CS3203: Software Engineering Project

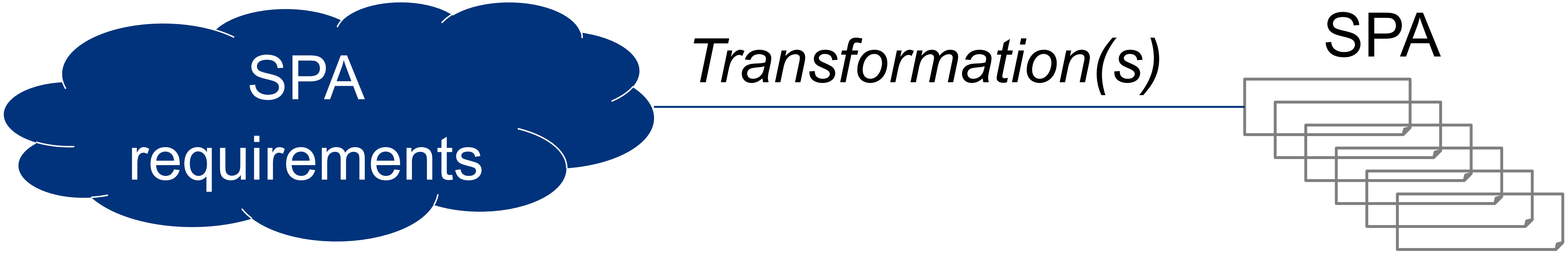
Design Decisions

By: Dr. Bimlesh Wadhwa

NUS
National University of Singapore

School of Computing

SPA Requirements to Functional SPA



The Design Process

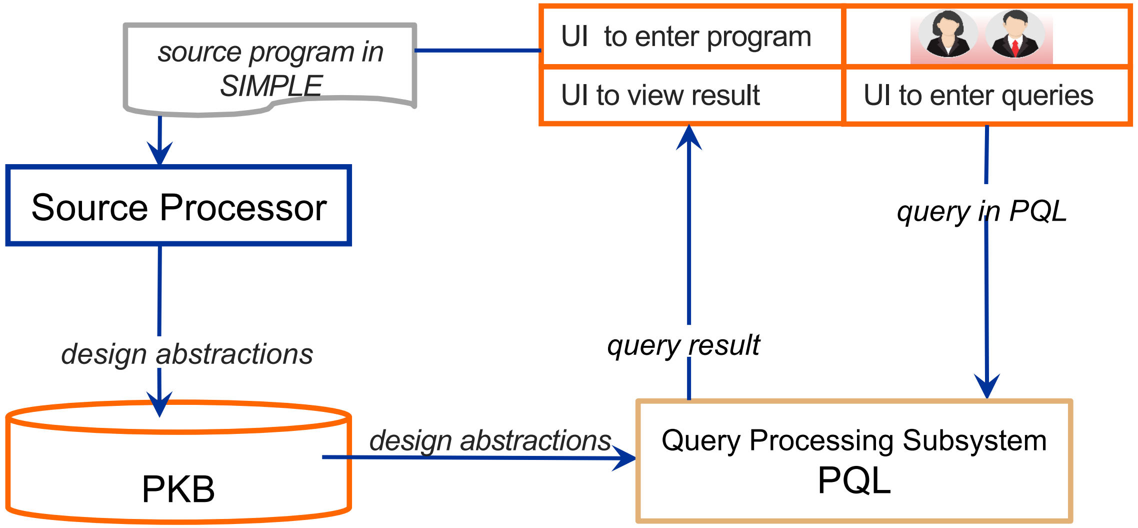
- Design is an iterative problem-solving process to find and describe a way:
  - To implement the functional requirements
  - To fulfil the non-functional requirements
- The result:
  - Design Specification(aka Design Models)
    - Architecture Design, Interface Design, Component Design, Data Design, Algorithmic design
  - A set of design decisions

Design Specification

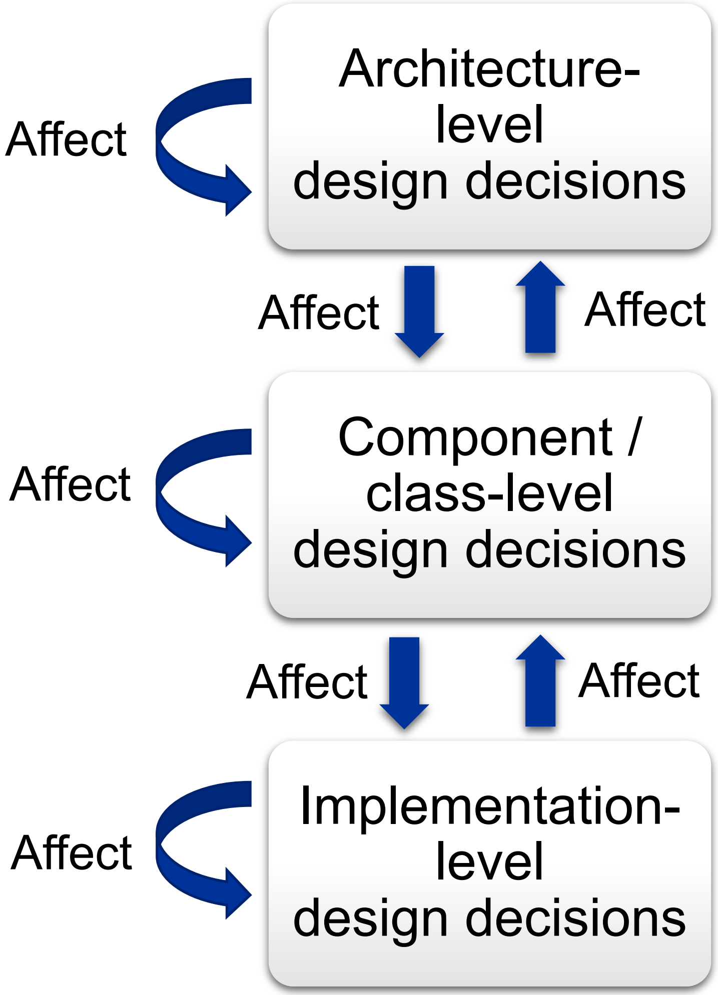
- Architectural design
  - overall structure of the system, the principal components (sometimes called sub-systems or modules), their relationships, and how they are distributed.
- Interface design
  - unambiguous interfaces between system components.
- Component design
  - Specification of component’s behaviour
- Data design
  - data structures and their representations

Design Decisions

- A designer is faced with a series of *design issues*
  - These are sub-problems of the overall design problem
  - Each issue normally has several possible solutions (*design options*)
  - The designer makes a *design decision* to resolve each issue
- Choosing the best option among the possible options



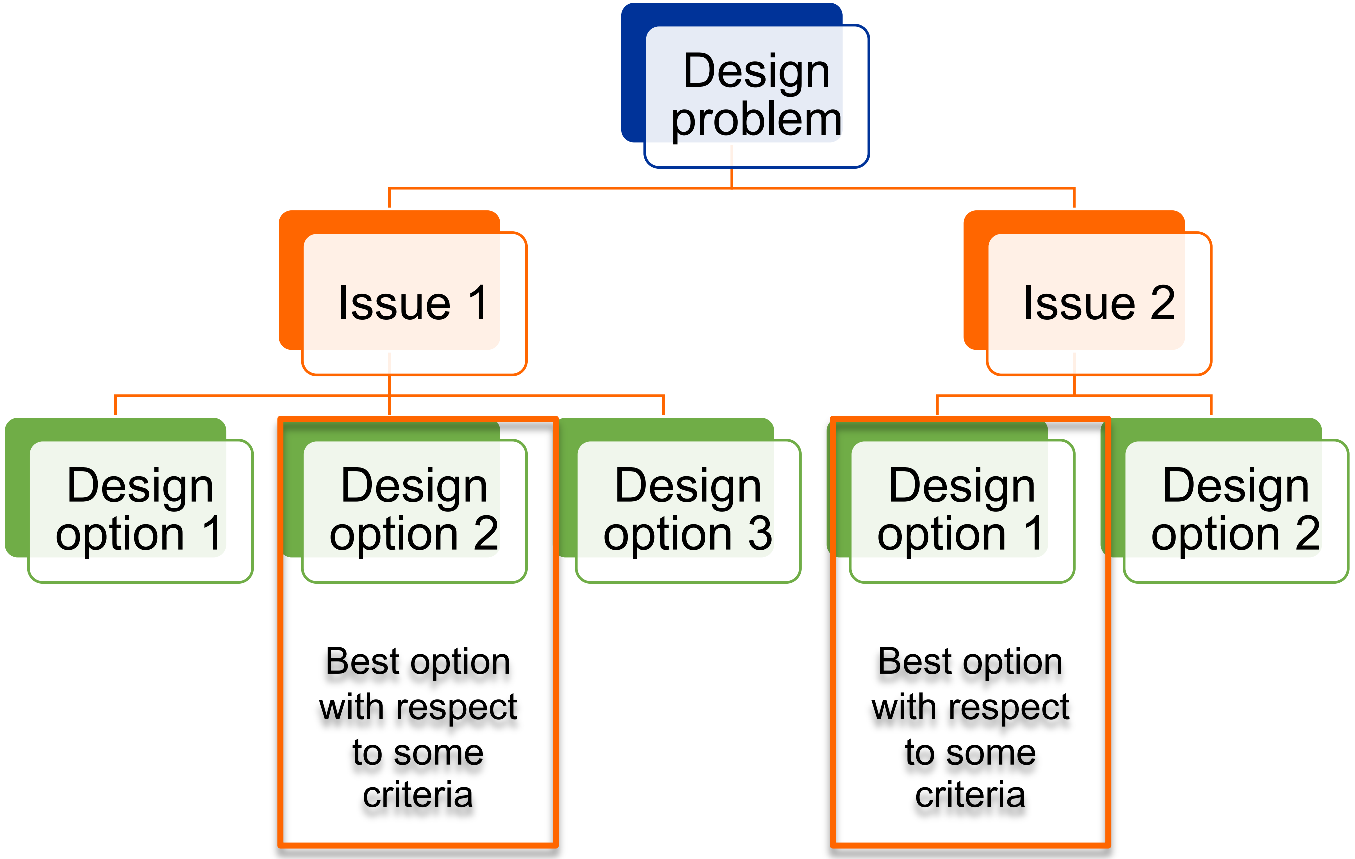
Making Design Decisions


Affect
Architecture-level design decisions
Affect
Affect
Affect
Component / class-level design decisions
Affect
Affect
Affect
Implementation-level design decisions

There are many solutions to each design problem – how do we make right design decisions?

Making Design Decisions

- Questions
  - What are the key design issues?
- Options
  - What are the possible answers?
- Criteria
  - How to assess and compare the options?



Example – sub-componentization

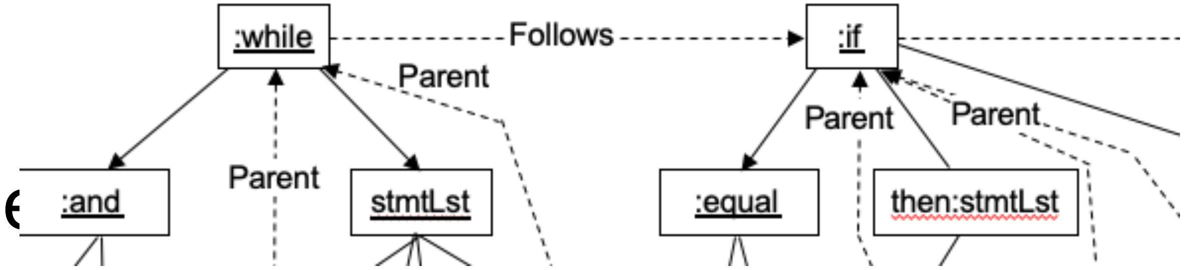
- Architectural Design decision
  - Performance
    - » the architecture should be designed to localize critical operations within a small number of components
    - » relatively large components rather than small, fine-grain components, which reduces the number of component communications.
  - Maintainability
    - » architecture should be designed using small, self-contained components that may readily be changed.
    - » Producers of data should be separated from consumers and shared data structures should be avoided.
- Conflicting options:
  - large components VS small components

→ If both performance and maintainability are important system requirements, then some compromise must be found.

→ using different architectural patterns or styles for different parts of the system.

Example - AST or no AST

- Should we implement an AST?
- AST usage: Follows, Parent
- Is there any alternative for these design abstractions ?
  - Table structure
- Criteria to analyse the options
  - Complexity of implementation
  - Performance at retrieval
  - Extendibility to Iterations 2 and 3



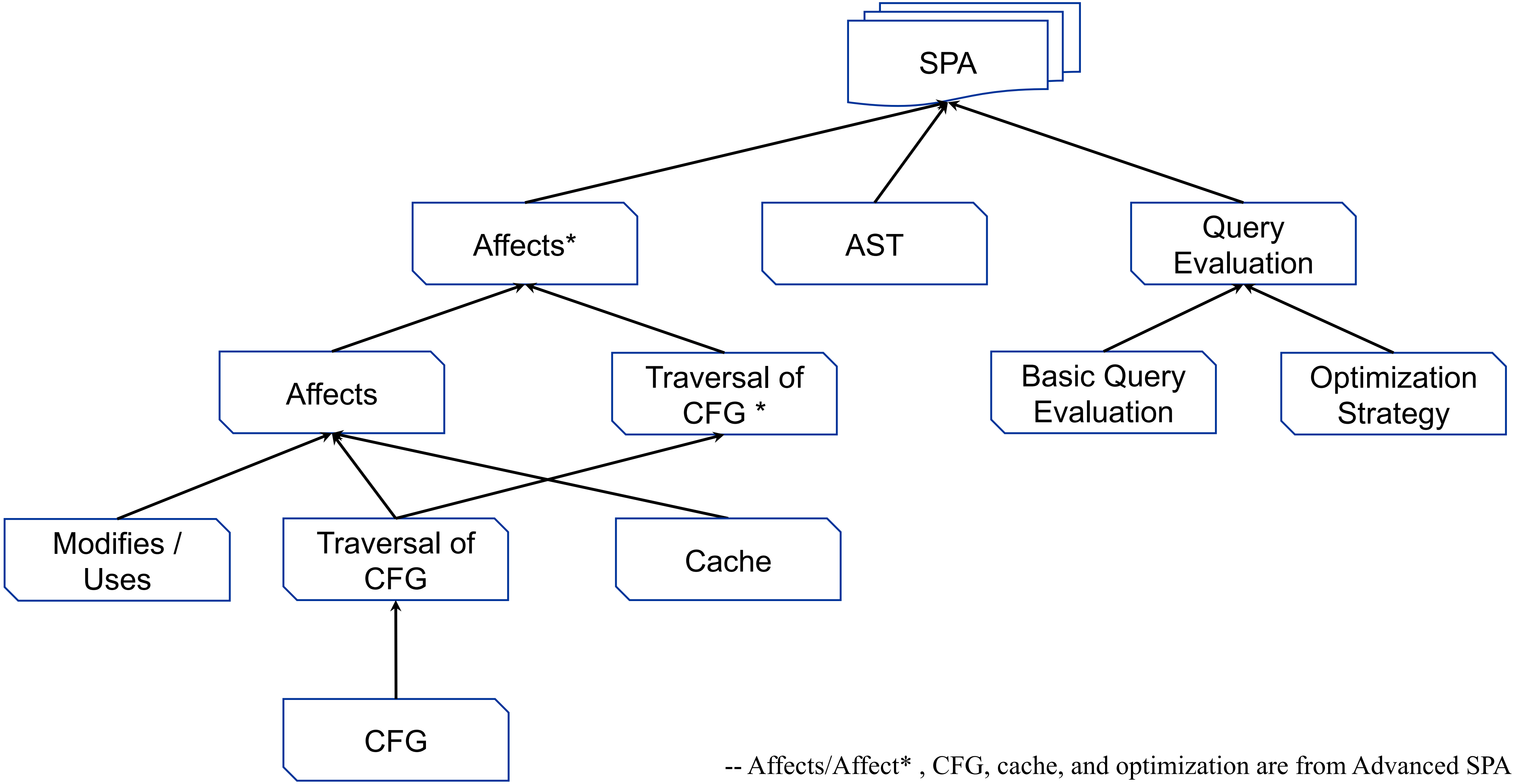
S1	S2
1	2
2	3
….	….

S1	S2
5	6,7,10
7	8,9
….	….

Example – Data structure for Uses

- What data structure should be used to store Uses?
- Options
  - Linked List
  - Byte Array
  - Bit Vector
- Criteria to analyse the options
  - Simplicity of Implementation
  - Performance
  - Memory Utilization
  - Impact on other decisions

Inter-dependent Design Decisions



-- Affects/Affect* , CFG, cache, and optimization are from Advanced SPA terms

Making Design Decisions

- current knowledge
  - the requirements
  - the design as created so far
- past knowledge
  - the technology available
  - what has worked well in the past
  - software design principles and "best practices"

Communicating Design Decisions

- Oral Presentations
  - Architectural design decision - sub-componentization and component interactions
  - Component design decisions- Choice of Data Structure and Algorithm
    - e.g. Which parsing algorithm ? How is Syntactic and semantic validation handled ? Data structure choices?
  - SPA quality decisions - Test case and Test Suite design

- Documenting Design Decisions in SPA iteration Reports

  For each significant Design Problem you handle
  - Alternative solutions
    - Enumerate the possible solutions
  - **Compare and contrast** solutions based on different criteria
    - Choose a set of criteria essential in the context
  - For each alternative solution, explain the positive and negative aspects
  - **Decision and justification**
    - State the “winning” alternative and explain how to mitigate its cons

Design Principles and their benefits

- Abstraction
- Separation of Concerns
- High Cohesion
- Low Coupling
- Encapsulation
- Information Hiding
- SOLID (SRP, OCP, LSP, IS, DI)

Benefits :
Manage complexity, Reduce component dependencies, easier to understand, maintain, and extend, shared vocabulary for early feedback

Summary

- The design process and design decisions
  - knowledge of best practices
  - requirements and design at hand
- Examples of design decisions for SPA
- Communicate your design decisions
  - Problem, options, criteria, decision