Project List – Autumn 2015

SE Lab, IIITB

1 Advanced Methods of White-Box Testing

Mode: Research Elective

Keywords: Program analysis, software testing, model checking, formal methods

Description: State-of-the-art test generation methods of today invariably use tools and techniques based on formal methods, e.g. SAT-solving, model-checking and program analysis, etc. In this project, we implement our own test generation tools involving sophisticated core algorithms.

2 Test case generation from architecture models

Mode: Project Elective

Keywords: Embedded software testing, embedded software architecture, non-functional testing

Description: AADL is a standard language for specifying architectures of embedded software and is used as a basis for ensuring non-functional requirements of software are met. This project involves generating test cases for verifying non-functional requirements of embedded software using their architecture specified in AADL.

3 Testing of IoT (Internet of Things)

Mode: Research Elective

Keywords: Internet of Things, Software Testing, interoperability testing.

Description: Internet of Things (IoT) is an emerging computing paradigm with tremendous potential for growth. This project involves testing of applications based on IoT. Specifically, we would like to explore a few aspects related to testing of IoT applications. In particular, this project will involve understanding testing related to functionality and interoperability aspects of IoT applications. Work will involve literature survey and developing prototype algorithms for testing in the above mentioned areas of IoT.

4 Advanced Testing of Mobile Apps

Mode: Research Elective

Keywords: Software testing, GUI, mobile computing, machine learning, model based testing

Description: Graphical user interfaces (GUI) often accounts for a major part of the system code and complexity in many software systems. Modern day GUIs are very large systems. Designing mathematical models which precisely capture GUI semantics and are succinct enough to scale are difficult to build. This project explores the problem of mathematical modelling of GUI with the primary motive of specification, testing and verification. We also develop a tool that implements sophisticated testing algorithms using the above models.

5 Evaluation of Test Generation Algorithms

Mode: Research Elective

Keywords: Software testing, program analysis, algorithms, model based testing, embedded software, Matlab Simulink

Description: Test automation tools often generates a test coverage report to give an estimate of the effectiveness of testing. Often, these coverage reports are made the basis of choice of testing tools. However, these coverage reports may not be dependable basis for evaluating testing tools or algorithms due to a phenomenon called *pseudo coverage*. In this project, we develop and implement an approach to detect pseudo coverage.

6 Web-based Book Request Processing System

Mode: Project Elective

Keywords: Web-based systems, Java, Python, Javascript, web-services

Description: In this project, we implement a web-based system that:

1. allows institute members to order books for the library and track the status of their request
2. allows administrator to view/approve/reject a request.

7 Automated Training Program Design for a Large Organisation

Mode: Research Elective

Area: Algorithms, optimisation

Description: Large organisations have a complex and dynamic scene when it comes to training their employees for new skills. While training is central to develop a skilled work-force, they also cost a lot to organise. Hence, there is a need to optimally define training modules to meet the organisation's skill requirements with less expenses.

8 A Generic Queueing System for Service Organisations

Mode: Project Elective

Keywords: Queuing theory, algorithms, software architecture, Java, OOAD

Description: Organisations like banks, hospitals, consulates etc. are basically queuing systems. In presence of a variety of services, sporadic variations in load, a central queueing algorithm that has a view of the entire system and the intelligence to assign jobs to resources to maximise the quality of service would be a significant component in the efficient running of such system. Another aspect of this system is that it must be architected to support a flexible operation where algorithms, revenue models, system components (e.g. network protocol, client systems etc.) can be dynamically plugged in and out. This project implements this core queueing infrastructure.