Chapter 5
Software Modeling and Design Process

Smrity Dwivedi
Indian Institute of Technology (BHU), Varanasi, India

ABSTRACT

Modeling is the designing of software applications before coding according to the object-oriented groups where under the model-based software design and development, software modeling and design are used as an important and mandatory part of the software development process. Software models are methods of expressing a software design. Basically, some sort of abstract language or pictures are used to express the software design under modeling. For object-oriented software, an object modeling language such as UML and many more are used to develop and express the software design. Also, the design model is based on the analysis and architectural essentials of the system. It denotes the application components and determines their appropriate placement and use within the overall architecture.

INTRODUCTION

Here, view of external system, their connection with system and their needs like what it provides and what its requirements are very important. Always before giving the design, quality has to be checked of the software, just because of requirements of software like security, performance and maintainability, which should have fulfilled before in use. Software design architecture is having high level as well as low level design, but high level is important to check the every point in well before as functional and non functional. It is all about to start the program. Also there are few notations to represent the figure and text for software design. The best way that is being used. UML and its notation to presents the design and modeling with different class and object as UML is object oriented software and used for many applications. Also these notations are used to perform a design but it is not used for develop the design. As software design architecture is a fundamental approach like security of information, that is why, design is perform with proper strategy, planning and way to direct towards the best results under overall design. Software strategy is like object oriented discrimination. Structuring is also important

DOI: 10.4018/978-1-6684-9809-5.ch005

while designing to provide the help to designer with proper steps to create the design as per applications. Structuring helps a designer and design team identify the design decisions. Under the design sequence and structuring, proper concepts, one and more than strategy, documentation are employed, in which collaborative object modeling and design, UML notation are used. Here, collaborative concepts are used for information hiding, classes, concurrent woks and inheritance, which address the active and passive objects and its interconnection. Also, under this structuring criteria during analysis of design and some additional criteria to get the information regarding subsystems and its tasks during the design. Each part given above will be discussed in this chapter with explanations and case analysis (Erl, 2006, 2008, 2009). By using modeling in software, it does not mean, to express any scientific theory or algorithm in software. But it is something that scientists traditionally call a software model and also software modeling is larger than an algorithm used or a single method. Software modeling should represent the entire software design including all interfaces, all interactions with other software being used, and all the software methods used.

Thinking of designing of software is like to design a house where rough sketch of the floor plan and layout of rooms and floors are decided first. Sketching is like modeling language and resulting view of overall thinking process is design. Modification is possible in the sketching to get final approved design to meet the requirements. Then start with, cutting and writing the code is the final touch. Also, in this design, any time, problems can be discovered and corrected with its code (Gomaa, 2011).

Now, the Software design is the process of defining software methods, functions, objects, and also the overall structure and interaction of code that is discussed above so that the results obtained will be satisfy to the users. According to requirements of users, code can be written. In the designing, there are so many methods to design the software with all initial design and modify it as necessary. Different software designers prefer different types of design while processing and during implementation phase (Friedenthal et al., 2009). Before, start the implementation, reviewing it again necessary. It is important and even easy to try out different designs first and also discover the problems initially in the development cycle to move on final design.

Software design should include all description of the overall architecture. This will cover the hardware, databases, and third-party frameworks so that designed software could interact with other software.

SOFTWARE DESIGN ARCHITECTURE

A software architecture separates the overall system structure from the underlying specifics of each individual component in a number of ways, including the components and how they are connected. According to Gamma et al. (1995), programming in the large (PITL) and programming in the small (PITS) are terms used to describe the main emphasis on components and their interconnections, respectively. It can provide a high-level description of the division of the software system into subsystems. According to H. Gomaa (2011), it can specify the breakdown of subsystems into modules or components at a lower level. In each instance, the interfaces, demands, and connections with other subsystems and components serve as the primary points of focus for the external picture of the subsystem or component. When creating the software architecture, it is important to take into account the system's software quality attributes. These characteristics show how crucial nonfunctional needs like performance, security, and maintainability are addressed by the architectural design. Another effective method for expressing design graphically, visually, or both is a software design architectural notation (Gomaa, 1994, 1986, 1989a, 1989b).