# 1 Basic Concepts of Design

Design methodology emerged in the 1960s as an independent scientific discipline. This chapter looks to the theory of design methodology as a source of inspiration to understand the basic concept of design in the most general context. The objectives of the chapter are:

- To understand the basic characteristics of design processes;
- To understand the elements of designs;
- To understand the factors that affect design processes and outcomes.

The chapter is organised as follows. Section 1.1 gives a brief introduction to the notion of design. Section 1.2 examines the main characteristics of design activities in the creative processes of design. Section 1.3 considers designs as plans to produce man-made artefacts, and identifies the essential elements of all designs. Finally, section 1.4 presents Mayall's axioms of design to outline the basic relationships between various issues involved in every design.

## 1.1 INTRODUCTION

The word 'design' as defined in the *Longman Dictionary of Contemporary English* (1987) has the following meanings. As a noun, it means:

1. *A drawing or pattern showing how something is to be made;*
2. *The art of making such drawings or patterns;*
3. *The arrangement of parts in any man-made product, such as a machine or work of art, as this influences the product's practical usefulness;*
4. *A decorative pattern, esp. one that is not repeated;*
5. *A plan in the mind.*

The word design is also used as a verb with the following meanings.

6. *To make a drawing or pattern of (something that will be made or built); develop and draw the plans for;*
7. *To plan or develop for a certain purpose or use.*

As Christopher Jones pointed out in the book *Design Methods: Seeds of Human Futures* [ 1 ], design methodologists have been moving away from 'drawings and patterns' in the notion of design, although it is perhaps still a common action of designers of all kinds. The literature on design methods began to appear in the 1950s and 60s. Since then, design methodology has become an independent discipline of scientific study. The Design Research Society[1] publishes a quarterly journal *Design Studies* in London by Elsevier Science, which provides an insight into design issues affecting a wide range of fields of applications for design techniques. Researchers in the general theory of design have tried to answer two interrelated fundamental questions about design. The first question is:

*What are the essential characteristics of design*?

[1] The web address of the Design Research Society is: http://www.drs.org.uk/

This question relates to understanding when an activity is designing and when it is not.

The second question is:

*What processes are used by designers*?

It can be asked in a number of different ways with emphasis on various aspects of design processes. For example,

*Is one process better than another, constituting 'right' and 'wrong' ways to design?*

*Why are some processes favourable over others?*

*Do different processes lead to different qualities of results?*

The last few decades have seen a significant amount of research devoted to developing design theories with the ultimate aims at clarifying the human ability of designing in a scientific way, and at the same time, producing the practical knowledge about design methodology. Such knowledge is believed to be useful and essential to construct computer aided design systems.

As one of the most complex man-made artefacts, computer software is very difficult to design. There are many factors that affect designs and many stakeholders, i.e. people who participate in the design process, play various different roles in the design processes and influence the design of software. The questions that researchers in the area of design theory have been searching for answers to are also questions that computer scientists are looking for answers to in the context of software development. In fact, software design shares many characteristics with designs in other fields. As McPhee pointed out [2], much can be learned from the philosophical and methodological studies of design in general. This chapter is only a brief review of the basic concepts of design theory.

## 1.2 CHARACTERISTICS OF DESIGN ACTIVITIES

Let's first have a look at how design theory characterises design activities in the most general sense.

### 1.2.1 The input and start point of designs

Many design researchers believe in the aphorism 'necessity is the mother of invention'. It is considered as one of the basic characteristics of design that design can only be undertaken intentionally. Lawson [3] and Dasgupta [4] pointed out that a real or perceived need forms the basis for the definition of design projects. A need acts as the initial motivational force that provides the basis for starting design work. Willem [5, 6] explicitly expressed that the universal feature of design is simply the intentional devising of a plan or prototype for something new. The need or intention forms the first basic elements of all designs, i.e. the problem to be solved. In software design, the need or intention is usually explicitly or implicitly specified as users' requirements. Without users' requirements, there will be no software design.

### 1.2.2 The outcome and results of designs

Many designers believe that the output or product of a design is a symbolic representation of an artefact for implementation. For example, Booker (1964) regards design as simulating what we want to make (or do) before we make (or do) it as many times as may be necessary to feel confident in the final result. Dasgupta [4] expressed that design is essentially 'the formation of a prescription or model for an unfinished work in advance of its embodiment'. Design representation serves as the basis to conceptualise and compare various design decisions. This is very true in software design. Computer scientists and software engineers learned this lesson mostly from practical experiences that neglected design stage can only cause problems at later stages of implementation and so on.

However, the true output of design is more than just a plan or symbolic representation. MacLean *et al*. [7] pointed out that, the final output of a design also include what they call 'design space', which is a body of knowledge about the artefact, its environment, its intended use, and the decisions that went into creating the design. Designers must consider the representations of this kind of meta-knowledge about how they arrived at a particular design. Recently such meta-knowledge about software designs has been collected and studied systematically.

Two forms of systematic studies of such knowledge emerged in the literature of software design. One is about software architecture and the other is design patterns of object oriented systems. These types of knowledge form the main contents of this book.

### 1.2.3 Transformation of data

A basic feature of design that almost all design researchers accept implicitly or explicitly is the transformational nature of design. For example, Dasgupta [4] noted that need acts as a seed that design transforms into a form that is eventually used to guide the implementation of an artefact, plan or process. Simon [8] wrote that design is the restructuring of a current situation to achieve some preferred situation. Willem [5, 6] preferred to use the term 'development' to describe the transformation that occurs during design. Page [ 9 ] regarded design as an 'imaginative jump from present facts to future possibilities'.

### 1.2.4 Generation of new ideas

The requisite of the generation of new ideas during design processes is another commonly cited characteristic of design. Reswick defined design as a creative activity – it involves bringing in something new and useful that has not existed previously. However, creativity remains an elusive subject of design researches and still beyond science's firm grasp. The precise manner in which new ideas are generated still cannot be codified. Some researchers, such as Freeman [10], have postulated that idea generation is not entirely a haphazard activity. He believed that two styles of idea generation exist: abstraction and elaboration. Abstraction is used to make generalisations while elaboration attempts to develop into great detail the specifics of a design. In fact, these two styles of idea generation play a significant role in software design methodology.

### 1.2.5 Problem solving and decision making

Design methodologists tend to characterise design as a type of problem solving or decision making activity. For example, Asimow defined design as decision making, in the face of uncertainty, with high penalties for error. For many scientists and engineers, design invariably involves the application of some sort of logical analysis on the problem. Others, including Willem, believe that various design problem solutions are not necessarily connected through logic to their initial problem state. Design problems are often described as 'ill-structured' problems because of their complexity and the difficulties in determining their associated constraints and requirements. Freeman [10] preferred to use a decision-making analogy to view design problem solving. He characterised design as a series of decisions between various design alternatives. Each alternative is determined by the current state of abstractions, elaborations, operational statements and other known and unknown factors. Both design-as-problem-solving and design-as-decision-making views characterise design as goal directed activity and design process as navigation in a design configuration space.

### 1.2.6 Satisfying and discovering constraints

An initial need not only determines the problem to be solved, but also imposes the most basic constraints on the solution. In general, more constraints are often discovered during the design work itself. Many researchers agree that a major part of designing involves discovery and satisfaction of constraints on the eventual form of the design. Such constraints apply both to the designed artefacts and to the processes and participants involved in the design activity. For example, Mostow [11] regarded design as an activity with the goal of creating an artefact description that satisfies constraints derived from functional and performance specifications of the artefact, limitations of the medium and process by which the artefact is rendered or produced, and aesthetic criteria on the form of the artefact. For Alexander [12], design is 'finding the right physical components of a physical structure'. In the context of architectural design, Lawson [3] presents design problems as the assembling of constraints.

### 1.2.7 Evolution and optimisation in a solution space of diversity

As consequences of the complexity of design problems, diversity presents in almost all design solutions. Diversity often leads to uncertainty, because the knowledge that there are many other solutions to the same design problem causes designers to question the optimality of their initial solution. Thus, they test, evaluate and modify their design. Designers compensate for weaknesses exposed during testing and evaluation. They redesign as necessary until they are satisfied with their design. Therefore, design processes often demonstrate an evolution process. The evolution of a design is often closely related to the consolidation of the constraints and requirements applied in a particular design situation. Design requirements are often imprecise and incomplete. The consequences of a design decision often cannot be forecast with complete accuracy. Hence, design solutions evolve in tandem with known problem constraints and requirements. Eventually, a successful design process includes a convergence of requirements, constraints, and knowledge about the design and its effects on the implementation environment.

## 1.3 ESSENTIAL ELEMENTS OF DESIGNS

Having studied the characteristics of design processes, now let's examine the plan facet of design and identify the basic elements of designs. We will look to the designs made by a mastermind of designs, Thomas Edison, as our examples. Figure 1.1 is the one of Edison's USA patents of electric lights [13].

Notice that, the design presented in the document is not the kind of electric lights that we are using nowadays. It can be considered as one of Edison's designs of electric lights at an early stage of a long evolutionary design process. Although the design was not presented as a complete engineering design document due to the nature of the document, it still contained the basic components of all engineering designs. These components are discussed below.

### 1.3.1 Statement of design problem and objectives

The objective of a design is the problem to be solved and the goal to achieve. For example, in Edison's patent, it was stated that the design was about electric lights.

Design problems are often described as ill-structured [14] or 'wicked' [15] in contrast to well-structured or well defined problems such as chess-playing or crossword puzzles. Well defined problems have a clear goal, often one correct answer, and specific rules or known ways of proceeding that will generate an answer. The characteristics of ill-structured or wicked problems can be summarised as follows.

**(a) No definitive formulation of the problem.**

When a design problem is initially set, the goals are usually vague and many constraints and criteria are unknown. The context of the problem is often complex and poorly understood. In the example of Edison's design of electric lights, the goal of designing an electric light was obviously vague and the context was unclear. Understanding such a design problem is bound up with the ideas that we may have about solving it. This may lead to certain temporal formulations of the problem. For example, in the course of problem solving, Edison made a temporal formulation of the problem by assuming that an electric light was an apparatus that produces electric light '*by coil or strip of platinum or other metal that requires a high temperature to melt, the electric current rendering the same incandescent*'. This led to the problem that '*there is danger of the metal melting and destroying*

# UNITED STATES PATENT OFFICE.

THOMAS A. EDISON, OF MENLO PARK, NEW JERSEY.

## IMPROVEMENT IN ELECTRIC LIGHTS.

Specification forming part of Letters Patent No. **214,636**, dated April 22, 1879; application filed October 14, 1878.

CASE 156.

*To all whom it may concern:*

Be it known that I, THOMAS A. EDISON, of Menlo Park, in the State of New Jersey, have invented an Improvement in Electric Lights, of which the following is a specification.

Electric lights have been produced by a coil or strip of platina or other metal that requires a high temperature to melt, the electric current rendering the same incandescent. In all such lights there is danger of the metal melting and destroying the apparatus, and breaking the continuity of the circuit.

My improvement is made for regulating the electric current passing through such incandescent conductor automatically, and preventing its temperature rising to the melting-point, thus producing a reliable electric light by rendering conducting substances incandescent by passing an electric current through them.

In my apparatus the heat evolved or developed is made to regulate the electric current, so that the heat cannot become too intense, because the current is lessened by the effect of the heat when certain temperatures are reached, thereby preventing injury to the incandescent substance, by keeping the heat at all times below the melting-point of the incandescent substance.

Various devices for carrying my improvement into practice may be employed, and I have tested a large number. I however have shown in the drawings my improvement in a convenient form, and contemplate obtaining separate patents hereafter for other and various details of construction, and I state my present invention to relate, broadly, to the combination, with an electric light produced by incandescence, of an automatic thermal regulator for the electric current.

Figure 1 represents the electric-light apparatus in the form in which the thermal regulator acts by the heating effect of the current itself, and Fig. 2 illustrates the same invention when the radiated heat from the incandescent conductor operates the thermal regulator.

The incandescent metal is to be platinum, rhodium, iridium, titanium, or any other suitable conductor having a high fusing-point, and the same is used in the form of a wire or thin plate or leaf.

I have shown the platinum wire *a* as a double spiral, the two ends terminating upon the posts *b c*, to which the conductors *d e* are connected. The double spiral *a* is free to expand or contract by the heat, as both ends are below the spiral.

A circuit-closing lever, *f*, is introduced in the electric circuit, the points of contact being at *i*, and there is a platinum or similar wire, *k*, connected from the lever *f* to the head-piece or other support *l*.

The current from a magneto-electric machine, a battery, or any other source of electric energy, is connected to the binding-posts *n o*, and when contact at *i* is broken the current passes from *o* through lever *f*, wire *k*, support *l*, wire *e*, post *c*, platina coil *a*, post *b*, and wire *d*, or metallic connection, to binding-screw *n*. In this instance the wire *k*, being small, is acted upon by the electric current and heated, and by its expansion the lever *f* is allowed to close upon *i* and short-circuit the current.

The contact-point *i* is movable, and it is adjusted so that the shunt will not be closed until the temperature of the apparatus arrives at the desired height, and, by diverting a portion or the whole of the current, the temperature of the incandescent conductor is maintained in such a manner that there will be no risk of the apparatus being injured by excessive heat or the conductor fused.

If the wire *k* is small, so as to be heated by the electricity itself, it may be placed in any convenient position relatively to the light; but if such wire is heated by radiation from the electric light, then it should be adjacent to the incandescent material.

In all instances, the expansion or contraction of a suitable material under changes of temperature forms a thermostatic current-regulator that operates automatically, to prevent injury to the apparatus and to the body heated by the current.

In Fig. 2 the current does not pass through the wire *k*, and the short-circuiting lever is

2 **214,636**

operated by the radiated heat expanding the wire *k*. This in practice does not operate as rapidly as the device shown in Fig. 1.

The electric light may be surrounded by a glass tube or any other suitable device, such as two concentric glass tubes with the intervening space filled with alum-water or other bad conductor of heat, the object being to retain the heat of the incandescent metal and prevent loss by radiation, thus requiring less current to supply the loss by radiation.

I am aware that the electric current has been used to produce heat, and that such heat has been employed to vary the relative position of the light-giving electrodes and the length of the intervening arc. In my light there is no electric arc.

I claim as my invention—

1. In combination with an electric light having a continuous incandescent conductor, a thermostatic circuit-regulator, substantially as set forth.

2. In combination with an electric light, a thermostatically-operated shunt, substantially as set forth.

Signed by me this 5th day of October, A.D. 1878.

THOMAS A. EDISON.

Witnesses:
ALFRID SWANSON,
STOCKTON L. GRIFFIN.

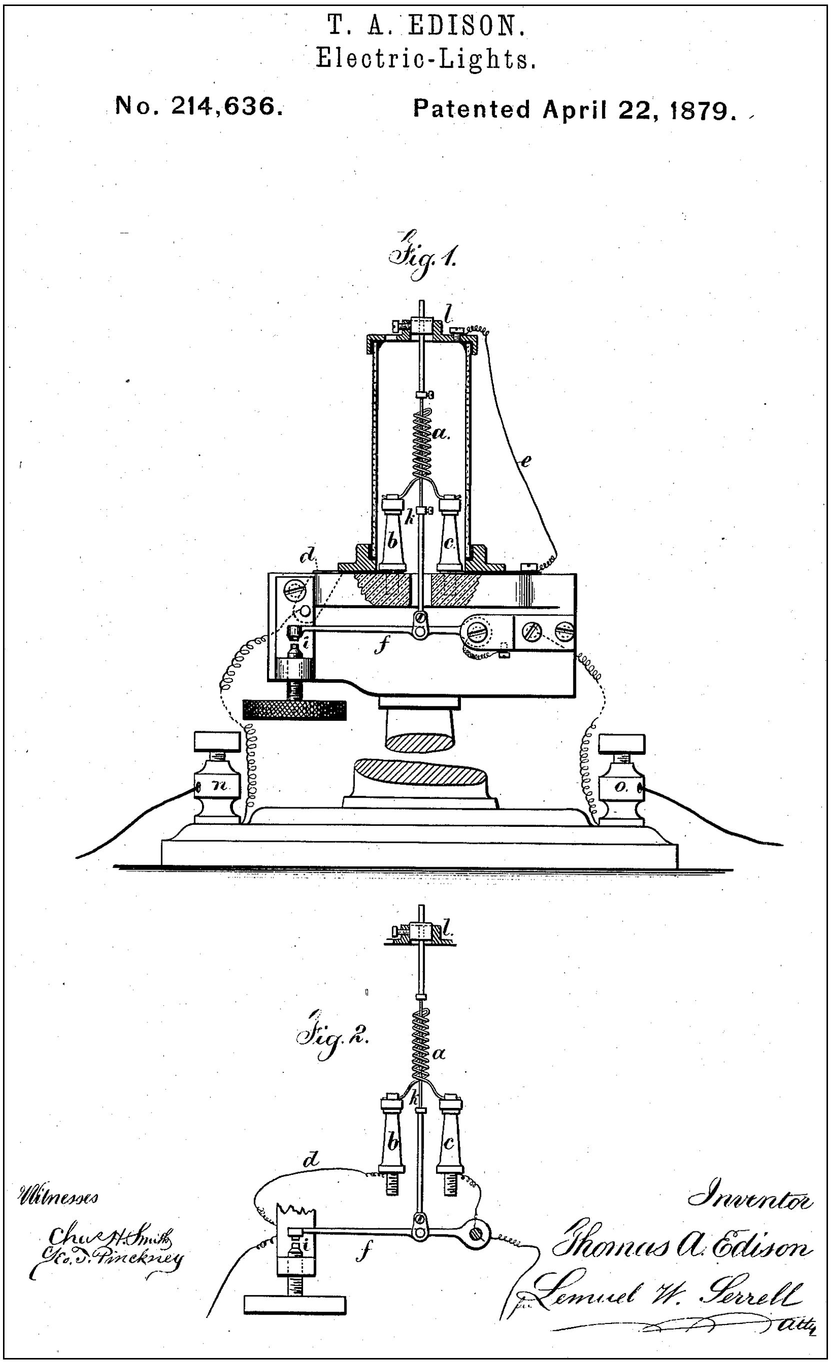


**Figure 1.1 One of Edison's designs of electric lights**

*the apparatus, and breaking the continuity of the circuit*'. However, as often occurs in all design processes, temporal formulations of design problems are unstable and can change as more information becomes available. Notice that, Edison's design of electric light presented in this patent is not that commonly used nowadays. Moreover, problem formulations are commonly inconsistent. Many conflicts and inconsistency must be resolved in the solution.

**(b) No definitive solution to the problem.**

Solutions to a design problem are often not true or false, but good or bad. Different solutions can be equally valid responses to the initial problem. There is often no objective criterion for the evaluation of a solution. However, solutions are assessed as good or bad, appropriate or inappropriate. For example, can we say Edison's design of electric lights presented in this patent is correct or incorrect? Instead, we might be able to say that this design is not as good as the design presented in Figure 1.3, which is now commonly in everyday use. Moreover, there is often no best solution. Essentially, this implies that there is a lack of any criteria that can be used as a 'stopping rule' to establish when the solution to a problem has been found such that any further work will not be able to improve upon it. No wonder why so many different types of electric lights have be designed, manufactured, marketed and used nowadays since Edison's invention.

**(c) No definitive way of solving the problem.**

There are no proven methods and rules that can definitely generate a solution to a design problem. Even a fairly precise problem statement gives no indication what a solution must be. Yet, solutions and problems influence each other in a 'wicked' way. A wicked problem can often be considered to be a symptom of another problem. Resolving a discrepancy or inconsistency in a design may pose another problem in its turn. The formulation of a problem often depends on the way of solving it. Many assumptions about the problem and specific areas of uncertainty can be exposed only by proposing solution concepts. Many constraints and criteria emerge as a result of evaluating solution proposals. Sub-solutions of the design problem can be found to be inter-connected with each other in ways that form a pernicious circular structure to the problem. For example, a sub-solution that resolves a particular sub-problem may create irreconcilable conflict with other sub-problems.

The ill-structured and wicked problem of design is the main reason why design is difficult. In Chapter 3, we will have a closer look at the particular reasons why software design is difficult.

### 1.3.2 Constraints

The objectives of a design often have to be achieved within certain constraints. These constraints define the solution space of the design problem. For the electric light problem, the most significant constraints are that the power must be electricity and it must be able to provide light continuously for a practically acceptable length of time. There are also other physical constraints, for example, the fusing-point of the metal used for the coil. Some of the constraints are explicitly mentioned in the patent document, and many others left as implicit. Some constraints are discovered and/or introduced by the designer during the course of design. In general, constraints can be classified along the following three dimensions according to Lawson [3].

**(a) The generator of the constraint.**

Constraints can be generated by the eventual users of the artefact, by the designers themselves, by legislators (e.g. safety related constraints), and by the design clients (i.e. the people who have commissioned or sponsored the design and who may or may not be eventual users of the artefact).

**(b) The domain of the constraint.**

According to Lawson, constraints fall into two domains, external and internal. External constraints are imposed by the factors not under the designer's control, while internal constraints give the designer at least some ability to control them.

**(c) The function of the constraint.**

Lawson's third dimension, constraint function, relates to the rationales behind the imposition of each of the constraints. Constraints can exist for reasons relating to symbolism and social norms, formal intentions of the designer, practical implications brought by the implementation technologies, and 'radical' reasons that deal with the primary purpose of the artefact.

### 1.3.3 Description of product

The main result of the design activity must be presented in the form of a description of the designed product. In this example, the product is described by the diagrams and in the text as well. The product of Edison's patent is an *automatic thermal regulator* for the electric current that prevents the melting and destroying of the apparatus. In addition to the description of the structure of the apparatus, there are also descriptions of the materials used to make the various parts of the apparatus and how the apparatus works.

Notice that Edison used two diagrams to indicate two different states of the apparatus. In more complicated products, engineers often find that it is too complicated to use only one diagram to describe the structure and states of a system. Therefore, a number of different diagrams or drawings may be used to give details of various parts of the system. Sometimes, different notations may be used to specify different aspects of the system. This is a common practice in all engineering designs. For example, Leonardo Da Vinci, a great artist and inventor in the 15th to 16th century, often produced a number of drawings for one design of machinery. As shown in Figure 1.2a, Leonardo presented his design of a crossbow by a drawing that shows its overall structure and several drawings of the critical parts of the crossbow. Similarly, Figure 1.2b is a design of a wing as a part of Leonardo's flying machine dated to between 1486 and 1490. On the lower right corner of the drawing shows how the wings are to be connected and operated. The main part of the drawing gives more details of the design of the wing itself. In software engineering, software designs are also represented in a number of interrelated diagrams and with text explanations. We will see some examples later in the book.

Representing a complicated design in a number of different levels of abstraction and with different details is, in fact, an important principle of the representation of engineering designs. It is recognised in software engineering as *structural* or *hierarchical representation*.

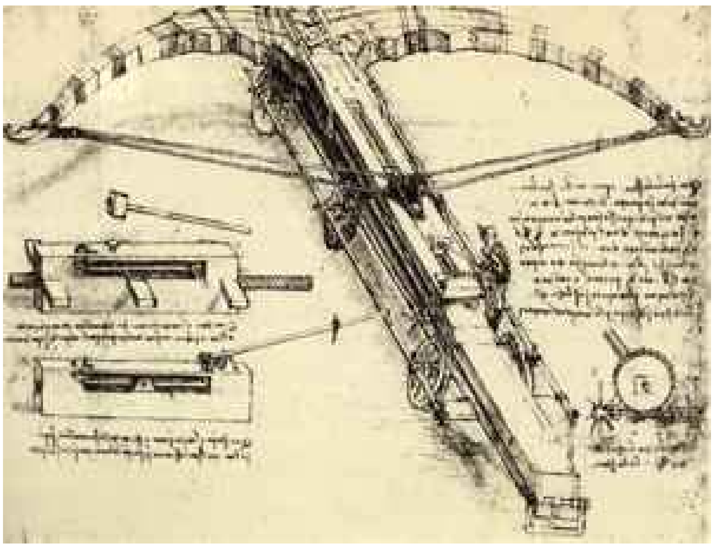

**Figure 1.2a Leonardo Da Vinci's design of crossbow**

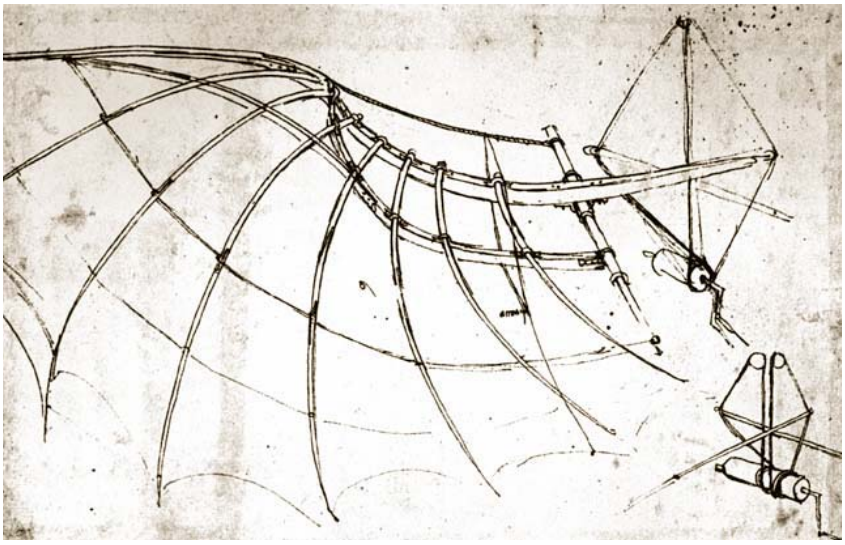

**Figure 1.2b Leonardo Da Vinci's design of a flying machine**

Using a number of diagrams to represent a complicated design is also related to another important principle of design, which is called *multiple views* in software engineering. Because the design and production of a system may involve many people of different backgrounds who play different roles in the course, the design should be presented to each person only with the information that they are concerned with and in the way that is suitable for their background and is convenient for them to work on. Therefore, different notations should be used to describe different aspects of a design. Such descriptions of one system in various notations form a set of models of the system. Each model presents a view of the system. These views must be consistent with each other.

## 1.3.4 Rationale

Engineering designs must be based on scientific principles and technical information. The rationale underlying a design justifies the design by applying such scientific and technological knowledge to show how the design problem is actually solved, or at least why we can predict that the design problem will be solved by the design. Edison gave a very clear expression of the rationale underlying this design. In particular, the automatic thermal regulator is deployed to keep the heat 'at all times below the melting-point of the incandescent substance'.

Since the document is only a partial result of an early stage of Edison's design of electric lights, there are a few important elements missing from the document.

These elements were presented in Edison's other patent documents. For example, the following elements can be found.

### 1.3.5 Plan of production

As an engineering design, we not only require to know whether a design can solve a problem, but also we must know the design is practical. One of the most fundamental questions related to the practicality of a design is how to bring about the design. In this case, it is the problem of the manufacture of light bulbs. Edison addressed this problem and obtained a number of his patents, which include the manufacture of Carbon filament, etc. Figure 1.3 shows the diagram in a patent by Edison in 1889 about how to manufacture electric lights[2] [16]. Sometimes, how to bring about a design is a common knowledge when the product is clearly described. However, engineers often found it is one of the most challenging problems of design.

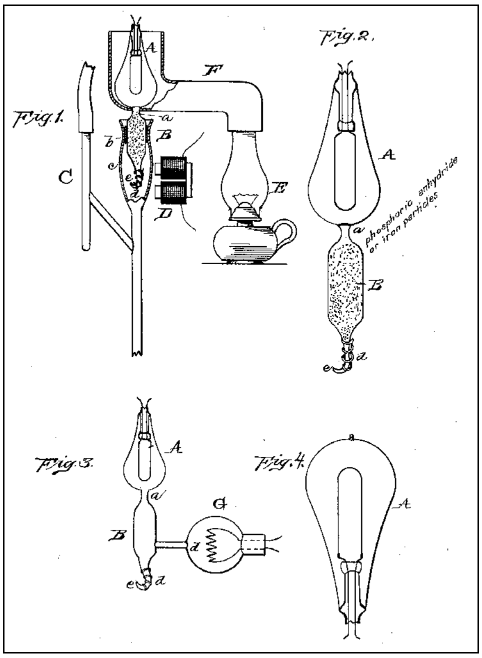


**Figure 1.3 Manufacture of electric lights as patented by Edison**

---

[2] Notice that the electric light is not the same as that in Figure 1.1.

### 1.3.6 Description of usage

There are always certain conditions under which a product can be used safely and effectively so that the objectives of the design can be achieved. How a product is to be used is sometimes not a trivial problem. Instead, it can be as hard as, sometimes even more difficult, to solve than the problem of designing the product itself. It may appear in the form of designing a system of which the product is only a part. Figure 1.4 shows the system of electric light as designed by Edison in one of his patents [17].

These elements of designs that we found in Edison's designs of electric lights appear in all designs including the designs of software systems although they may appear in different forms.

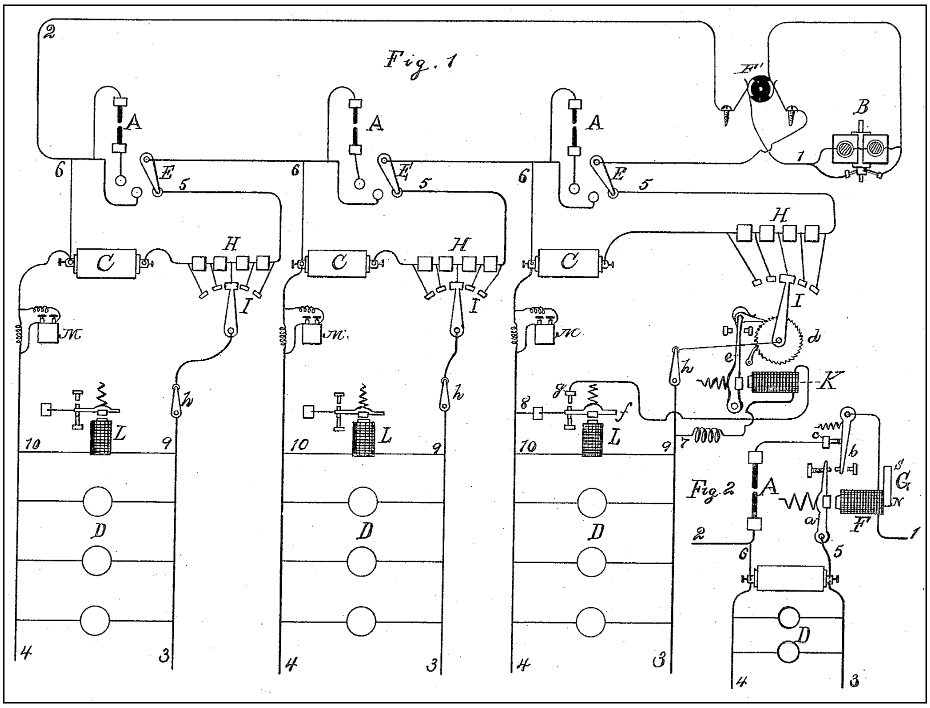


**Figure 1.4 System of electric lights as patented by Edison**

## 1.4 THE FACTORS THAT AFFECT DESIGNS

In search of the factors that affect design processes and outcomes and the analysis of their interrelationships, Mayall proposed a set of axioms as the principles of designs [18]. Although different types of product may have different features and their design processes may involve different domain-specific activities, Mayall's axioms present a set of general laws of design that characterise the nature of design. The following looks at these general laws and explains them in the context of software design.

(1) *The Principle of Totality*: All design requirements are always interrelated and must always be treated as such throughout a design task.

This axiom states that design requirements are the most important factor that affects the whole design process. Moreover, the elements of design requirements are interrelated and should be treated as a whole during design.

This axiom is very much true in software design. The relationships between software requirements vary significantly. Sometimes, users' requirements conflict one to another. It has been widely recognised that conflict resolution and requirements prioritisation must be considered as an essential part of requirements analysis in software development. The axiom also tells us that design decisions must be made on the basis of a deep understanding of the interrelationships between the requirements.

(2) *The Principle of Time*: The features and characteristics of all products change as time passes.

This axiom states that time is an important factor that affects designs. This nature of design has been well demonstrated in software designs. For example, 20 years ago, a program that interacts with the user through command line input/output may be considered as user-friendly, if it gives prompts for user's input and displays outputs in the form of dialogues. Now, with the wide spread of graphic user interfaces, such computer-human interaction can hardly be considered as user-friendly. A program that uses 10 mega-bites of memory space would be considered as requiring too much resource and impractical 20 years ago, but nowadays a student dissertation project would normally take more than that size of memory space. Therefore, a design cannot be evaluated without taking into consideration of the times when the design was made and when the evaluation was made.

(3) *The Principle of Value*: The characteristics of all products have different relative values depending upon the different circumstances and times in which they may be used.

This axiom states that the relative value of a product is an important issue that must be taken into consideration in the design. It further states that the value depends on the circumstances and time when the product is used. It is not fixed. A good software design made ten years ago is probably out of date and less satisfactory now because users' requirements have changed and the hardware and software platforms to implement and execute the software have also changed. Features that were considered desirable years ago may have become less important, unnecessary even harmful, while new features become the most important. Different users may value the same product differently. Even the same user may value the same product differently upon different circumstances such as in different use purposes. Designers should be aware of the user types and their purposes of use and to apply this knowledge in the design of the software. Adaptability for a wide range of user types should be considered.

(4) *The Principle of Resources*: The design, manufacture and life of all products and systems depend upon the materials, tools and skills upon which we can call.

This axiom states that the resources that are available for the design, manufacture and operation of the product are important factors that design must take into consideration. Such resources include tools, skills and materials. Software development, operation and maintenance demand a large amount of resources, which include the following types: (a) *development tools*, such as programming languages and compilers, CASE tools including configuration management and testing tools, etc. (b) *run time support systems*, including software and hardware platforms, other system software such as database management systems, network protocols, etc. (c) *human resource*, which is perhaps the most important among all resources of software development, (d) *application domain-specific tools and equipment*, especially in the development of embedded systems.

(5) *The Principle of Synthesis*: All features of a product must combine to satisfy all the characteristics we expect it to possess with an acceptable relative importance for as long as we wish, bearing in mind the resources available to make and use it.

This axiom states that features, or functions, of a product constitute a factor that affects design as the objective of design. They combine together to satisfy the requirements. In software engineering, a good design must take all functional and non-functional requirements and their relationships into account. In software design practice, it is almost inevitable to make trade-offs between desirable features and functions. Such trade-offs must be based on a deep understanding of the users' requirements as a whole and bear in mind the constraints on the resource available. In the next chapter, we will examine in detail the quality attributes of software systems and software development processes. We will see that certain quality attributes may affect other quality attributes in a negative way.

(6) *The Principle of Iteration*: Design requires processes of evaluation that begin with the first intentions to explore the need for a product or system. These processes continue throughout all subsequent design and development stages to the user himself, whose reactions will often cause the iterative process to continue with a new product or system.

This principle states the importance of design process. It emphasises the importance of evaluation of design and users' feedback, which is no exception to software design. In fact, being perhaps the most complicated artefacts that human beings have ever designed and built, software systems are difficult to design, evaluate and validate. Software development practices indicate that errors made during design stage are difficult even impossible to rectify at later stages during implementation and maintenance. As a consequence of evaluation and validation, also for many other reasons, designs have to be changed to correct errors and to improve quality. Such changes often need to go through many loops of evaluation-modification iteration to reach a satisfactory status.

(7) *The Principle of Change*: Design is a process of change, an activity undertaken not only to meet changing circumstance, but also to bring about changes to those circumstances by the nature of the product it creates.

This axiom states that the consequence of design is to bring about changes. The application of computers has significantly changed the way we work and live, and brought the human civilisation to the so-called information age. Information systems can be classified into three types according to the changes that they bring about. An information system is *automational*, if it automates certain activities that were originally carried out by human beings. It is called *informative*, if it generates information that was not available before. It is called *transformational*, if it changes the way that business or certain tasks were carried out within the organisation. The design of a software system must take into consideration about how the software is to be used. We need to consider not only how the design fits into the way that we are working and living, but also how it changes the way that we will work and live as the consequence of using the system. Software designers must be aware of their responsibility. On the other hand, the changed world raises new requirements for software designers. This requires new designs of software systems as well as modifications on existing systems. An important quality issue of software designs is modifiability, which means whether it is easy to make modifications.

(8) *The Principle of Relationships*: Design work cannot be undertaken effectively without established working relationships with all those activities concerned with the conception, manufacture and marketing of products and, importantly, with the prospective user, together with all the services he may call upon to assist his judgement and protect his interests.

Of course, the people involved in the design process are a very important factor that affects design. What's important is not only the individuals involved in design, but also the working relationships between them. Software design methodologies have also identified various stakeholders who may contribute to software design in various ways. Typically, in addition to software designers, other stakeholders involved in software design include:

- *Customers:* who purchase the software
- *Users:* who use the software and are responsible for executing the software
- *System administrator:* who is responsible for managing the data repositories used by the systems
- *Project managers:* who manage the software development process
- *Developers*: who are responsible for developing and/or modifying the runtime functions of the system. Developers can be further divided into a number of sub-types of stakeholders, such as requirements analysts, designers, programmers, testers, etc.
- *Requirements analysts:* who analyse, specify and approve the requirements of the system which is the basis of design
- *Designers*: who design the software system at architectural level or at detail level. In particular, designers of software architectures are often call software architects
- *Programmers:* who implement the software according to the design
- *Testers:* who review and/or inspect various development documents and test the software product after implementation
- *Auditors:* who audit the development activity

- *Support technicians:* who provide technical supports, such as providing and maintaining software development and design tools.

(9) *The Principle of Competence*: Design competence is the ability to create a synthesis of features that achieves all desired characteristics in terms of their required life and relative value, using available effective information about this synthesis to those who will turn it into products or systems.

This axiom states that an important factor that affects design is the competence of the designer. In the context of software design, this principle reads that design competence is the ability to design a software system that satisfies all requirements including functional and non-functional requirements and to effectively document the design so that the software can be implemented according to the design.

(10) *The Principle of Service*: Design must satisfy everybody, and not just those for whom its products are directly intended.

This axiom states that service must also be considered in design. In the context of software design, a good design should not only satisfy the requirements of the intended usage of the system, but should also satisfy other stakeholders. For example, it must be easy to maintain, easy to reuse, easy to transport to other operation environments and to be inter-operable to other software systems, etc.

## SUMMARY

There are two facets of the concept of design. Firstly, a design is a *plan* to bring about a man-made product. Such a plan must achieve a prescribed goal and satisfy certain constraints. Secondly, it is a *process* of the creative development of such a plan. During this process, the designer must use related scientific principles, technical information and imagination to discover constraints and to solve the design problem. Therefore, we can define engineering design as follows.

*Engineering design* is the use of scientific principles and technical information in the creative development of a plan to bring about a man-made product to achieve a prescribed goal with certain specified constraints. The consequence of the implementation of the design will bring changes to the environment, while the environment of the designer influences the design itself. Software design is a branch of engineering design where the product to bring about is software.

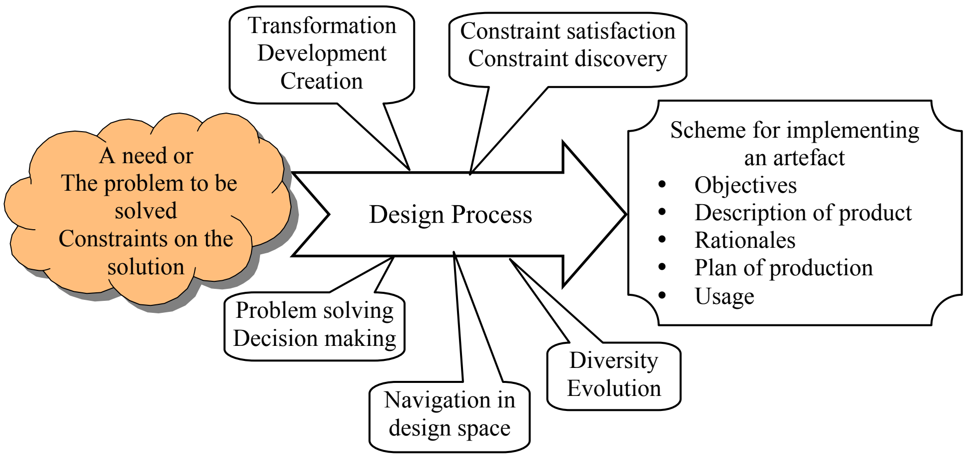


**Figure 1.5 Basic concepts of design**

As depicted in Figure 1.5, design activities have the following characteristics. Design starts with a need and requires intention. It results in a scheme for implementing an artefact. It involves transformations and the generation of new ideas is fundamental to all designs. Design is goal directed problem solving and decision making. It must satisfy the constraints and the requirements. The design process is also a constraint discovery process. Design is to achieve optimality in a solution space of diversity; hence, it is often an evolutionary process.

An engineering design should contain at least five basic elements: (a) the objectives of the design, (b) a description of the designed product, (c) the rationale of the design, (d) a plan of the production, and finally, (e) the designated usage of the product.

There are a number of factors that effect design processes and their outcomes. These factors include (a) the requirements to be satisfied by the design, (b) the time design was made and evaluated, (c) the value of the product, (d) the resource available to the design, manufacture and use of the product, (e) the features, or functions, of the product, (f) the process of design, (g) the consequence of design, i.e. the change to be brought about, (h) the people involved in the design process and their working relationships, (i) the competence of the designer, (j) the service of the designed products. These factors interrelate with each other as stated in the axioms of design proposed by Mayall.

## FURTHER READING

There are a number of books on design methodology, most of which illustrate the theory with examples of designs of architectures, physical articles and machinery. For example, Jones' book *Design Methods: Seeds of Human Future* [1] is one of the best-known. The book *Developments of Design Methodology* [19] edited by Cross and published in 1984 collected a number of papers on design methodology that represent the milestones in the development of design methodology into an independent scientific discipline. Many of the authors of the papers collected in the book have been referred to in this chapter. The current state of art in the research on design methodology is well represented by the publications in the Design Research Society's quarterly journal *Design Studies*, which is published by Elsevier Science and Technology in London. More details about the journal including table of contents in each issue can be found at the website of the journal at the URL: http://www.drs.org.uk/. The research on software design has evolved rapidly over the past few decades. Most of the earliest work on the subject is collected in the *Tutorial on Software Design Techniques* [20] edited by Freeman and Wasserman and published by IEEE in 1980. Since then, significant progress has been made.

# EXERCISES

**(1–1)** Discuss whether an activity can be considered as a design activity if it has no fixed goals or objectives to achieve.

**(1–2)** Discuss why the output of design is a symbolic representation of an artefact for implementation rather than a real product.

**(1–3)** Discuss whether code in a high level programming language is an appropriate form to represent the results of software design.

**(1–4)** Regarding design as a process of transformation, discuss what is transformed in software design.

**(1–5)** Discuss why design is a creative activity.

**(1–6)** Discuss why design problems are often said to be ill-structured. Give an example of an ill-structured design problem.

**(1–7)** Discuss why decision making in design process often faces uncertainty.

**(1–8)** Consider the following constraints imposed on the design of a software system. Apply Lawson's theory of constraints to analyse these constraints and find out the three dimensions of each constraint.

(a) The software is to be executed on a PDA (Personal Data Assistance);

(b) The output of the software must be displayed on the screen of the size 3″×2″;

(c) The software should be easy to operate through a small keypad that has character buttons and some functional buttons;

(d) The software can only be used if the user subscribes to a specific online service of the client's company;

(e) The software will be implemented using the C language.

**(1–9)** Discuss why testing and evaluation of designs are important.

**(1–10)** Assume that you want to make a bookshelf to put in your room.

(i) Make a design of the bookshelf, and record the activities that you perform during the design process.

(ii) Answer the following questions.

(a) What are the objectives of the design? How do they interrelate to each other?

(b) What are the constraints on the design at the beginning? Do you discover any constraints during the course of design?

(c) What are the stakeholders involved in the design? (*Hint: a person involved in a design may play different roles. In such cases, the person should be considered as different stakeholders*.)

(d) What is the designed product? How do you describe it?

(e) What is your plan of making the bookshelf? How do you describe the plan?

(f) What are the design decisions you made during the design and what are your rationales of the specific design?

(g) What are the normal use conditions of the bookshelf? Are the conditions the same as the condition in which the bookshelf will be used? What are the consequences if any of the conditions is not satisfied?

(h) What are the changes that the bookshelf will bring to you?

**(1–11)** Design a software system and present your design in a document.

(i) Does your document contain all the elements discussed in section 1.3?

(ii) Replace the term 'bookshelf' in the questions (a)–(h) of exercise (1–10) with the software that you design. Then, find their answers from the design document.

(ii) Discuss the consequences if one element is missing from the document.

**(1–12)** Give an example of a software system that was considered as a good design several years ago, but now it is considered as out of date. Discuss the reasons why it happened by referring to Mayall's axioms.

**(1–13)** Give an example of a software system that was considered as a bad design several years ago, but now has become widely used and considered as a good design. Discuss the reasons why it happened by referring to Mayall's axioms.

**(1–14)** Give an example of information system for each type of automational, informative, and transformational information systems.

**(1–15)** A hospital would like to develop an online patient information system. Discuss the stakeholders who might be involved in the development of the system.

**(1–16)** Use the record of the design process that you made in answering question (1–10) to examine and explain Mayall's axiom of designs.

## REFERENCES


1 Jones, C. J., *Design Methods: Seeds of Human Futures,* John Wiley & Sons, 1970.

2 McPhee, K. , *Design Theory and Software Design*, Technical Report TR 96-26, October 1996, Department of Computing Science, The University of Alberta, Canada, 1996.

3 Lawson, B., *How Designers Think*, The Architectural Press Ltd., London, 1980.

4 Dasgupta, S. , The structure of design processes, in *Advances in Computers*, Yovits, M. C., Ed., Academic Press, 1989, pp1–67.

5 Willem, R. A. , Design and science, *Design Studies*, Vol. 11, No. 1, pp43–47, 1990.

6 Willem, R. A. , Varieties of design, *Design Studies*, Vol. 12, No. 3, pp132–136, 1991.

7 MacLean, A. , Bellotti, V. and Young, R., What rationale is there in design?, in *Human-Computer Interaction – INTERACT'90*, 1990, pp207–212.

8 Simon, H. A. , The structure of ill-structured problems, *Artificial Intelligence*, Vol. 4, pp181–200, 1973.

9 *Building for People*, 1965 Conference Report, UK Ministry of Public Building and Works, London, 1965.

10 Freeman, P. , The nature of design, in *Tutorial on Software Design Techniques*, Freeman, P. and Wasserman, A. I. Eds, IEEE, 1980, pp46–53.

11 Mostow, J. , Toward better models of design process, *AI Magazine*, Vol. 6, No. 1, pp44–57, Spring, 1985.

12 Alexander, C., *Notes on the Synthesis of Form*, Harvard University Press, Cambridge, Mass., 1964.
13 Edison, T. A., Improvement in electric lights, United States Patent Office, No. 214,636, April 22, 1879.
14 Simon, H. A., The structure of ill-structured problems, in *Developments of Design Methodology*, Cross, N. (Ed.), pp145–166, Wiley, 1984.
15 Rittel, H. J. and Webber, M. M., Planning problems are wicked problems, in *Developments in Design Methodology*, Cross, N. (ed.), pp135–144, Wiley, 1984.
16 Edison, T. A., Manufacture of incandescent electric lights, United States Patent Office, No. 411,019, Sept. 17, 1889.
17 Edison, T. A., System of electric lighting, United States Patent Office, No. 446,666, Feb. 17, 1891.
18 Mayall, W. H., *Principles in Design*, Design Council, London, 1979.
19 Cross, N. (Ed.), *Developments of Design Methodology*, Wiley, 1984.
20 Freeman, P. and Wasserman, A. I. (Eds.), *Tutorial on Software Design Techniques*, IEEE, 1980.