A Collaborative Systems Architecting Framework for Capability-based Acquisition

Dr. Santiago Balestrini-Robinson, SENIOR RESEARCH ENGINEER  
Electronic Systems Laboratory, Georgia Tech Research Institute

We need to move away from the traditional Systems Engineering methods, and recognize the intrinsic complexity of the problems we face. Rather than following the linear functional decomposition and system aggregation approach, what is required is (1) to capture the portions of knowledge distributed throughout the multidisciplinary community of subject matter experts, (2) integrate them into a cohesive environment that (3) allows decision makers to evaluate multiple alternatives and identify shortcomings in (a) the proposed solutions and (b) the knowledge captured by the framework. By re-evaluating traditional systems engineering concepts, the proposed framework addresses the issues associated with rigid functional decomposition and system aggregation. The proposed framework is a process for the integration of expertise and experience into a database structured framework that (1) allows users to test the proposed solutions and (2) enables an integrated knowledge-base to support the automatic generation of alternatives. The framework implementation is illustrated with a proof-of-concept requirements definition problem for a modern naval surface combatant.

# Motivation

We attempt to simplify the problems we face by linearizing them and arbitrarily bounding them. For example, when we study a complex system-of-systems, the levels of functionality (i.e., capability, activity, and function) are rigid and arbitrarily chosen. They are, an abstraction of the functional realities, which are not bounded by hierarchies with tree structures with distinct levels, and these abstractions often limit what we know and observe.[[1]](#footnote-1) After decomposing to the lowest levels of functionality, we then proceed to select subsystems, systems and groups of systems in an effort to satisfy the different levels of required functionality. In reality, these arbitrary levels of functionality (e.g., capability, ability, function) are not clearly delineated and oftentimes the relationships between them cannot be captured by the strict abstraction rules. Furthermore, the selection of the solution chosen (i.e., the systems chosen to perform those functions), can lead to added required functions. These added functions must in-turn be satisfied by other solutions, which may require other functions to be performed. The epitome of these are commonly referred to as Wicked Problems, i.e., problems that cannot be clearly formulated and can only be better understood through the iterative formulation and testing of solutions.

## Wicked Problems

Wicked problems are not intrinsic to a “problem,” but to our understanding. Every time we are faced with an unprecedented phenomenon we are faced with a wicked problem. As a matter of fact, the scientific method follows said approach, i.e., an observation is made, a hypothesis is formulated—i.e., a proposed answer to the perceived problem—and subsequently tested. Based on the testing of the hypothesis, we determine if the hypothesis must be refined or can be accepted for the time being. As we iterate over this process, we will gain understanding on the problem, and allow us to improve our formulation of it, but we cannot begin to comprehend the problem by decomposing it, we must propose perceived solutions and assess why they are not satisfactory in order to understand the problem.

Wicked problems are not intrinsically wicked, what today is perceived as a wicked problem, it may very easily be a well understood phenomenon in the future. They are wicked because we lack the tools to characterize them, to capture their intrinsic elements. We cannot understand them because we lack the abstract concepts to comprehend them, in essence, we lack *the models*.

Many of today's wicked problems are characterized by complexity, i.e., an intractable number of interdependent elements that interact nonlinearly. It has been demonstrated through experimentation that humans cannot easily predict the behaviors of a large number of interdependent elements whose interactions are nonlinear, even when presented with a large set of data.[citations of psychology studies on the topic] For this reason, we resort to the use of tools to support us in keeping track of these nonlinear interdependencies. The primary field for addressing these large problems with intricate interdependencies is *Systems Engineering*.

## Systems Engineering

Systems engineering is a broad field and is not a particular framework or process. In general, it focuses on supporting decision makers tasked with organizing and preparing a solution to an engineering problem. Traditional systems engineering methods propose a series of steps, which may be iterated, but attempt to understand a problem and then reconstitute it in order to formulate a solution, i.e., functional decomposition and system aggregation.



Figure 1. A systEms engineering Process for Capbility-based Assessments.

Put a waterfall chart with capability, activity, function, etc. going down, and next to it a network of interdependencies. The associated text should explain that truly complex problems cannot be made to fit the rigid hierarchies that we usually pursue in systems engineering.

# Background

There are a number of related efforts in the field of systems engineering and, more generally, decision support.

## Knowledge-Based Analysis and Design

Knowledge-based Analysis and Design, or KBAD, is a methodology that aims at providing decision makers with a cost-effective decision making framework.[Dam, 2009]

KBAD is DODAF centric and it still promotes a hierarchy.

Borrow the idea of integrating SME knowledge into a knowledge base.

## de Tenorio's

Focuses on the modeling and simulation using models of the physics and performance of the system. Focuses on the architecting of aircraft power systems.

Borrow the idea of induced functions.

# Proposed Framework

Given that we recognize that today's wicked problems are generally composed of large number of elements that interact nonlinearly, the

What is a capability, what is an ability? What is a system, what is a sub-system, what is a system-of-systems? These are all dependent on the scope of the question. In trying to tackle an unknown problem we must remain unbounded by these temporary problem scopes, otherwise, the models we create will become irrelevant when we evaluate the problem at a higher level or dwell in the detailed intricacies.

We can categorize the functions and the systems that perform them to facilitate our navigation of the knowledge-base, but we need the flexibility categorize not to a single level, but to a number of different groupings. For example, a function may belong to a category called capabilities because at that point in time it was relevant to the problem understanding, but as understanding improves, a more specific category may be created, e.g., ship capabilities, and the function can then also be mapped to that. With this approach the user can navigate

## Defining the functions

* Functions have to be defined clearly so other users can understand them clearly and be able to link their own functions and systems to them.
* Functions cannot reference a system or class of system in their name, unless they are indicating a specific functionality requirement for a specific system or class of systems, otherwise they should be a system or a mapping between functions and systems.

## Defining the systems

* Systems can be defined as entities that perform a function, e.g., a radar receiver, or computational elements that perform a function, e.g., identification algorithm
* Generally computational elements may reside on specific hardware so it is customary to focus on physical systems, but as more functionality is assigned to software that can reside on a multitude of hardware systems, it is important to acknowledge this explicitly in the knowledge base.
* A system exhibits processes that fulfill a function, systems that do not perform a function that is relevant to the needs of a project are not relevant.

## Mapping the Functions and Systems

All functions are

### Relations **Allowed** in the knowledge Base

*A* ***function*** *may be* ***satisfied*** *by other* ***functions***, e.g., to detect an airborne threat, the AAW officer could receive a warning from a warning receiver (if the aircraft is targeting the ship), detect it with an active radar, or receive information from another sensor through a datalink. This corresponds to a series of OR statements for satisfying a function, e.g., this function can be satisfied by this function OR this other function.

*A* ***function*** *may* ***require*** *a* ***series of functions*** *to be satisfied*, e.g., in order to successfully intercept an airborne threat, the ship must be able to find, fix, track, target, engage the threat and assess if the engagement was successful. Each one of those functions must be satisfied for the interception to be achieved. In this case the function is satisfied by a series of AND statements, e.g., in order to achieve this function, this function AND this other function have to be achieved.



Figure 2. Allowed RElationships between Functions.

*A* ***system*** *may* ***satisfy functions***, e.g., a radar may provide the function track airborne target. As it was the case with the functions being satisfied by other functions, this statement allows for an OR aggregation, e.g., this function can be satisfied by this system OR this other system.

*A* ***system*** *may* ***require*** *a* ***series of functions***, e.g., an SM-2 may require a function called <*Launch SM-2*> and <*Control SM-2*>. As it was the case with a function requiring a series of functions, these statements allow for an AND aggregation of the functional requirements, e.g., in order for this system to be operational, this function AND this other function must be achieved.



Figure 3. Allowed Relationships between systems and functions.

*A* ***system performing a function*** *may* ***require*** *a* ***series of functions***, e.g., a SAM with a semi-active seeker will require illumination in order to engage a target. The system may be able to perform other functions, but may be incapable of achieving a sub-set of possible functions if other functions are not achieved. This statement allows for an AND aggregation, e.g., in order for this system to perform the given function, this different function AND this other function must be achieved.

*A* ***system******performing a function*** *may be* ***incompatible*** *with* ***another system******performing a******function***, e.g., a radar interfering with a satellite communication system, as reported[Brown, 1987] in the case of HMS Sheffield. The radar was not able to search for airborne threats while the SATCOM system was in use, at the same time, the SATCOM system could not receive data while the radar was emitting.



Figure 4. allowed relationships between relationships.

### Examples of Relations **Not Allowed** in the Knowledge Base

***Functions*** *may* ***not require systems,*** e.g., shooting down an airborne target may not require an SM-2, nor may it require a missile, as this relationship would prescribe a system solution to a particular need. When users specify these relationships they are biasing the solution space. It is possible that the solution the user has in mind is in fact the most effective one, but the system should not allow that relationship to occur because conditions may change, or the user may not have a complete understanding of the possible means to achieve that function. Furthermore, if one was to face the need to state that one function requires more than one system, it is more valuable to decompose that function into the sub-functions the systems in question would perform towards the satisfaction of the former function. This is a better representation of reality because if both systems are required it means that they are inherently performing different functions, each of which is required to satisfy the first function.



Figure 5. functions may not require systems.

What if a function requires two systems? A function is not be allowed to require a system, which must be represented as a function requiring two functions, each of which can be satisfied by one system.

***Systems*** *may* ***not******require******another system***, e.g., an SM-2 may not require a MK 41 Vertical Launch System (VLS). If a user desires to specify a system requiring another, that system requirement is an indication that the requiring system (e.g., the SM-2 in the prior example) requires a function that is provided by the required system (e.g., the MK 41 VLS may provide “host and launch Standard Missile”). It may be true that at the time of the specification, only one system could provide this functionality, but that does not mandate that another system in the future may not be able to provide the required functionality. By having systems require functions, the framework is made more robust to future developments and can identify functions that may be of interest for future development in order to develop more robust architectures.



Figure 6. Systems may not require Systems.

***Functions*** *may* ***not*** *be* ***incompatible*** *with* ***another function***, e.g., <*detect airborne targets*> and <*communicate through satellite relays*> cannot be made incompatible because their incompatibility arises from the technologies used. During the Falklands War it may have been a true statement that those two functions were incompatible, but as technology progressed, said incompatibilities were removed by using newer systems that did not interfere with one another.

## Capturing the intuitive evaluation of a functional ACHIEVEMENT

Users may use a variety of mental constructs to define whether or not a function is satisfied, also referred to as *functional achievement*. It has been observed that oftentimes the users are not aware of how they rationalize, and may even be surprised to discover how they determine, if a set of functions achieve another function or not. For example, if a user is asked how he or she determines if a function is achieved based on the achievement of its dependent functions, a user may indicate that it is dependent on the average of said function. If then the user is shown a bar chart like those presented in Figure 2, he or she may realize that there are thresholds beyond which a function is deemed unachieved. This modifies the mathematical model used to capture the user's determination of functional satisfaction.



Figure 7. The analysis must capture the expert's intuition and aggregate it correctly.

It is important to gauge the user early to capture their perceptions, and when working in groups, to capture the entire group's perception, because it may not be consistent from one decision maker to another. For example, in order to conduct Anti-Air Warfare, the users agreed that a ship should perform <*provide targeting radar warning*>, <*maintain air surveillance in all conditions*>, <*neutralize anti-ship missiles*>, and <*intercept ballistic missiles*>. Some users may think that a ship that performs the first three functions at 100% and the third at 50% is satisfactory in achieving AAW, while others may deem that the partially fulfilled function is critical and that it must be achieved at 100%. Furthermore, users may disagree to what level AAW is achieved when the required functions are not fully achieved.

The most critical aspect of this stage of the process is a tradeoff between information required and model accuracy. In order to accurately reproduce each user's perceived level of functional accomplishment for all possible functional achievements, it would be necessary to request that information from the user. This is clearly an extreme case, and it may be possible to predict an unobserved functional achievement combination if all others are obtained. The probability that the mathematical model is accurate decreases as the number of observations decrease, i.e., the accuracy of our predictions is positively correlated with the information we have about the user's perspective. The tradeoff is the time and effort required to obtain said information. As the user is queried for longer and longer periods of time, it is likely that the user will grow impatient, frustrated, tired or unresponsive. It is critical that the user is engaged in the process while at the same time, the mathematical model accurately represents his or her functional achievement perceptions. This paper proposes an approach that has been observed to best represent a series of naval operators' perspectives with the minimum amount of information required.

The most efficient approach identified by the author asks the question: "What is the negative impact on <*Function X*> if <*Function Y*> is not achieved?" This may sound confusing at first but it allows the framework to accurately represent most functional achievement perceptions with a single value as it will be explained later in section XX. The scale of the impact can be left as a normalized numeric range, e.g., 0% to 100%, or it can be given qualitative values, e.g., Very High, High, Medium and Low, to which numerical values are later assigned. If using qualitative scales, it is very important that the meaning of each level is clearly understood.

Steps recommended by Chris Gross, "Enterprise Risk Management and the Total Cost of Risk," Presentation to Iowa Actuaries Club, November 12, 2009.

* Start by ranking the risks subjectively. If multiple people are involved in the assessment, get their rankings separately, and then concentrate on the differences.
* Where established models exist for quantification, use them.
* Consider history where it is applicable.
* To make it easier to deal with small numbers, think in terms of the probability of occurrence over the next ten years.
* Use reference to other items on the list.

## Integrating and operating on the Knowledge Base

Must capture the functional dependencies with sufficient fidelity and in a computer-readable manner so that data can be aggregated correctly and agree with experts' experience

\*\* functions may be required by other functions, by systems or by systems performing a given function

\*\* functions are satisfied by systems

Must organize the data to facilitate the user's navigation of the database and the analyses derived from it. Of particular importance is facilitating the expansion of the knowledge-base (i.e., addition of elements and relationships), and

systems may be required by other systems but always through a function. This is because when a system requires a special system, e.g., a Harpoon may require a harpoon launcher or a controller, but that does not mean that in the future a new system may be able to perform that function. For that reason, the framework does not allow for systems to require systems directly. For example, a Boeing RGM-84 Harpoon Anti-ship Surface may require RGM-84 Launching and RGM-84 Controlling, that can be provided by a Harpoon Launcher and a SWG-1A Harpoon weapon control systems.

# References

\*\* Look into relevant topic areas, e.g., knowledge-base, utility theory, strategic planning, risk modeling, subjective probability theory, predicate calculus

[1] Akerkar, R. and P. Sajja, "Knowledge-Based Systems," Jones & Bartlett Learning, 2010.

Apeland, S., Aven, T., and T. Nilsen, "Quantifying uncertainty under a predictive, epistemic approach to risk analysis,"

Brown, D., " The Royal Navy and the Falklands War," Naval Institute Press, 1987.

Budescu, D., Abbas, A., and L. Wu, "Does Probability Weighting Matter in Probability Elicitation?" Journal of Mathematical Psychology, vol. 55, pp. 320-327, 2011.

Dam, S. H., "KBAD – A Cost-Effective Way to Conduct Design and Analysis," Systems and Proposal Engineering Company, White Paper, 2009.

de Tenorio, C., "Methods for Collaborative Conceptual Design of Aircraft Power Architectures," Ph.D. Thesis, Georgia Institute of Technology, July 2010.

Fisher, G. W., "When Oracles Fail-A Comparison of Four Procedures for Aggregating Subjective Probability Forecasts," Organizational Behavior and Human Performance, vol. 28, pp. 96-110, 1981.

Huber, G. P., "Methods for Quantifying Subjective Probabilities and Multi-Attribute Utilities," Decision Sciences, vol. 5(3), pp. 430-458, 1974.

Kaplan, S. and B. J. Garrick, "On the Quantitative Definition of Risk," Risk Analysis, vol. 1(1), pp. 11-27, 1981.

Ludke, R. L., Stauss, F. F., and D. H. Gustafson, "Comparison of Five Methods for Estimating Subjective Probability Distributions," Organizational Behavior and Human Performance, vol. 19(1), pp. 162-179, 1977.

Raczynski, C. M., "A Methodology for Comprehensive Strategic Planning and Program Prioritization," Ph.D. Thesis, Georgia Institute of Technology, May 2008.

Rittel, H. W. J. and M. M. Webber, "Dilemmas in a General Theory of Planning," Policy Sciences, vol. 4, pp. 155-169, 1973.

1. These rigid decomposition structures impede direct mapping between non-adjacent levels, e.g., the ability to perform a function may directly impact the satisfaction of a capability, but if we adopt a rigid framework with activities as an intermediary abstraction for functions, those relations can only be captured through artificially generated activities which are oftentimes redundant and unintuitive. [↑](#footnote-ref-1)