# Design generation and validation

How do we generate the design of a system? And how is it validated? The procedures for that are discussed in this chapter.

## 1 Designing

### 1.1 The Design Option Tree

Let's suppose we're designing some system. How do we do that? Well, the first step is to start generating options. In what ways can we design our system?

A very handy tool for this, is the **Design Option Tree** (DOT). In such a tree, we start with a certain task our system needs to be able to perform. We then look at the ways in which this task can be performed. Every time, we split things up even more. It is important to note that the DOT is an 'or-tree'. Only one of the available options can be selected. An example of a DOT for registering images without delay can be seen in figure 1.

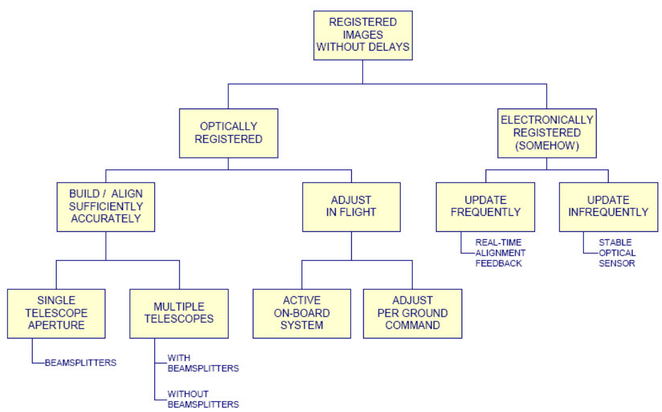


Figure 1: An example of a Design Option Tree for registering images without delay.

### 1.2 Choosing between designs

The DOT gives us a set of options. The next step is to choose the best option. This is a difficult process. There are therefore many ways to approach it.

A first thing we could do is eliminate concepts that are obviously not feasible. In this way, we can reduce the number of design options by quite a bit already.

The next step is to choose a **decision method**. We can distinguish two types of decision methods. In **ordinal (qualitative) methods**, the options are only ranked from best to worst, for every criterion. No individual score is given to them. So this is relatively easy. On the other hand, in **cardinal (quantitative) methods**, the options are given a score. This requires a bit more work.

### 1.3 Ordinal methods

Let's suppose we have a set of design options. Also suppose that, for every design criterion, we have a ranking of options, from best to worse. How do we now select the best option? There are, again, multiple methods for this.

In the **Majority Rule**, we compare two alternatives. If, for a certain design option, alternative A is better than alternative B, then alternative A gets a point. Otherwise alternative B gets a point. We continue to do this for every design option. Eventually, the alternative with the most points wins.

The **Copeland Rule** is similar. However, we now compare an alternative with all other alternatives. Eventually, the alternative with the most points wins.

The **Rank-Sum Rule** is a more simple method. We simply add the ranking of all the criterions. So if an alternative is second for criterion 1 and third for criterion 2, then its score is 5. The alternative with the lowest score now wins.

Finally, in the **Lexicographical Rule**, we first rank the criterions on their importance. We then only look at the most important criterion. The alternative that is best for this criterion wins. (In case of a tie, we look at the second most important criterion, and so on.)

### 1.4 Cardinal methods

To apply Cardinal methods, we first need to rate our alternatives. This must be done for every criterion. (So that will give us a lot of numbers.) Once this is done, how do we select the best alternative?

The most-often used cardinal method is the **Weighted Objectives Method** (WOM). To use this method, we must also give importance rankings for the criterions. Once that is done, we should find the score for the alternatives. This is done by multiplying the criterion importance by the rating score. The individual criterion scores are then added up. When doing this, a WOM table, as shown in figure 2, is often helpful. Eventually, the criterion with the best score wins.

| | *Criterion 1* | | *Criterion 2* | | ……… | | *Criterion n* | | *Maximum score* |
|---|---|---|---|---|---|---|---|---|---|
| Weighting factors | 10 | | 7 | | ……. | | 4 | | yyyy |
| | Raw score | Wtd score | Raw score | Wtd score | Raw score | Wtd score | Raw score | Wtd score | Solution score |
| *Alternative solution 1* | 5 | 50 | 8 | 56 | | | 6 | 24 | xxxx |
| *Alternative solution 2* | ….. | ….. | ….. | ….. | ….. | ….. | ….. | ….. | ….. |
| ………… | ….. | ….. | ….. | ….. | ….. | ….. | ….. | ….. | ….. |
| *Alternative solution m* | ….. | ….. | ….. | ….. | ….. | ….. | ….. | ….. | ….. |

Figure 2: The general form of a Weighted Objectives Method table.

## 2 Design verification

### 2.1 Verification methods

So now we've actually gotten a design for our system. This design should of course match the requirements. But we do have to be sure of this. The process of making sure the design meets the requirements is called **verification**.

There are several methods of verification (so-called **verification methods**). Examples are, in increasing level of complexity, . . .

- **Review of design:** Inspecting the design documentation, to make sure that the product satisfies the requirements.
- **Inspection of the product:** By inspecting the product, ensuring that it meets the requirements.
- **Analysis:** Using (mathematical) analysis techniques to check the properties of the product. (The FEM method is often used.)
- **Testing:** Subjecting the product to actual tests. For example, by using a prototype.

### 2.2 The verification level

Next to the verification method, also the **verification level** matters. Verification is usually performed on a level as low as possible. In **low level tests**, as few elements as possible are involved. This means that tests can be done early, and aren't very complicated. (For example, you often don't need to consider the satellite structure if you want to test its on-board computer.)

However, sometimes low-level tests are not possible. This is when elements start influencing each other. In this case **high level tests** may be necessary. (For example, when doing an automobile crash test, a nearly complete car is used. This is mostly because every part (having weight) influences the test.)

### 2.3 Test types

There are incredibly many ways to actually test a product. To list a few...

- **Functional tests:** Testing whether the product does the right things, and thus has the correct functionality.
- **Integration tests:** Testing whether all parts (mechanical/electrical/software) fit together in the correct way.
- **Structural tests:** Checking whether the structure meets the demands. By doing this, also the analysis tools (like FEM) and the workmanship are tested.
- **Aerodynamic tests:** Checking whether the aerodynamic properties of the product are sufficient. It also checks whether the mathematical model used during design was accurate. (Especially important for aircraft applications.)
- **Thermal tests:** Checking whether the thermal system works in a sufficient way. (Especially important for space applications.)

### 2.4 Aircraft design verification

When designing an aircraft, there are a lot of regulations it should comply with. Checking whether the aircraft design actually meets these regulations is called **compliance finding**.

The process of compliance finding can be quite elaborate. Several tests need to be done to show compliance with the airworthiness requirements. When this is done, the authorities can issue a **type certificate**. This certificate shows that the type of aircraft could be airworthy.

## 3 Design for production

### 3.1 What is design for production?

Let's suppose we're designing an aircraft. This aircraft eventually needs to enter production. With **production**, we mean all activities needed to turn a design into an actual product. During the design, we could make sure that production becomes relatively easy. By doing this, we **design for production**.

## 3.2 Recurring and non-recurring processes

A lot of processes are necessary to produce an aircraft. We can make a rather import distinction between recurring and non-recurring processes. **Non-recurring processes** are processes that are executed only once for every product type. (Examples include creating the design, programming machines and acquiring tools.) On the other hand, **recurring processes** are processes repeated for every product of a specific type. (Examples are manufacturing of parts, assembly of parts, and testing of the product.)

## 3.3 Designing the production process

To produce an aircraft, we need a production process. This production process of course also needs to be designed. The designing a production process consists of five phases. These are...

- **Concept generation:** Creating the general concept of the production process.
- **Feasibility check:** Checking whether the general concept is feasible/realistic.
- **Process definition:** Actually define the production process in detail.
- **Full scale development:** Set up the actual production process, according to the process definition.
- **Process validation:** Checking whether the production process actually meets the requirements and creates the right products.

When designing the production process, there are several things we need to pay attention to. The five most important points are...

- **Quality:** Is the quality sufficient? (Remember, we don't have to have the best quality for our product. A sufficient quality is good enough, and costs less.)
- **Time:** Is everything produced in time?
- **Money:** Is production as cheap as possible? (Is the amount of waste material minimized? Is the amount of elapse time minimized? Are transportation costs minimized? Etcetera.)
- **Volume:** Is the current production volume adequate to fulfill the demand? (Not too small, but also not too big?)
- **Law:** Does the production process obey safety, health, environmental and airworthiness regulations?

When all these points have received sufficient attention, then the production process is well designed.

## 3.4 Lean manufacturing

There is a philosophy in the world of production called **lean manufacturing**. According to this philosophy, production should be customer-focused, knowledge-driven, eliminating waste, creating value, dynamic and continuous.

The main focus lies on eliminating waste. According to lean manufacturing, **waste** is anything that uses resources, but does not add real value to the product or service. There are eight important kinds of waste, being

1. Overproduction
2. Waiting time
3. Work in progress (WIP) or inventory
4. Processing waste
5. Transportation
6. Movement or motion
7. Rework
8. Underutilizing people