

Homework 3

Students may work in groups of 3 or 4, but no more than one Aerospace student per group. Due date: October 4.

From Requirements to System Design Assessment

This homework will build upon homework 2. We will explore the pathway from high-level requirements to lower-level requirements, their allocation to components in the design solution, and their representation as mathematical constraints which can be evaluated as part of the system design assessment.

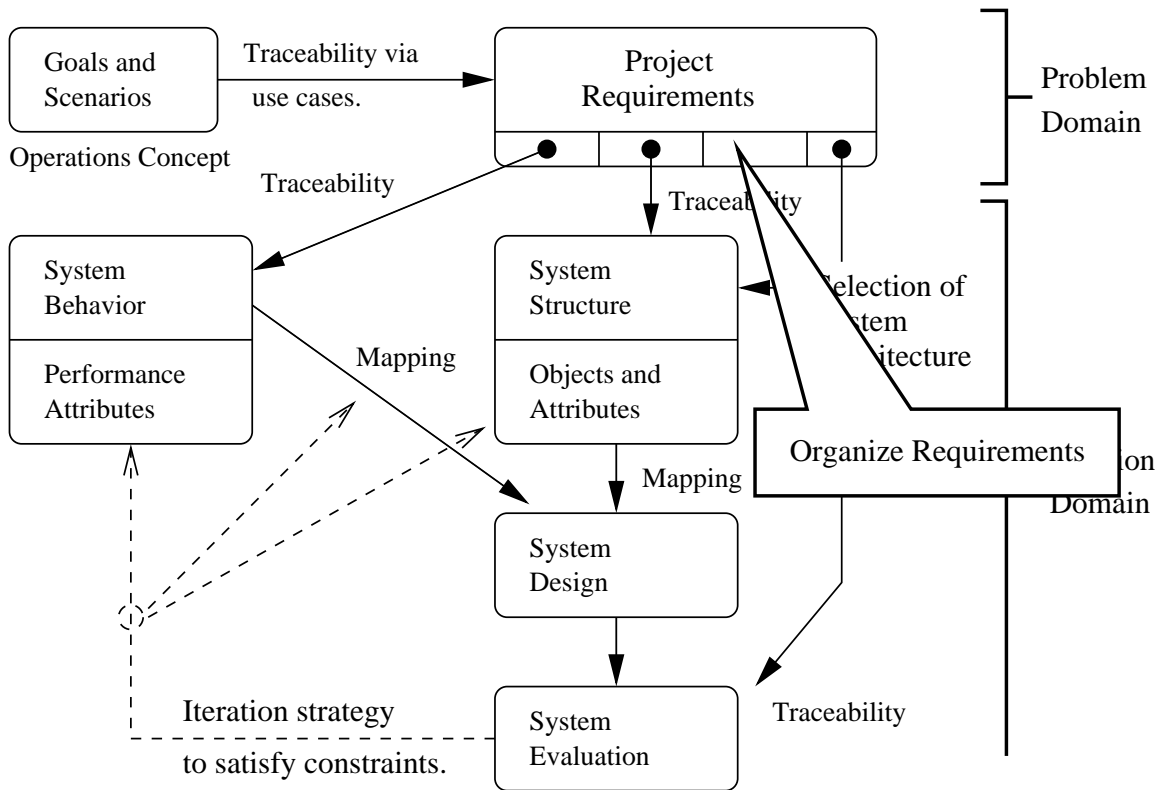


Figure 1. Requirements are organized according to the role they will play in evaluation of system behavior, the system architecture, and system testing.

Figure 1 shows the general pathway from requirements to constraints, the generation of models for sys-

sem behavior and system structure, the mapping (or allocation) of behavior to structure, and system design assessment.

Functional Allocation and Generation of Mathematical Constraints. Allocation is the process of breaking of a single attribute value into parts and assigning values to subordinate values. For example, a system budget will need to be allocated to the various subsystems. Functional allocation is the assignment of functions (requirements, specifications, behaviors, etc.) to system components (or groups of components).

We create the system design by mapping chunks of system behavior onto components in the system structure.

... before evaluation of the system design can occur, however, we need to identify the attributes of each component, and the fragments of functionality that each will perform.

Tables 1 and 2 illustrate the one way of systematically identifying objects and attributes associated with the system structure, and inequality constraints associated with the system performance.

Requirement	Structure	Behavior
	Objects, Attributes	Time, Performance, Sequence
Req. 1.0
Req. 2.0

Table 1. Identification of system attributes and functionality associated with each requirement.

Requirement	Specification	
	Attributes	Function/Performance
Req. 1.0		< 10 seconds
Req. 2.0	$1 \text{ kg} \leq \text{weight of container} \leq 2 \text{ kg}$	

Table 2. Generation of formal specifications from quantitative elements of requirements.

Three key points:

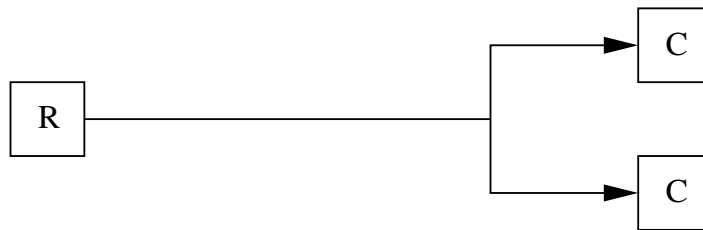
1. Specifications that are assigned to **attributes** will define the **state** of the system.
2. Specifications that are assigned to function/performance will be evaluated via performance analysis (engineering simulation).

3. Specifications can also be written as design constraints, an integral part of design optimization and trade-off analysis.

Generally speaking, acceptable levels of system performance will be described by inequality constraints (e.g., something needs to be greater than a certain amount). Other constraints will be equalities (e.g., the behavior of this system needs to satisfy physics described by $F = ma$).

Requirements and Constraints in Top-Down/Bottom-Up Design. Figure 2 shows two general scenarios: (1) generation of constraints from requirements, and (2) generation of requirements from external constraints.

Traditional Pathway: A requirement is transformed into constraints.



Synthesis of project requirements from external constraints.

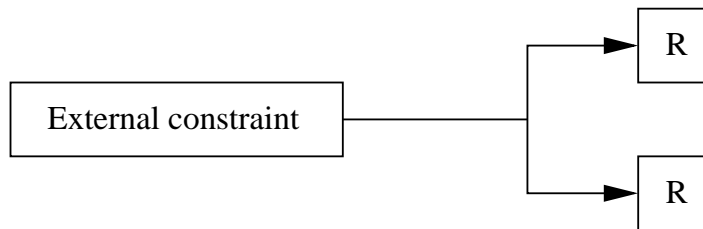


Figure 2. Generation of: (1) constraints from requirements, and (2) requirements from external constraints.

In the top-down development of a system, high-level requirements tend to flow down to lower-level requirements that can be written as mathematical constraints. Satisfaction of the mathematical constraints will imply that the lower-level requirements are satisfied which, in turn, will imply that the higher-level requirements are also satisfied.

But design is never just a top-down process. When a design solution involves use of previously design components, those components will impose constraints on the system solution. This is the lower part of Figure 2. Constraints will also be imposed on a design solution by the surrounding environment.

Group Homework Project

In this homework we will develop systems engineering representations (use cases, high- and low-level requirements, mathematical constraints) relevant to the operation of fighter jet taking off and landing on an aircraft carrier. See, for example, Figure 3.



Figure 3. Fighter jet landing on an aircraft carrier.

I suggest that you approach the problem by first getting a feeling for what the dynamics of the problem domain looks like, and then develop the systems engineering framework that connects use-cases to requirements to mathematical constraints to system design and assessment.

Things to do:

1. To get a feeling for the dynamics underlying the landing and take off tasks, develop a simple model for the behavior of an aircraft while it is taking off. I suggest that you keep the physics really simple, i.e.,

$$\text{Force} = \text{mass} \times \text{acceleration}. \quad (1)$$

Write a Matlab program to plot the change in velocity and distance as a function of time.

2. Develop a specification (see Figure 3.4 in the lecture notes) of fighter jet attributes relevant to the tasks of takeoff and landing.
3. Develop a specification (see Figure 3.4 in the lecture notes) for the attributes of a runway that will support the takeoff and landing of aircraft.

4. Develop use cases, textual scenarios, and activity diagrams for the landing and takeoff tasks.
5. Develop a set of high-level requirements for the landing and takeoff tasks.
6. Develop a set of lower-level for behavior of the aircraft during landing and takeoff.
7. Transform the lower-level requirements into set of constraints that can be evaluated against the attribute values in your runway specification. To the extent possible, identify typical values (or ranges of these values) that will occur in practice (e.g., aircraft mass and acceleration; braking and thrust forces) and evaluate the mathematical constraints.

A Few Hints. Use Google to find the attributes of runway design, the dimensions of an aircraft carrier, and so forth.