SAND2012-8138C

SAND**2012-8138C**

Description of the System Engineering Process to Develop [Sub]Component Specifications

Patrick O'Malley, 2627

Dept. 2627, Fuzing and instrumentation Technologies

9/10/2012

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Unclassified Unlimited Release

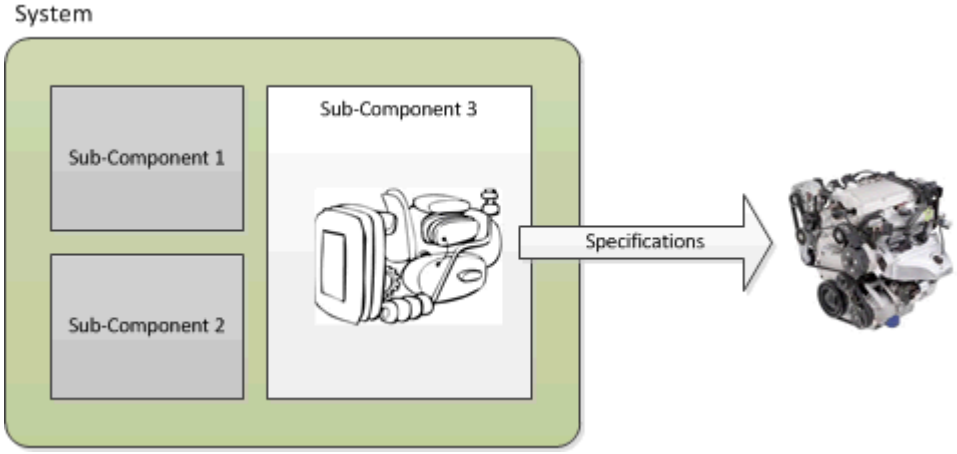
Sandia National Laboratories

Objective

The goal is to show how to develop sub-component specifications in a systems engineering process. The system modeling language (SysML) will be used for examples.

Role of Specifications

The role of specifications is to ensure that all requirements *necessary* and *sufficient* for the sub-component to meet its higher-level need are correctly flowed down.

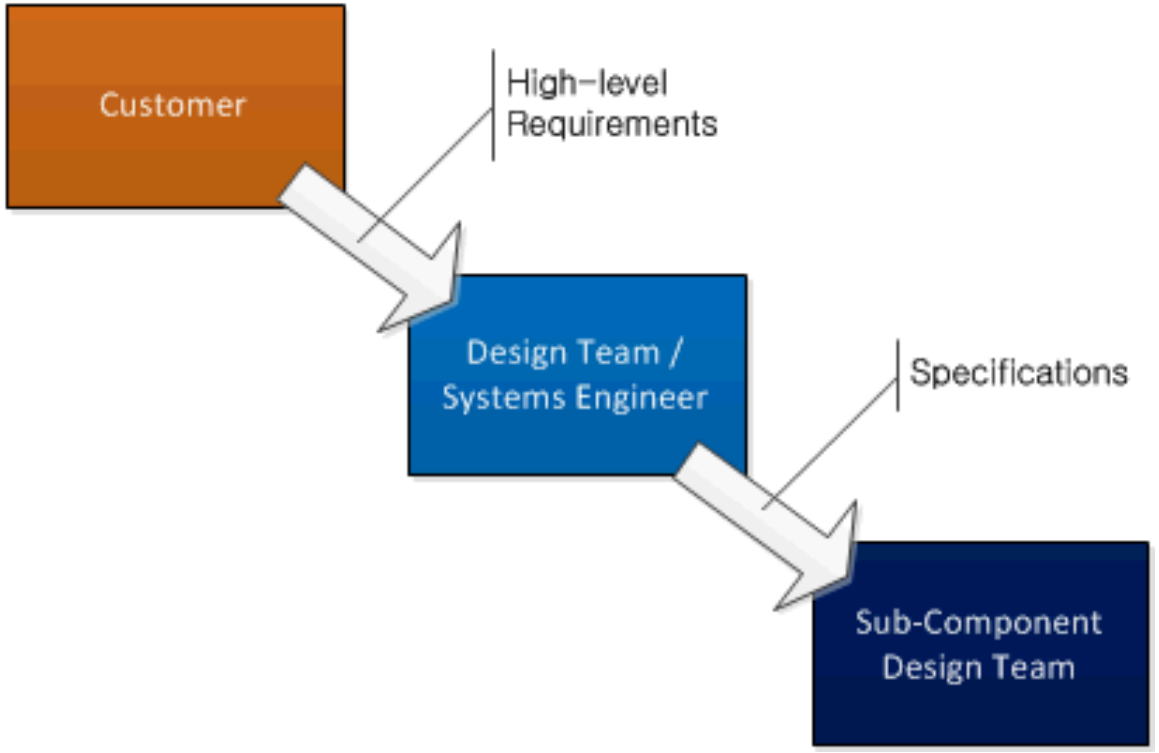


Usefulness and Risk of Specifications

- In a top-down, spiral development process, specifications are the "hand-off" to other development teams
- This process enables…
  - Specialists to work in their domain
  - Efficiency by allowing teams to work independently
  - Lower-risk integration later in development
- Correct specifications are absolutely required!
  - In this methodology, an incorrect specification can be costly!
  - Up-front work to ensure correct specifications is a good trade-off

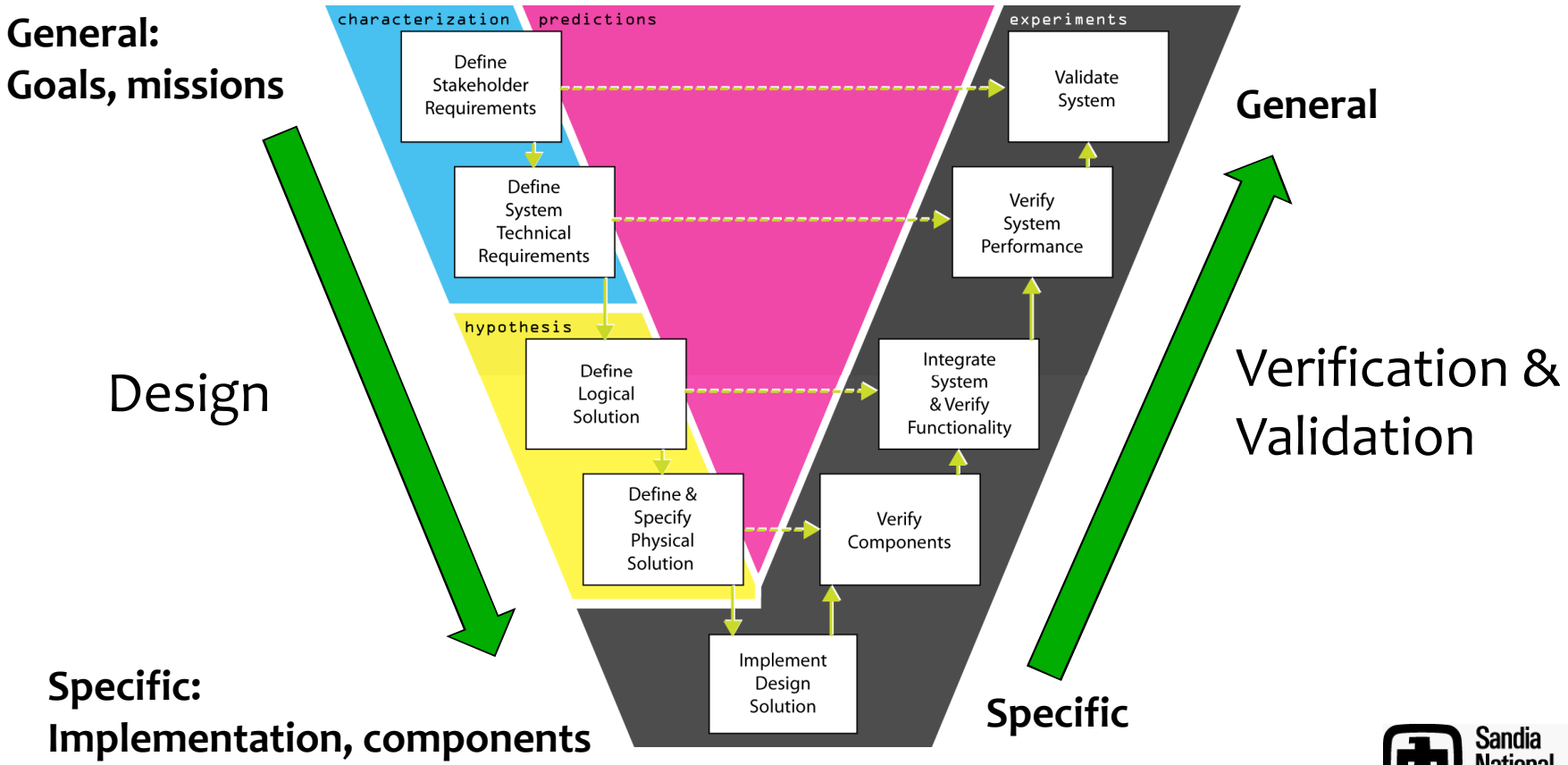
From Customer to Specifications

Specifications are derived from the customer's requirements. They may be the result of allocations, derivation or modeling.



Aside: The Systems Engineering Process

- The systems engineering (SE) process is generally shown as the "Vee Model"
- In this model, specifications are at the bottom of the Vee



Deriving Specifications

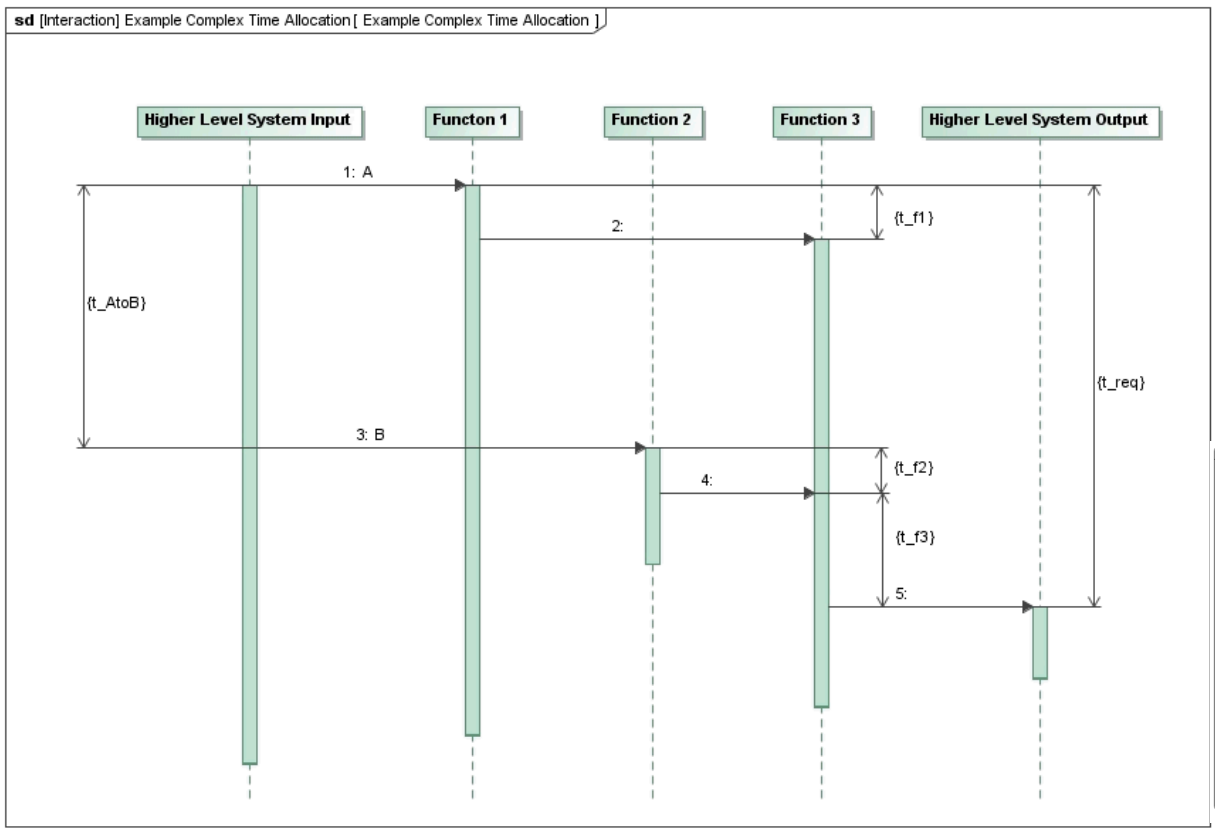
- There are generally 3 ways to develop specifications from requirements[1]
  - Decomposition
  - Derivation
  - Allocation
- We will only look at allocation, specifically, to exemplify the process

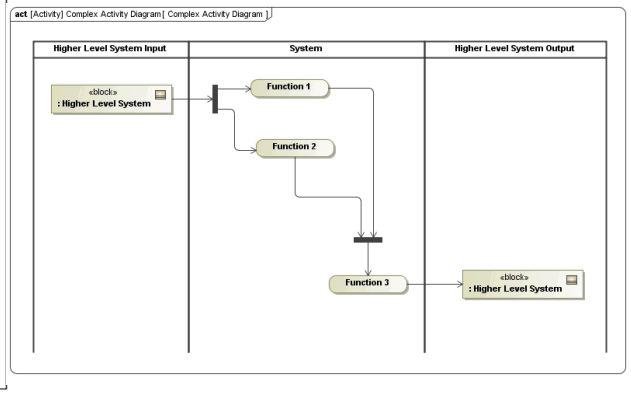
[1] – All of these ways, collectively, can be generally called “derivation”. Derivation is the formal method of determining whether and how a requirement will be flowed down to a lower-level.

Allocations

- One way to determine specifications is through allocation
  - Allocation distributes ("allocates") a limited resource to different items in a system
  - Examples: Time, power, mass, volume, etc.
- Models of allocations can be done in many ways
  - Spreadsheet (mass, volume, power)
  - SysML (see next slides)
  - MATLAB, physics tools (SPICE, Simplorer, etc)

A Time Allocation Model in SysML



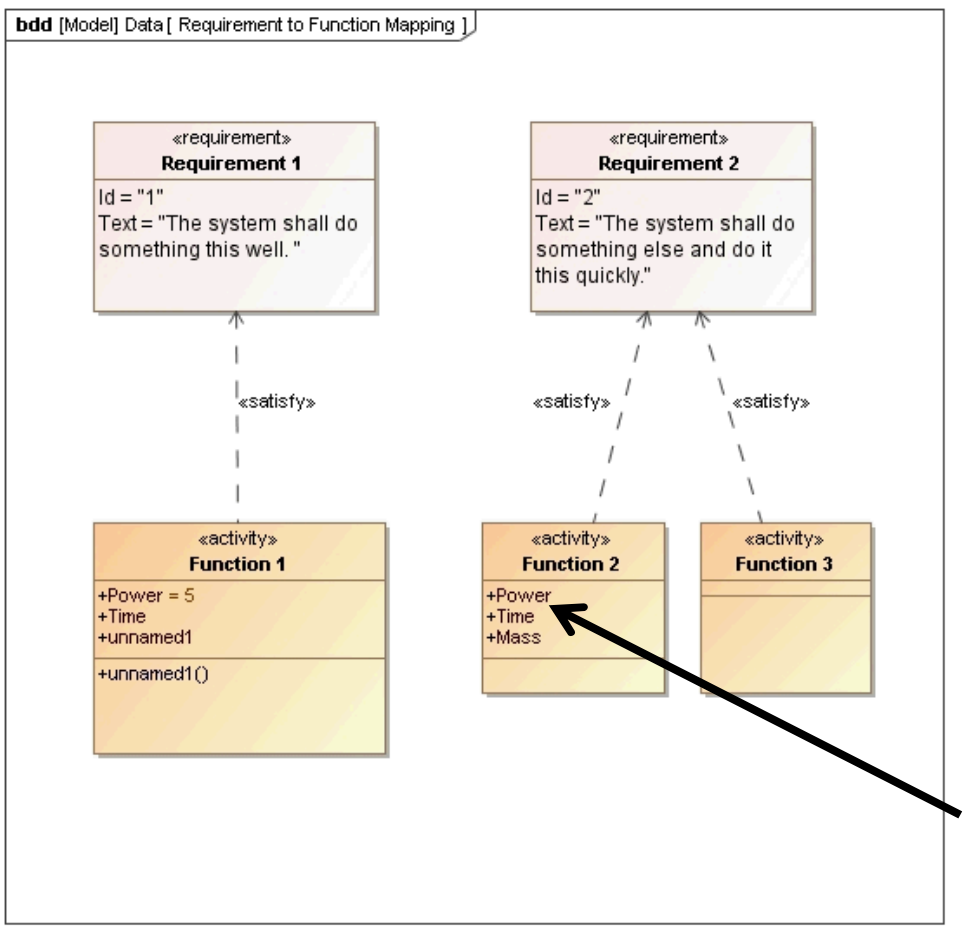


Two independent sets of allocations must be met in order to meet the requirement:

$$t_{AtoB} + t_{f2} + t_{f3} < t_{req}$$

$$t_{f1} < t_{AtoB}$$

Using SysML for Model Allocations



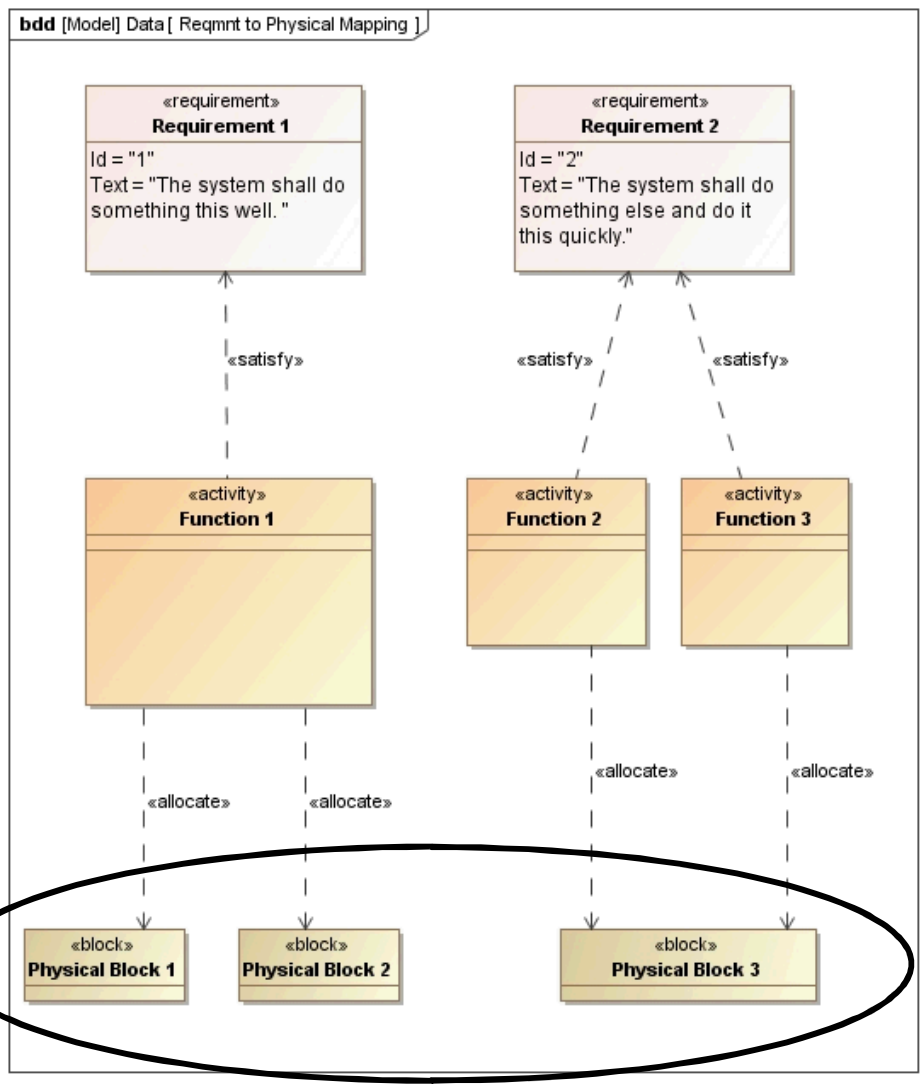
SysML maintains the ***traceability*** of the allocations to both the architecture and back to the requirements from where they were derived.

Allocations can be shown as properties/attributes in functional or physical blocks

Rational Allocation

- It usually isn't good practice to simply make "random" allocations
  - 2W = 1W + 1W? Simple but not useful!
- Make rational allocations: Have a reason for them
  - From data, analogy with other systems, analysis, etc.
  - For example: Allocating time between a switch (nanoseconds) and charging a capacitor (milliseconds) should be done rationally

Physical Blocks to Requirements Traceability



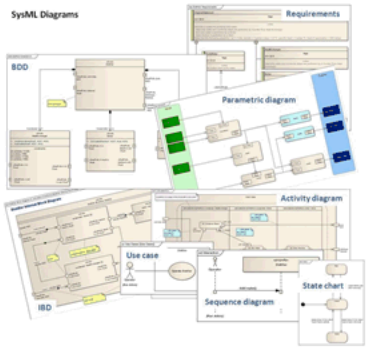
Through the “satisfy” and “allocate” relationships, we can show traceability from requirements to the physical blocks. (Allocations not shown in this view).

The physical level is likely where specifications would be handed off to sub-component teams.

Producing the Specs

Ultimately, the system model in SysML can be “transformed” into other [arbitrary] forms because all of the content and their ***relationships*** are available.

The only thing necessary is a report-generating script to transform the model into a different form.



SysML Models

+

Report Template

Component Specifications

Component Engineers