Software Design

Design...

- Design activity begins with a set of requirements
- Design done before the system is implemented
- Design is the intermediate language between requirements and code
- Moving from problem domain to solution domain
- Proceeding from abstract to more concrete representations
- Result is the design to be used for implementing the system

Design...

- Design is a creative activity
- Goal: to create a plan to satisfy requirements
- Perhaps the most critical activity during system development
- Design determines the major characteristics of a system
- Has great impact on testing and maintenance
- Design document forms reference for later phases

Levels in Design Process

- Architectural design
  - Identifies the components needed for the system, their behavior, and relationships
  - We have already discussed it
- High Level Design
  - Really is the module view of the system
  - I.e. what modules are in the system and how they are organized

Levels..

- Logic design
  - Components and modules are designed to satisfy their specs
  - How to implement components
  - Algorithms for implementing component are designed
- Complete design: the architectural design, the high level design, and Logic design of each component

Design Methodologies

- Many possibilities for design, methodologies aim to reduce search space
- Provide some discipline for handling complexity
- Most methodologies deal with high level design
- Provide a set of rules for guiding the designer
- Rules do not reduce design to a sequence of mechanical steps
- Many methodologies exist
- Different methodologies may be useful for different applications

Design Objectives

- Goal is to find the best possible design
- Have to explore different designs
- Evaluation criteria are often subjective and non quantifiable
- Major criteria to evaluate a design
  - Correctness
  - Efficiency
  - Maintainability
  - Cost

- Correctness is the most fundamental
  - Does design implement requirements?
  - Is design feasible, given the constraints?
- Efficiency
  - Concerned with the proper use of scarce resources - processor & memory
  - Other factors same, efficiency should be maximized

- Maintainability
  - Most important quality criteria
  - Most affected by architectural design
  - Should facilitate testing
  - Should facilitate discovery and correction of bugs
  - Make modifying the system easier
- Cost
  - For same quality, minimize cost
  - Design costs are quite small
  - Should try to minimize cost in later phases

Design Principles

- Design is a creative process
- How to create a design from abstract requirements
- There are principles for guiding during design
- Two fundamental principles in the design process
  - Problem partition
  - Abstraction

Problem Partitioning

- Basic principle "divide and conquer"
- Divide the problem into manageably small pieces
  - Each piece can be solved separately
  - Each piece can be modified separately
  - Pieces can be related to the application
- Pieces cannot be independent; they must communicate
- Communication adds complexity
- As number of components increases, this cost increases
- Stop partitioning when cost is more than benefit

Abstraction

- Necessary for partitioning
- Used in all engineering disciplines
- Abstraction of existing components
  - Represents components as black boxes
  - Hides the details, provide external behavior
  - Useful for understanding existing systems
  - Necessary for using systems
  - Useful for determining design of existing systems

- Abstraction during design process
  - To decide how components interact, the designer specifies the external behavior of components
  - Allows concentrating on one component at a time
  - Permits a component to be considered without worrying about others
  - Allows designer to control the complexity
  - Permits gradual transition from abstract to concrete

Functional Abstraction

- Employs parameterized subprograms
- Specifies the functional behavior of a module
- Module is treated as a input/output function
- Most languages provide features to support this eg functions, procedures
- A functional module can be specified using pre and post conditions

Data Abstraction

- An entity in the real world provides some services to the environment it belongs
- Similar is the case of data entities
- Certain operations are required from a data object
- The internals are not of consequence
- Data abstraction supports this view
  - Data is treated as a set of pre-defined operations
  - Only operations can be performed on the objects
  - Internals are hidden and protected
- Modern languages support data abstraction eg. Ada, C++, Java

Top-Down vs Bottom-up Design

- Top down design starts with the system specifications
- Defines a module to implement the specs
- Specifies subordinate modules
- Then treats each specified module as the problem
- Refinement proceeds till bottom level modules reached
- At each stage a clear picture of design exists
- Most natural for handling complex problems
- Have been propagated by many
- Many design methodologies based on this
- Feasibility is not know till the end

- In bottom up we start by designing bottom modules
- Building blocks Layers or abstraction or virtual machines
- Necessary if existing modules have to be reused
- Pure top-down or bottom-up is not possible
- In bottom-up must have some idea of the top
- Often a combination is used

Modularity

- A concept closely tied to abstraction
- Modularity supports independence of models
- Modules support abstraction in software
- Supports hierarchical structuring of programs
- Modularity enhances design clarity, eases implementation
- Reduces cost of testing, debugging and maintenance
- Cannot simply chop a program into modules to get modularly
- Need some criteria for decomposition: coupling and cohesion.

Detailed Design

- HLD does not specify module logic
- This is done during detailed design
- Process Design Logic (PDL) can also be used for detailed design of modules
- PDL can be used to specify the complete design - architectural as well as logic design
- The degree of detail desired is decided by the designer

- One way to communicate a design: use natural language
- Is imprecise and can lead to misunderstanding
- Other extreme is to use a formal language
- Such representations often have a lot of detail,
  - necessary for implementation, but
  - not important for communicating the design
- These details are often a hindrance to understanding

- Ideally would like a language that is
  - as precise as possible
  - Does not need too much detail,
  - target language independent
  - can be easily converted in to an implementation
- This is what PDL attempts to do.
- PDL has outer syntax of a structure programming language, but vocabulary of a natural language
- It can be thought as "structured english"
- Some automated processing can be done on PDL

- E.g.. determine the min and max of a set of numbers
- A design in PDL is:

```
minmax (in file)
ARRAY a
   DO UNTIL end of input
     READ an item into a
   ENDDO
   max, min: = first item of a
   DO FOR each item in a
     IF max < item THEN set max to item
     IF min > item THEN set min to item
   ENDDO
END
```


- The entire logic is described
- Few implementation details
- For implementation, the pseudo statements will have to be converted into programming language statements
- PDL allows a successive refinement approach
- Encourages use of structured language constructs

Design Verification

- Main objective: does the design implement the requirements
- Analysis for performance, efficiency, etc may also be done
- If formal languages used for design representation, tools can help
- Design reviews remain the most common approach for verification

Summary

- Design is a creative activity
- Goal is to find the best possible design
- Two levels in the design process
- Architectural design and logic design
- Correctness of design is most fundamental property
- Design principles
  - Problem partitioning
  - Abstraction
- When using functional abstraction aim for
  - Low coupling
  - High cohesion
- Design Methodologies - a set of rules/steps to guide the designer