

Object Oriented Analysis and Design

CS 6359.001: Reading Assignment #2

Due on Sunday September 15, 2019 at 11:59am

Instructor: Prof.Mehra Nouroz Borazjany



Shyam Patharla (sxp178231)

1. What is Software Architectural Design and why is it so important?

A software architectural design is a decision making process or a set of decisions which are taken to create the software architecture of the system to be created.

- A software architectural design for a system has long term and long reaching impacts on the system's:
 - adaptability to changes
 - performance and efficiency
 - security and maintenance
- It is needed for conceptualizing, managing and evolving the system while it is being created.
- software architectural design flaws may lead to system not being successfully created or system crash when it is operational.

2. What is the architectural design process?

The architectural design process for a system is a decision making process to make an architectural plan for a system. It involves steps such as

- formulate the **design objectives** i.e. the goals that must be achieved by the architectural design
- the **type** of system to be developed
- based on the above two steps, select an **existing** architectural style from the available set or make a **custom** architectural design
- specify the **functions, interfaces and interactions** of the system with other systems or subsystems
- **review** the architectural design and see whether the design objectives were met or not

3. What are the most common types of systems? Find an example for each type of the system.

The commonly observed types of systems are:

- Interactive systems (eg. a web browser which interacts with a user)
- Event driven systems (eg. a fire alarm system, which detects smoke)
- Transformational systems (eg. a compiler which transforms the source code to the desired output, or shows an error)
- Object persistence/ database systems (eg. a bank database)

- **What are the main characteristics of an interactive system?**

- interacts with an actor, which could be a human or another system
- must respond to all queries of the actor
- similar to a client-server relationship
- consists of a fixed sequence of requests and responses
- interaction begins and ends with the actor (usually a single actor in the process of a use case)

- **What are the main characteristics of an event-driven system?**

- system receives events from and controls external entities
- do not have a fixed sequence of requests and responses
- do not have to respond to all events
- often interact with more than one external entity at the same time, which is usually a system

- **What are the main characteristics of a transformational system?**

- consists of a network of information processing each of which transforms some input into some output
- is usually stateless
- while the transformation of input to output is being done, there is no interaction between actor and system
- may require computation-intensive algorithms
- actor may be human or another system

- **What are the main characteristics of a object persistence system?**

- used to store objects in a database and retrieve them later
- databases may be relational, hierarchial or object oriented
- uses a database manager to provide an object oriented interface to access the objects in the database

4. What is an architectural style?

An architectural style is a generic architectural design that may be used to create the architectural design for a system.

5. Why different types of systems require different design methods?

Because a design method which works well for one system may not work well for another system. This usually happens due to mismatch between the design methodology and the system which requires a certain software architecture.

6. Why do we need to perform custom architectural design?

When none of the available architectural styles are suited for the type of system we are building and are unable to fulfill the design objectives, a customized architectural design must be performed for this system.

7. What is a package diagram?

A package diagram is logical view of the classes of our system. During software development, a variety of software items are produced. A package diagram helps the understand which software items belong to which package. It helps us in managing the system software compo-

nents.

8. What are software design principles?

Software design principles are widely accepted guidelines in the software industry, which if implemented, can significantly improve the quality of our software. These are better approaches learnt by organisations in the software industry over a large period of time.

- **Design for change:** Design the architecture such that it can adapt to changes in the future.
- **Separation of concerns:** Focus on one aspect of the problem at a time rather than tackle all aspects simultaneously.
- **Information hiding:** Hide the implementation details of a software component so any changes to the component does not impact other components of the software system.
- **High cohesion:** achieving a higher degree of relevance of the functions of a module to the modules core functionality.
- **Low Coupling:** Reducing the run-time effect and change impact of a subsystem to other subsystems.

9. Which agile principles should be applied during architectural design?

- **Value working software over comprehensive documentation**
 - Over-documenting the architecture must be avoided.
 - Comprehensive documentation of architecture consumes a lot of time and resources and pushes the actual implementation back.
 - the design architecture generally evolves and is refined as iterations are implemented, hence wasting too much time on documentation of design architecture in the beginning is not good.
- **Apply the 20/80 rule i.e good enough is enough**
 - There is no optimal architecture for a system.
 - It is better to proceed with a good enough architecture in the beginning and improve it over many iterations.
 - Beyond a certain point, returns over improvement in architecture are negligible.

10. What is an architectural design and what is a design pattern?

Design patterns are generally proven good solutions to common design problems. Architectural design is the actual design for the software system which we create. Hence design patterns are used in the process of creating an architectural design for our system.

- **What is an MVC pattern?**

The MVC pattern is a design pattern. It consists of:

- a data model
- a controller, and
- a number of views

The controller separates the views from the data model, which allows the same data to be displayed in different views.