SFWR ENG 3A04: Software Design II

Dr. Ridha Khedri

Department of Computing and Software, McMaster University
Canada L8S 4L7, Hamilton, Ontario

Term 1, 2008–2009
Outline of Part I

1. Questions???
Outline of Part II

2. Overview

3. Where does software design sit in SDLC?

4. What is meant by Software Architecture?

5. Software Architecture Design Guidelines
Part I

Review of Previous Lecture
Overview

Where does software design sit in SDLC?

What is meant by Software Architecture?

Software Architecture Design Guidelines

Part II

Today’s Lecture
Introduction to Software Architecture  

**Overview**

- During the last half century, the world witnessed growth in the scale, complexity, and space distribution of software systems.

- When a system is complex in its functionality, its aspects (qualities), and its environment, an obvious need to reduce that complexity emerges.

- Gives the need to a top-down strategy in software construction to divide and conquer the complexities faced by the constructors.

- Roughly speaking, the decomposition of a system/artefact into “manageable” parts is one of the aims of design.
What is Design?

- One cannot define design as would define a mathematical activity such as “rewriting”, “derivation”, etc.

- The nature of design is not easy to grasp in a clear definition

- Looking for a definition of design may not help you grasp what it is

- Design can be seen in any human artefact and in nature

- The single word “design” encompasses an awful lot of objectives and subjective aspects
They can be

- aesthetic (subjective)
- functional (Objective)
- many other aspects of an object or a process, which usually require considerable research, thought, modelling, interactive adjustment, and re-design (Objective or subjective)
Depending on the designed entity, either the objective or the subjective aspects might take-over.

When the subjective aspects take over, design is then viewed as a more rigorous form of art, or art with a clearly defined purpose.

When the objective aspects take over, design is then viewed as a simple mathematical transformation of the mathematical entities expressing the functional requirements into a design which is a collection of mathematical entities with their mathematical connectors.

So, design involves both mathematics and art as transformation means (and may be other skills?!!?)

Software Architecture Design Guidelines
In the literature, you find some unsuccessful attempts to define the term design:

- In some literature, you find that design is “making things better for people”

- But, for instance, we all could say that we too “make things better for people”!!!!!!!

- “Design is that area of human experience, skill and knowledge which is concerned with man’s ability to mould his environment to suit his material and spiritual needs.”
  
  [Archer, B (1973) The Need for Design Education. Royal College of Art]
Introduction to Software Architecture  Overview

- Design could be viewed as an activity that translates an idea/goal/wish into a blueprint for an artefact or a process that is fit for an environment.

- We will focus on examples of design that would give each one of his own understanding of design (which enhances creativity).

- We however agree that the obtained artefact or process is suitable for (fit for use in) the intended environment.
What is in design more than the translation of an idea to an artefact or process?

- Design can be seen as an activity of reconciliation between conflicting needs or constraints put by the environment.

- Design is a creative process based around the "building up" of ideas.
  - Avoiding negative judgements of contributors when designing eliminates the fear of failure and encourages maximum input and participation.
  - Encourage outside the box thinking in this process since this can often lead to creative solutions.
Introduction to Software Architecture

Overview

- Although design is subject to personal taste, design thinkers share a common set of values that drive innovation.

- These values are mainly creativity, fertile imagination, “whole brain thinking”, teamwork, end-user focus, curiosity.

- The design thinking process should evolve through the following stages: define, research, ideate, prototype, choose, implement, and learn.
A bad software design may result in:

- A poor quality software product which does not meet user requirements
- A system that is not adaptive to any further requirement changes
- A system that is not reusable
- A system that has unpredictable behaviour
- A system that has poor performance
- Software production itself may be very ineffective in terms of time and cost
A good software design

- Reduces risks in software production
- Coordinates development teams to work together orderly
- Makes the system traceable for implementation and testing
- Allows the software product to have better overall system-qualities (quality attributes)
Where does software design sit in SDLC?

Software Development Life Cycle (SDLC)

1. Pre-requirements
3. Architecture Design
4. Detailed Design
5. Coding
6. Unit Testing
7. Integration & Testing
8. System Testing
9. Production Operation & Maintenance

Figure: Software Process V-model
Introduction to Software Architecture

What is meant by Software Architecture?

- Software architecture can be seen as a blueprint guideline for developing a large and complex software system based on the software requirements analysis specification.

- The architecture highlights early design decisions that have a decisive impact on the ultimate success of software products.

- Must be included in a high level design called "software architecture”
  - The specialisations of software elements
  - Connection types
  - Set of constraints (space, time, budget, etc.) on how they can be planned
  - Set of desired quality attributes (such as performance)
However, the proposed architecture can address partially or totally some of the desired quality attributes.

Families of architectures that have “similar” choices on element types and connections are very often called architecture style.

Each architecture style represents:
- The way the elements are arranged
- The connections and interactions among the elements
- The control transfer among the elements in the system
- The semantic constraints and behaviours concerning data transfer of the architecture style
- The quality attributes tradeoff
Introduction to Software Architecture  What is meant by Software Architecture?

- A partial list of quality attributes: performance, reliability, portability, usability, security, maintainability, adaptability, usability, etc.

- Quality attributes are closely related to architecture styles

- Each architecture style contributes some quality features

- An architecture style represents tradeoffs among many conflicting quality attributes
Introduction to Software Architecture

What is meant by Software Architecture?

Figure: Rough representation of a software system architecture
Introduction to Software Architecture  

What is meant by Software Architecture?

- IEEE Std 1471 defines the system architecture as "the fundamental organisation of a system embodied in its components, their relationships to each other, and to the environment, and the principle guiding its design and evolution"

- The basis of a software architecture design is the software requirement specification document

- The requirement process covers information domain modelling, data modelling, function modelling, behavioural modelling, and user interface modelling

- Software requirements has two aspects (functional, non-functional)
Introduction to Software Architecture

What is meant by Software Architecture?

Role of the software architect

- System static partitioning and decomposition into sub-systems and communications between sub-systems

- Establishing dynamic control relationships between different sub-systems in data flow controlling, or control flow orchestration, or message dispatching

- Considering and evaluating all alternative architecture styles for the problem domain

- Analysis of the tradeoff on quality attributes during the selection of architecture styles
Introduction to Software Architecture

What is meant by Software Architecture?

- Map a requirement specification to the software architecture and guarantee that the software architecture satisfies all functional and non-functional requirements
  - There are many kinds of traceabilities
  - The architect does backward traceability
  - Might get involved in the forward traceability
- If the requirements cannot be practically materialised, system analysts and software architects can go back to stakeholders in order to amend the requirements
Introduction to Software Architecture

Software Architecture Design Guidelines

- Think of what to do in abstract terms before think of how to do
- Think of non-functional requirements as early as possible
- Think of software reusability and extensibility as much as possible
- Try your best to promote the high cohesion within each element and loose-coupling between elements
- When thinking about design, ALWAYS keep in mind the design guiding principles
Overview

Where does software design sit in SDLC?

What is meant by Software Architecture?

Software Architecture Design Guidelines