Sharif University of Technology

Department of Computer Engineering

Patterns in Software Engineering (40787)

Raman Ramsin

Graduate Course, 3 Units, Elective (Software: Software Engineering)
Prerequisites: None

Overview

The aim of this course is to familiarize graduate students with software patterns, their relevant concepts and principles, and their applications in software engineering. In addition to gaining knowledge and insight on common patterns of software systems analysis, design, architecture, reengineering, and development process engineering, students will also be introduced to antipatterns and refactoring techniques. Widely-used patterns will be described in detail; however, due to the large number of the patterns introduced, the main focus is on introducing the basic pattern structures and their underlying principles and philosophies. The course thus strives to provide a basis for pattern analysis and pattern-based software engineering.

Topics

1) Introduction to the History and Basic Concepts of Software Patterns (1 session – each session is 90 minutes in duration)
2) Forerunners: Coad Patterns (1 session)
3) GoF Design Patterns (5 sessions)
4) Object-Oriented Rules and Principles as Patterns – GRASP (2 sessions)
5) GoV Architectural Patterns (4 sessions)
6) GoV Design Patterns (2 sessions)
7) Refactoring Patterns (4 sessions)
8) Reengineering Patterns (4 sessions)
9) Software Process Patterns (1 session)
10) Antipatterns (3 sessions)
11) Analysis Patterns (2 sessions)
12) Methods of Pattern Classification, Complexity Management, and Analysis (1 session)

Assessment

- Two exams (Midterm and Final) – Comprising 60% of the total grade
- Three study assignments – Comprising 20% of the total grade
- Research Project: Based on a topic of the student's choice, the research project is conducted under the guidance and supervision of the instructor – Comprising 20% of the total grade

Main References

- E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley, 1995.
- F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture: A System of Patterns,* Vol. 1. Wiley, 1996.
- M. Fowler, *Analysis Patterns: Reusable Object Models*. Addison-Wesley, 1996.

- J. Kerievsky, *Refactoring to Patterns*. Addison-Wesley, 2004.
- A. Shalloway and J. Trott, *Design Patterns Explained: A New Perspective on Object-Oriented Design*, 2nd ed. Addison-Wesley, 2005.
- D. Manolescu, M. Voelter, and J. Noble, *Pattern Languages of Program Design*, Vol. 5. Addison-Wesley, 2006.
- F. Buschmann, K. Henney, and D.C. Schmidt, *Pattern-Oriented Software Architecture: On Patterns and Pattern Languages*, Vol. 5. Wiley, 2007.
- C.J. Neill, P.A. Laplante, and J.F. DeFranco, *Antipatterns: Managing Software Organizations and People*, 2nd ed. CRC Press, 2012.
- M. Fowler, *Refactoring: Improving the Design of Existing Code*, 2nd ed. Addison-Wesley, 2019.