performing and directing systems engineering and the recipient of many other awards, plaques and letters of commendation and appreciation. He holds a Doctor of Science in Engineering Management from The George Washington University, is a Certified Manager and a certified member of the Association for Learning Technology. He gave up his positions as a Deputy Director and DSTO Associate Research Professor at the Systems Engineering and Evaluation Centre at the University of South Australia in early 2007 to move back to the UK to develop the world’s first immersion course in systems engineering as a Leverhulme Visiting Professor at Cranfield University. He is an INCOSE Ambassador and also served as the initial president of INCOSE Australia and as a Region VI Representative to the INCOSE Member Board. He is currently a principal at the Right Requirement Ltd. in the UK and based in Singapore as a Visiting Associate Professor at the National University of Singapore.

11 REFERENCES

Biemer, S. M. and Sage, A. P., "Systems Engineering: Basic Concepts and Life Cycle," in *Agent-Directed Simulation and Systems Engineering*, L. Yilmaz and T. Oren (Editors), Wiley-VCH, Weinheim, 2009.

Checkland, P. and Scholes, J., *Soft Systems Methodology in Action*, John Wiley & Sons, 1990.

Hall, A. D., *Metasystems Methodology. a new synsthesis and unification*, Pergamon Press, Oxford, England., 1989.

Hitchins, D. K., *Systems Engineering. A 21st Century Systems Methodology*, John Wiley & Sons Ltd., Chichester, England, 2007.

Kasser, J. E., "*Yes Virginia, You Can Build a Defect Free System, On Schedule and Within Budget"*, proceedings of INCOSE 7th International Symposium, Los Angeles, CA, 1997.

Kasser, J. E., "C3I Operational Analysis Group Study Report (PowerPoint Presentation)," Systems Engineering and Evaluation Centre, University of South Australia, Mawson Lakes, South Australia, 2000.

Kasser, J. E., "*The Cataract Methodology for Systems and Software Acquisition"*, proceedings of SETE 2002, Sydney Australia, 2002.

Kasser, J. E., "*The Hitchins-Kasser-Massie (HKM) Framework for Systems Engineering"*, proceedings of the 17th International Symposium of the INCOSE, San Diego, CA., 2007a.

Kasser, J. E., "*A Proposed Framework for a Systems Engineering Discipline"*, proceedings of The Conference on Systems Engineering Research, Hoboken, NJ, 2007b.

Kasser, J. E., Cook, S. C., Scott, W., Clothier, J. and Chen, P., "*Introducing a Next Generation Computer Enhanced Systems Engineering Tool: The Operations Concept Harbinger"*, proceedings of SETE 2002, Sydney Australia, 2002.

Kasser, J. E. and Mackley, T., "*Applying systems thinking and aligning it to systems engineering"*, proceedings of the 18th INCOSE International Symposium, Utrecht, Holland, 2008.

Miller, G., "*The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information."* The Psychological Review, Vol. 63 (1956), 81-97.

Ward, P. T. and Mellor, S. J., *Structured Development for Real-Time Systems*, Yourdon Press, 1985.