Basic Concepts of Design

Lecture 1

In this lecture you will learn:

- What is design?
  - Characteristics of design activities
  - Elements of designs
  - Factors that effect design processes and outcomes
- What is a good design?
  - Software quality
  - Effects of design on software quality
  - Quality attributes of software design

Introduction

- How to develop information systems to meet a given requirements specification?
  - Detailed Design
  - Coding
  - Project Management
  - Development Process
  - Architectural Design
  - Testing, Verification and Validation

What is Design?

- The Plan facet:
  - Design as **Noun**
  - The plan of bringing about a man-made artifact
  - The outcome of design activities
- The Process facet:
  - Design as **Verb**
  - A sequence of activities of design
  - The process that a plan is made

As a noun:

A drawing or pattern showing how something is to be made;

The art of making such drawings or patterns;

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;

A decorative pattern, esp. one that is not repeated;

A plan in the mind.

As a verb:

To make a drawing or pattern of (something that will be made or built); develop and draw the plans for;

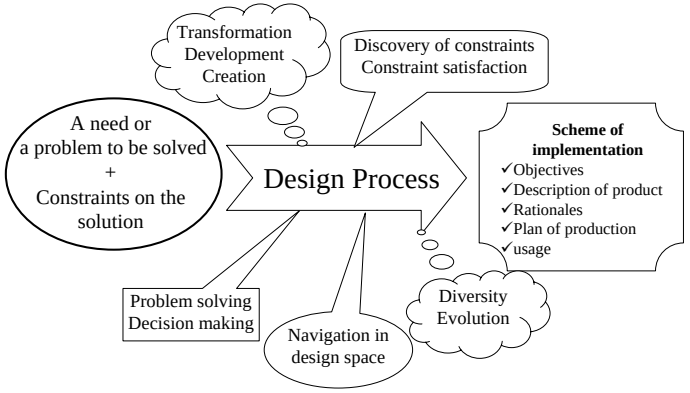
To plan or develop for a certain purpose or use.

[Dictionary of Contemporary English (Longman,1987)]

Design Methodology

- Emerged in 1960s as an independent scientific discipline
- Studies the fundamental questions about design in the widest context
  - What are the essential characteristics of design?
    - What are design activities?
    - How design activities are interrelated to each other?
  - What processes are used by designers?
    - 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 Process Facet



Characteristics of Design Activities

- Design starts with a need and requires intention
- Design results in a scheme for implementing an artefact
- Design involves transformation
- Generation of new idea is fundamental to design
- Design is problem solving and decision making
- Design must satisfy constraints and discovers constraints
- Design is to achieve optimality in a solution space of diversity
- Design demonstrates an evolution process

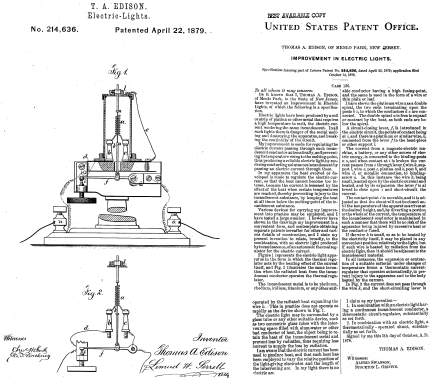
Elements of Designs

- Statement of the Objectives
- Constraints on the Design
- Description of the Product
- Plan of Production
- Rationale of the Design
- Conditions of Usage

1. Statement of the Objectives
   - Description of what is to be achieved by the designed product
2. Constraints on the Design
   - Description of the constraints under which the designed product achieves the objectives
3. Description of the Product
   - Description of the product to be produced
4. Plan of Production
   - Description of how the product is to be produced
5. Rationale of the design
   - Statements that justify why the designed product and its associated production plan can achieve the objectives and satisfy the constraints
6. Conditions of Usage
   - Description of the conditions under which the product should be used

Example

- Edison (愛迪生) 's designs of electric lights



T. A. EDISON.
Electric-Lights.
No. 214,636. Patented April 22, 1879.

Fig 1.

Fig 2.

Witnesses:
Inventor:
Thomas A. Edison

BEST AVAILABLE COPY
UNITED STATES PATENT OFFICE.

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

IMPROVEMENT IN ELECTRIC LIGHTS.

Example: Objective

- A design may have a number of different objectives that must be satisfied as a whole
- Design objectives is often expressed in a number of different levels of abstraction
- Between different levels of objectives, there may have some assumptions

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.

Example: Three Dimensions of Constraints

- The generator of the constraints
- The domain of the constraints
- The function of the constraints

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.

The generator of the constraints:

- The eventual users
- Legislators
- The design clients
- The designers themselves

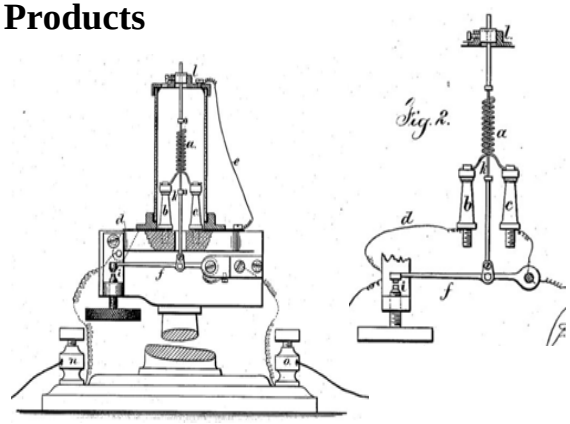
The domain of the constraints:

- External: imposed by the factors not under the designer's control
- Internal: the designer has at least some ability to control

The function of the constraints: relates to the rationales behind the imposition of each of the constraints

- Symbolism and social norms
- Formal intentions of the designer
- Practical implications brought by the implementation technologies
- 'Radical' reasons that deal with the primary purpose of the artefact

Example: Descriptions of Designed Products



U08182 © Peter Lo 2010 11

Multiple Views

- Using different notations to describe different aspects of a design

Structural or Hierarchical Representation

- Representing a complicated design at a number of different abstraction levels and with different details

Example: Rationale of the Design

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.

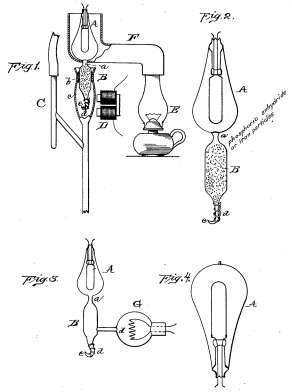
If the wire *k* is small, so as to be heated by the electricity itself, it may be placed in any convenient position relatively to the light; but if such wire is heated by radiation from the electric light, then it should be adjacent to the incandescent material.

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.

U08182 © Peter Lo 2010 12

- Engineering design must be based on scientific principles and technical information.
- A part 'design space': a body of knowledge about artefact, its environment, its intended use, and the decisions that went into creating the design.
- A kind of meta-knowledge about how they arrived at a particular design.

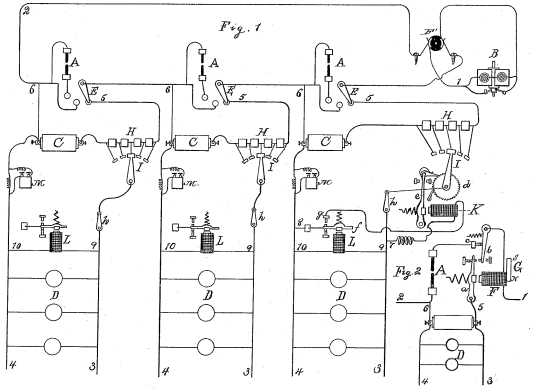
Example: Description of Production



U08182 © Peter Lo 2010 13

The description of the production of a product can be as complicated as, even more complicated than the product.

Example: Description of Usages



U08182 © Peter Lo 2010 14

Definition of Engineering Design

- **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 pre-specified constraints.

Factors that Effect Designs

- Mayall's Axioms on Principles of Design
  - The Principle of Totality
  - The Principle of Time
  - The Principle of Value
  - The Principle of Resources
  - The Principle of Synthesis
  - The Principle of Iteration
  - The Principle of Change
  - The Principle of Relationships
  - The Principle of Competence
  - The Principle of Service

1. **The Principle of Totality -** All design requirements are always interrelated and must always be treated as such throughout a design task.
2. **The Principle of Time -** The features and characteristics of all products change as time passes.
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.
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.
5. **The Principle of Synthesis -** All features of a product must combine to satisfy all the characteristics we expect it to posses with an acceptable relative importance for as long as we wish, bearing in mind the resources available to make and use it.
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.
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.
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 judgment and protect his interests.
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.
10. **The Principle of Service -** Design must satisfy everybody, and not just those for whom its products are directly intended.

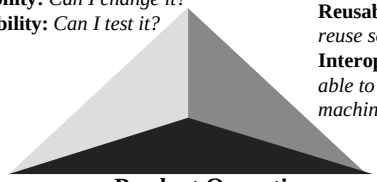
What is a Good Design?

- General Theory of Quality
  - The nature of quality
  - Different people have different views of the quality of a product
  - Producer's view
  - Quality can be decomposed into a set of measurable quality attributes
- Application of Theories to Software Design
  - Software quality models
  - Effect of design on software quality
  - The quality attributes of software design

McCall's Hierarchical Software Quality Model

Product Revision
Maintainability: *Can I fix it?*
Flexibility: *Can I change it?*
Testability: *Can I test it?*

Product Transition
Portability: *Will I be able to use it on another machine?*
Reusability: *Will I be able to reuse some of the software?*
Interoperatability: *Will I be able to interface it with another machine / software?*



Product Operation
Correctness: *Does it do what I want?*
Reliability: *Does it do it accurately all the time?*
Efficiency: *Will it run on my machine as well as it can?*
Integrity: *Is it secure?*
Usability: *Can I run it?*

McCall's quality factors were proposed in the early 1970s. They are as valid today as they were in that time. It's likely that software built to conform to these factors will exhibit high quality well into the 21st century, even if there are dramatic changes in technology.

Product Operations – Operation Characteristics

- ***Correctness***: The extent to which a program satisfies its specification and fulfills the customer's mission objectives.
- ***Reliability***: The extent to which a program can be expected to perform its intended function with required precision.
- ***Efficiency***: The amount of computing resources and code required by a program to perform its function.
- ***Integrity***: The extent to which access to software or data by unauthorized persons can be controlled.
- ***Usability***: The effort required to learn, operate, prepare input, and interpret output of a program.

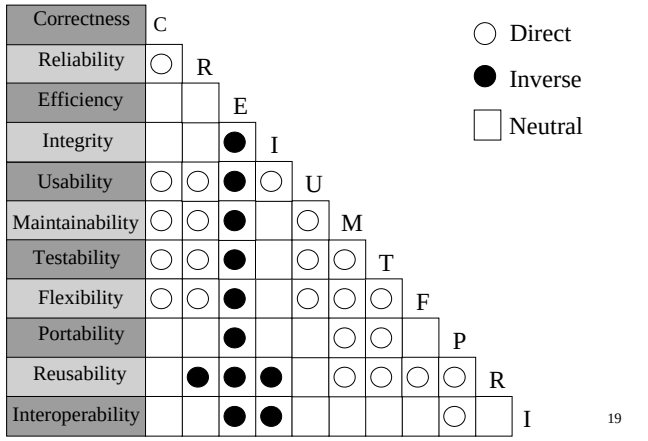
Product Revision – Ability to Undergo Change

- ***Maintainability***: The effort required to locate and fix an error in a program.
- ***Flexibility***: The effort required to modify an operational program.
- ***Testability***: The effort required to test a program to ensure that it performs its intended function.

Product Transition – Adaptability to New Environments

- ***Portability*:** The effort required to transfer the program from one hardware and / or software system environment to another.
- ***Reusability*:** The extent to which a program (or parts of a program) can be reused in other applications - related to the packaging and scope of the functions that the program performs.
- ***Interoperability*:** The effort required to couple one system to another.

Perry's Relational Model



19

Notes on Software Quality Attributes

- Not all attributes are of equal importance
- An attribute may have different importance in different software systems
- The value of an attribute for a given software may change as time passes
- Relationships between attributes are very complicated; some are positive; some are negative
- Different people may have different view on quality

[Alan Gillies, Software Quality: Theory and Management, Second Edition, International Thomson Computer Press, 1997]

Effects of Design on Quality

Effects on / Apply to	Efficiency	Correctness and Reliability	Portability	Maintainability	Reusability
Architectural Design	✓	✓	✓	✓	✓
Detailed Design	✓	✓	✓		✓
Interface Design		✓			
Implementation	✓				

20

•Software quality must be considered at all phases of analysis, design, implementation, deployment and operation/maintenance.

•Design is the first stage in software system creation in which quality requirements can begin to be addressed.

•Different quality attributes manifest themselves differently during various phases.

•Design is critical to many of the quality attributes of software systems, and these qualities should be designed in and evaluated at the design phase.

•Some quality attributes are not structural design sensitive, and attempting to achieve these qualities or analyse for them through structural design means is not fruitful.

Effects on Efficiency

- Performance
  - The responsiveness of a system, i.e. the time required to respond to stimuli (events), or the number of events processed in certain interval of time.
- Effects on Efficiency apply to
  - Architectural Design
  - Detailed Design
  - Implementation

Effects of Architectural Design

Performance often depends on how much communication and interaction between the components of the system

Performance also depends on how components are distributed on network and whether they are executed in parallel

Effects of Detailed Design

Once given an architectural design, the performance depends on the choice of algorithms to implement selected functionality assigned to the components.

Effects of Implementation

Performance also depends on how the algorithms are coded.

Effects on Correctness and Reliability

- Correctness and Reliability
  - The properties that relate to how well the software implements the specified users' requirements.
- Effects on Correctness and Reliability apply to
  - Architectural Design
  - Detailed Design
  - Interface Design

Effects of Architectural Design

A good architectural design provides an understandable solution of the problem, hence helps to reduce the probability of errors made at lower level design and implementation.

Well-structured design helps at testability and maintainability so that errors in design and implementation can be tested and fixed.

Software fault-tolerant features can be introduced in architecture design to detect failures and to recover from failures at run-time, hence improve the reliability of the system.

Effects of Detailed Design

Detailed designs have great impact on the probability that the programmer correctly implement the algorithms.

Effects of Interface Design

Good HCI design can prevent invalid input and misinterpretation of output, hence significantly reduce the probability of system failures caused by human operation errors.

Effects on Portability

- Portability
  - The easiness of a software system to be transported from one hardware / software platform to another.
- Effects on Portability apply to
  - Architectural Design
  - Detailed Design

Effect of Architectural Design

A well structured design should group environment dependent code in a small number of components so that the change on the code to move to another environment can be achieved by replacing such components with new ones rather than rewrite the whole systems.

Effect of Detail Design

Algorithms and data structures should not heavily depend on the platform specific feature so that portability can be obtained.

Effects on Maintainability

- Maintainability
  - The easiness of maintain a software system in both corrective and adaptive maintenances.
  - Both corrective and adaptive types of maintenance operations require software engineers understand how the software system works so that bugs can be fixed and changes in the environment can be adapted.
- Effects on Maintainability apply to
  - Architectural Design

Effect of Architectural Design

Well-structured design helps software engineers to understand the system.

Effects on Reusability

- Reusability
  - The property that the components of a software system can be easily reused in the development of other software systems.
  - Reusability depends on the generality of the components in a given application domain and the extent to which the components are parameterised and configurable.
- Effects on Reusability apply to
  - Architectural Design
  - Detailed Design

Effect of Architectural Design

It determines how the functionality of a software system is decomposed into components and how they are inter-connected.

It play important role in component-based software developments.

Effect of Detailed Design

Algorithm and data structure design determine how easily the components can be parameterised and the way to configure the components.

Quality Attributes for Design

- Design Quality Attributes
  - The product-oriented quality attributes
    - The quality attributes associated to the description of the product and defined in terms of the model of the solution
  - The process-oriented quality attributes
    - The quality attribute associated to the process facet of the design and defined in terms of the process

Design has two Facets:

- The product:
  - The description of the solution to the problem and/or the model of the product
- The process:
  - The description and/or model of the plan about how to make the product

Parnas & Weiss' Requirements

- Well Structured
- Simple
- Efficient
- Adequate
- Flexible
- Practical
- Implementable
- Standardized

- **Well Structured -** Consistent with chosen properties such as information hiding
- **Simple -** To the extend of being 'as simple as possible, but no simpler'
- **Efficient -** Providing functions that can be computed using the available resources
- **Adequate -** Meeting the states requirements
- **Flexible -** Able to accommodate likely changes in the requirements, however these might arise
- **Practical -** Module interfaces should provide the required facilities, neither more nor less
- **Implementable -** Using current and available software and hardware technology
- **Standardized -** Using well-defined and familiar notation for any documentation

Process Facet of Design Quality

- Software design does have a process facet, such as:
  - project schedule and milestones
  - development team organisation
  - test plans, etc.

  } *These are essential parts of a software design.*
- The process facet of software design has been studied under separate subject titles in computing science, e.g.
  - Software process models
  - Software project management
  - Software risk management
  - Configuration management
  - Economics of software development

Process-Oriented Quality Attributes

- Feasibility
- Simplicity
- Reliability
- Productivity
- Time to Market
- Risk
- Resourcefulness
- Cost
- Technical Requirements
- Material Requirements
- Legitimacy

- **Feasibility -** Can the development plan bring about the required product?
- **Simplicity -** How complex is the development process to produce the required product? Is the development process manageable?
- **Reliability -** Can the development plan fail? How probable will it fail?
- **Productivity -** How productive would it be if the development follows the development plan?
- **Time to Market -** What is the time to market if the product is developed following the development plan?
- **Risk -** What are the consequences if the development process fails?
- **Resourcefulness -** Does the development process use resources economically?
- **Cost -** What is the cost of the development following the plan?
- **Technical Requirements -** Does it require intensive training to obtain required technical skills for the people to develop the system? Does it rely on expensive equipment and tools?
- **Material Requirements -** Does it use expensive materials to produce the product?
- **Legitimacy -** Is the development process and the product allowable according to law and conformant to applicable standards and regulations?

Further Readings

- Zhu, H., Software Design Methodology - From Principles to Architectural Styles, Chapter 1 and 2.
- Budgen, D, Software Design, Addison-Wesley, 1994. Chapter 1 and 4, pp1~23, 57~76.
- Cross, N., Engineering Design Methods: Strategies for Product Design, Third Edition, John Wiley and Sons, 2000, Part One: Understanding Design.