KU 건국대학교 KONKUK UNIV.

Software Design: An Introduction

[발표 #2] 2010.09. 29 (Wed)

T1	
200511311	김진수
200511350	장범석
200714170	모진종
200714172	배보람 (발표자)

Contents

1. The Role of Software Design
   - 1.1 The software design process
   - 1.2 Design in the software development cycle
   - 1.3 Design qualities
2. Describing Designs
   - 2.1 Recording the design model: design viewpoints
   - 2.2 Design representation forms
   - 2.3 Some examples of design representations
3. Software Design Practices and Design Method
   - 3.1 Rationale for software design method
   - 3.2 Design strategies
4. Features of Some Software Design Method
   - 4.1 Jackson Structured Programming (JSP)
   - 4.2 Structured System Analysis and Structured Design
   - 4.3 Jackson System Development (JSD)
   - 4.4 Object-Oriented Design
5. Question

1. The Role of Software Design

- Q : “What exactly is the purpose of design?

- A : “To Produce a workable solution to a given problem.”

- Fitness for Purpose
  - The Key measure of the appropriateness of any system.

1. The Role of Software Design

1.1 The software design process

- Software design은 왜 이해하기 어렵고 다루기 힘든가?
  - Software properties
    - Complexity
    - Problem of conformity
    - Ease of changeability
    - Invisibility
- Procedural manner
  - Representation part
  - Process part
  - Set of heuristics

1. The Role of Software Design

1.1 The software design process (cont')

- Software designer 두가지 전략
  - to postpone making a decision where information is not yet available
  - 정보를 쓸수 없다면 선택을 미뤄두든가
    Ex) 엄마 너무 공부가 하고 싶은데 문제집이 없네요 정말 아쉬워요
  - to define components for which the information is ready to hand, in anticipation of further developments in the design
  - 사용할 부분들을 예상되는 사용가능 범위로 한정시켜 두던가.
    Ex) 아들아 그러면 일단 네가 가지고 있는
    1992~2010 기출문제집만 싹 풀도록 해라

1. The Role of Software Design

1.2 Design in the software development cycle

Activities of analysis	Software designer
사용자의 요구를 만족할 만한 solution 형식을 식별	해당 양식을 준수하는 solution 제공

- 상호작용 필요 !
  - 요구 사항과 개발 후 솔루션의 불일치를 없애기 위해서

- Designer는 …
  - Provide a set of specifications for those who are to construct a system

    : Clear, Complete, and Unambiguous as possible
  - Needs to "think ahead"

    : in planning a solution

1. The Role of Software Design

1.3 Design qualities

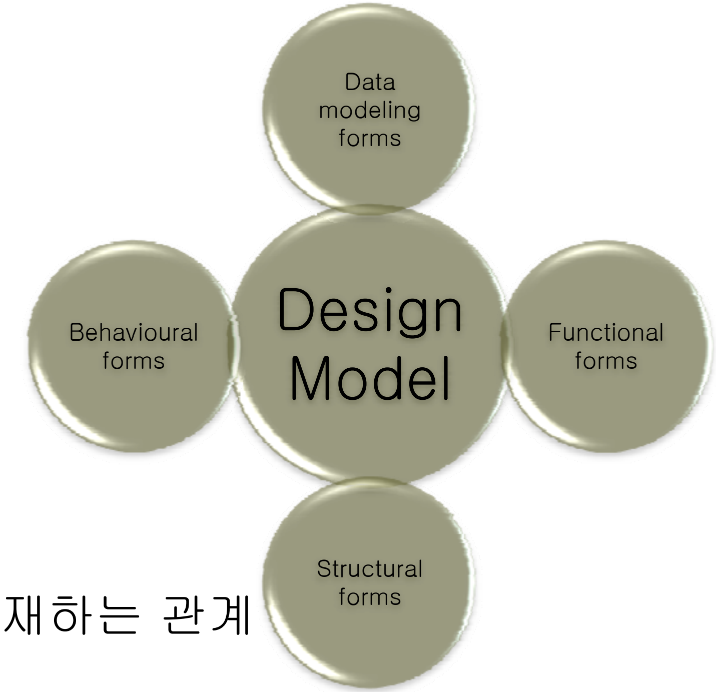
- “fitness for purpose” 가
  design의 quality를 절대적으로 측정할 수는 <u>없다</u> !

- quality factor
  - Reliability
  - Efficiency
  - Maintainability
  - usability

2. Describing Designs

2.1 Recording the design model: design viewpoints

- Behavioural
  - 프로그램 실행 동안 외부 이벤트와 시스템 활동 사이의 연결
- Functional
  - 시스템이 무엇을 하는지
- Structural
  - 시스템 구조상 요소간의 상호 의존
- Data Modeling
  - 시스템에 사용된 data objects 사이에 존재하는 관계



2. Describing Designs

2.2 Design representation forms

- Three forms of description

Textual	✓ heading, lists, and indentation 등을 사용 ✓ Limitation • 정보에 포함된 암시적인 구조가 list와 table에서 쉽게 발견되지 않는다면 애매해질 수 있다. • Natural language는 모호한 경향이 있다.
Diagram	✓ 도식화 하여 만듦 ✓ property • A small number of symbols. (circles, lines, boxes) • A hierarchical structure. • Simplicity of symbol form.
Mathematical	✓ 추상적 ideas를 간결하게 묘사하는데 적합하지만 ✓ designer의 추상적 ideas 항상 묘사할 수 있는 것은 아니다

2. Describing Designs

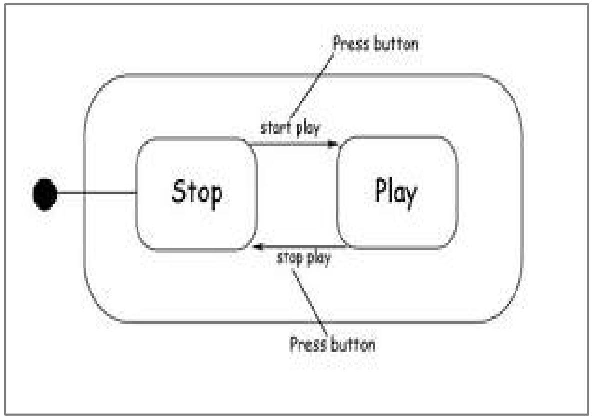
2.3 Some examples of design representations

- The State chart
- The Jackson Structure Diagram
- The Structure Chart
- The Entity-Relationship Diagram

2. Describing Designs

2.3 Some examples of design representations (cont')

- The State chart
  - 한 객체의 상태변화를 다이어그램으로 나타낸 것
  - 모서리가 둥근 사각형으로 표현



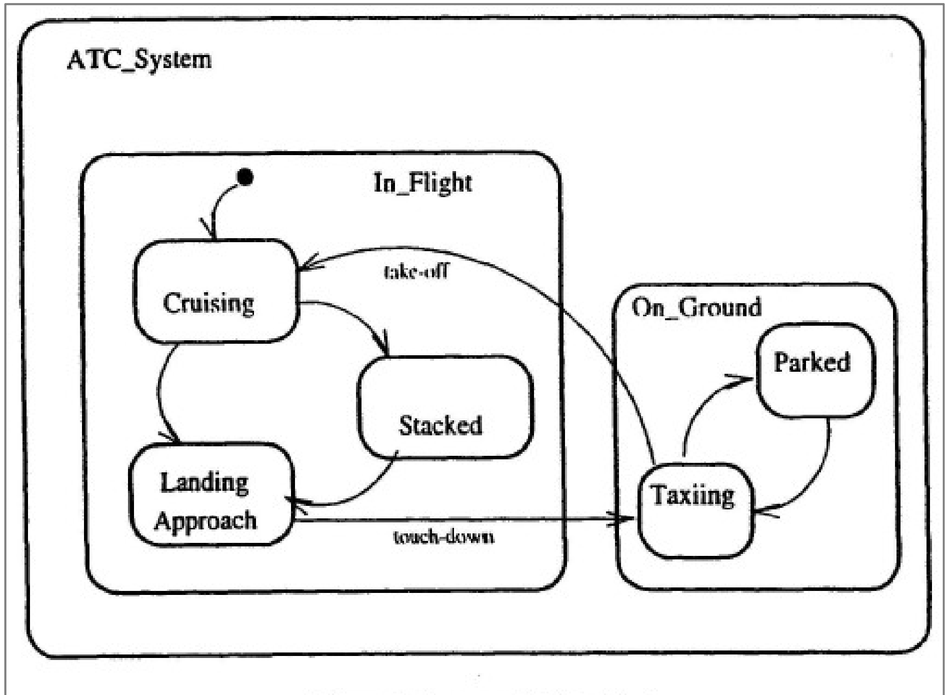
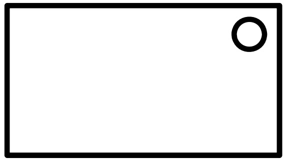


Figure 3. An example Statechart.

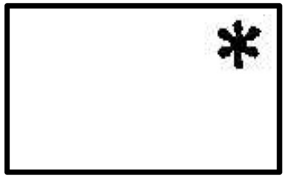
2. Describing Designs

2.3 Some examples of design representations (cont’)

- The Jackson Structure Diagram
  - 순차, 선택, 반복을 통해 데이터 구조를 기술
  - 시간 대로 행동을 묘사하는데 사용
  - 선택



  - 반복



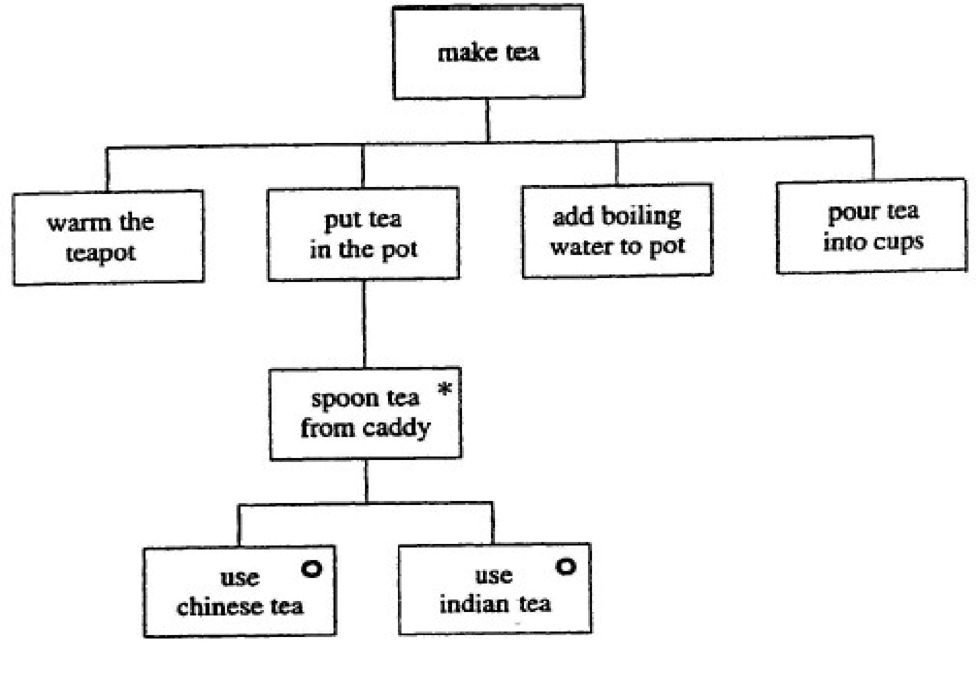


Figure 4. An example of a Jackson Structure Diagram.

2. Describing Designs

2.3 Some examples of design representations (cont')

- The Structure Chart
  - 프로그램 내 모듈 간 호출 구조를 기술
  - Jackson Structure Diagram과 유사해 보이지만 다르다 !
    - The elements represent physical entities (subprograms)
    - Hierarchy shown is one of invocation (transfer of control)

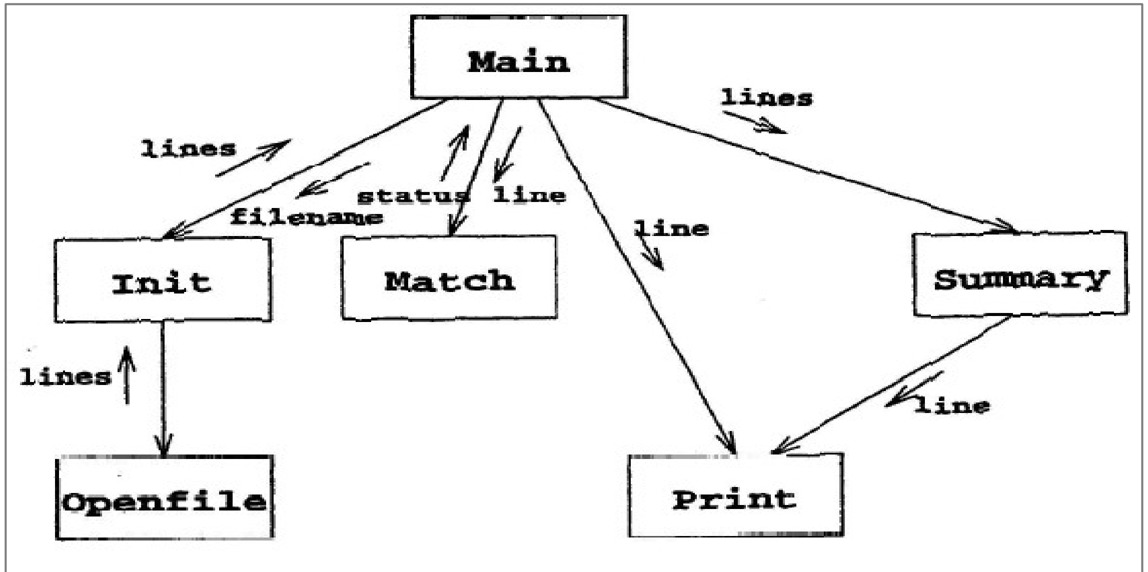
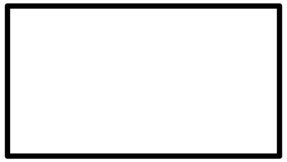


Figure 5. A simple Structure Chart.

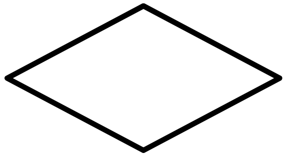
2. Describing Designs

2.3 Some examples of design representations (cont')

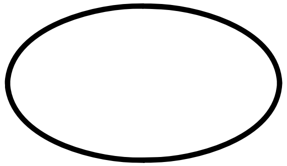
- The Entity-Relationship Diagram (ERD)
  - Data elements 사이에서 발생하는 상호 관계를 표현
  - Entity



Relationship



Attribute



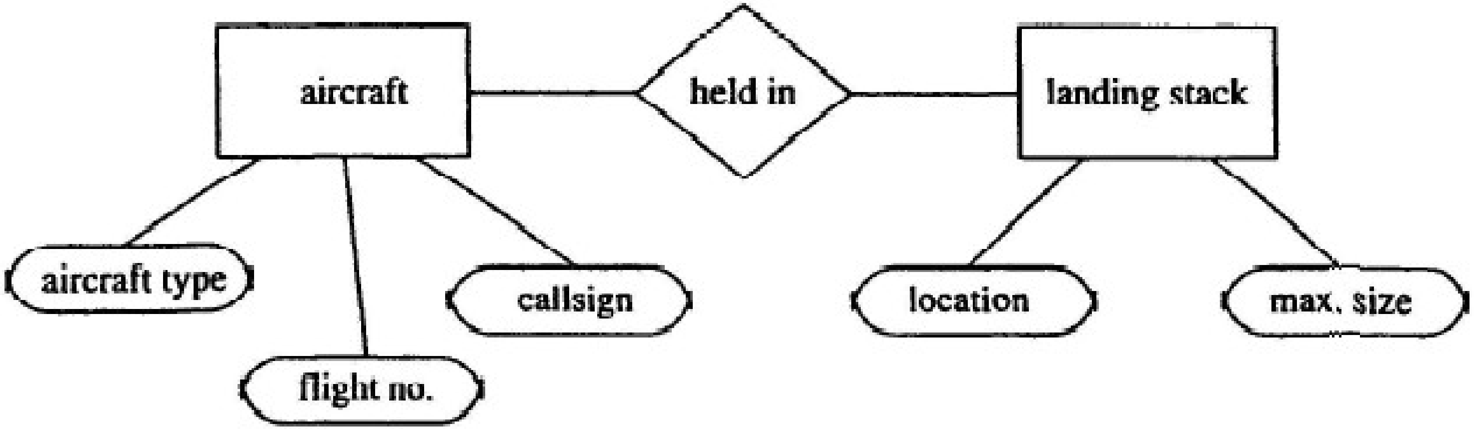


Figure 6. A simple Entity-Relationship Diagram.

3. Software Design Practices and Design Method

3.1 Rationale for software design method

- Roles for Software Design Method
  - Establishing common goals and styles
  - Generating “consistent” documentation
  - Helping to make some of the features of a problem more explicit
- Constraints that limit their usefulness
  - The process part of a method provides relatively little detailed guidance as to how a problem should be solved
  - The Need to Use a Procedural Form Leads to Practices That Conflict with the Behavior Observed in Experienced Designers

3. Software Design Practices and Design Method

3.2 Design strategies

- Top-down
  - Separating a large problem into smaller ones
  - Disadvantage : Important structural decisions may need to be made at an early stage
- Compositional
  - Identifies a set of “entities” that can be modeled and then assembled to create a model for the complete solution
- Organizational
  - Use where the needs of the development organization and its management structures impose constraints upon the design process
- Template
  - Some general paradigm describes a reasonably large domain of problems

4. Features of Some Software Design Method

- Jackson Structured Programming (JSP)
- Structured Systems Analysis and Structured Design
- Jackson System Development (JSD)
- Object Oriented Design (OOD)

4. Features of Some Software Design Method

4.1 Jackson Structured Programming (JSP)

- Is provided by Jackson Structured Diagram
  - used for modeling data structures
  - used for functional modeling

< JSP process part >		
Step 1	Draw Structure Diagrams for inputs and outputs	elaboration
Step 2	Merge these to create the program Structure Diagram	transformation
Step 3	List the operations and allocate to program elements	elaboration
Step 4	Convert program to text	elaboration
Step 5	Add conditions	elaboration

4. Features of Some Software Design Method

4.2 Structured Systems Analysis and Structured Design

- top-down 기법
- consists of “analysis” and “design”
- Using “Data-Flow Diagrams (DFD)”
- this one has strong intuitive attractions
- disadvantage of having a large and relatively disjoint transformation step.

< The process part of SSD and SD >			
Analysis	Step 1	Develop a top-level description	elaboration
	Step 2	Develop a model of the problem (SSA)	elaboration
Design	Step 3	Subdivide into DFDs describing transactions	elaboration
	Step 4	Transform into Structure Charts	transformation
	Step 5	Refine and recombine into system description	elaboration

4. Features of Some Software Design Method

4.3 Jackson System Development (JSD)

- Encourages the designer to create a design model around the notion of modeling the behaviour of active "entities"
- In the initial stages, these entities are related to the problem, but gradually the emphasis changes to use entities that are elements of the solution

< JSD process part >		
1. Entity Analysis	Identify and model problem entities	elaboration
2. Initial Model Phase	Complete the problem model network	elaboration
3. Interactive Function Step	Add new solution entities	elaboration
4. Information Function Step	Add new solution entities	elaboration
5. System Timing Step	Resolve synchronization issues	elaboration
6. Implementation	Physical design mappings	elaboration

4. Features of Some Software Design Method

4.4 Object Oriented Design (OOD)

- OOD is concerned with developing an object-oriented system model to implement requirements

- An object-oriented design process: Structured design processes
  - involve developing a number of different system models
  - require a lot of effort for development and maintenance of these models
  - For small systems, this may not be cost-effective
  - However, for large systems developed by different groups design models are an essential communication mechanism

4. Features of Some Software Design Method

4.4 Object Oriented Design (OOD)

- OOD is concerned with developing an object-oriented system model to implement requirements

< The Fusion design process >

Phase	Step	Action
Analysis	1.	Develop the Object Model
Analysis	2.	Determine the System Interface
Analysis	3.	Development of the Interface Model
Analysis	4.	Check the Analysis Models
Design	5.	Develop Object Interaction Graphs
Design	6.	Develop Visibility Graphs
Design	7.	Develop Class Description
Design	8.	Develop Inheritance Graphs

5. Questions