ABSTRACT

Software testing is performed to determine whether or not a system satisfies the requirements of customer. Due to the increasing size of software systems, manual testing becomes untenable. Hence, there is a need to automate techniques for software testing. Specification based testing intends to confirm implementation of specifications. This work is concerned with user requirements collected during requirement engineering of an intended system.

The Object Management Group defines the Unified Modelling Language (UML) as a general-purpose visual modeling language that is used to specify, visualize, construct and document artifacts of a software system. UML captures information about the static structure as well as dynamic behaviour of a system. The static structure defines objects as well as the relationship between objects that are part of the system implementation usually represented in use case, class and component diagram. Dynamic behaviour of the system is specified by the activity, sequence and state diagram.

The semi-formal nature of UML has both advantages and disadvantages: the advantage primarily lies in its ease of use as well as understandability by various stakeholders of the system. Also, different diagrams can be used to model varying aspects of the system. The same leads to difficulties in the form of maintaining completeness and consistency within and between UML diagrams. Specification based testing using UML needs consistent and complete UML diagrams. Again for testing, scenarios representing working of intended system are extracted and studied for the purpose. The study includes generation, prioritization and selection of scenarios from UML specification.

Diagram consistency checking follows a horizontal transformation approach and well formed rules are applied. Particularly, diagrams (through their artifacts) are stored in RDBMS and consistency checking rules are formulated as triggers, assertions, etc to alert user incase inconsistency is encountered.

Each use case representing functional system requirement is illustrated by ac-
tivity diagrams; and scenarios are extracted by traversing these diagrams. Domain specific relations among activities are considered to reduce the number of scenarios that may turn large due to multiplicative factor contributed due to concurrent activities. A tool is developed to support scenario extraction process.

Use cases and scenarios are prioritized to obtain an ordering for execution of test cases. Use case priority is obtained both from the customer as well as computed from structural complexity of a use case diagram. Also, the structural primitives of the activity diagrams are used to obtain the priority of a scenario. The priority of a scenario is computed by a weighted sum of both use case priority and scenario priority.

An effective ordering of test scenarios for execution helps in early detection of defects. However, given constraints of cost and time, it may not be possible to execute the test suite completely. Instead, a subset of test scenarios may be selected for testing. The fourth contribution is selection techniques to help select a subset of scenarios based on the similarity that exists between scenarios. The first technique uses Levenshtein distance as a measure to calculate dissimilarity between scenarios. A second technique looks at finding common subscenarios, their length and relative position to calculate similarity. A third technique uses Agglomerative Hierarchical Clustering(AHC) to cluster similar scenarios for selection of a representative scenario.

Given a large number of use cases and scenarios for even a medium sized system, there is a need for effective management of use cases and scenarios to facilitate testing. The thesis proposes an ontology to aid scenario management. Scenario management is demonstrated using Protege, an open source ontology editor. A tool also has been developed for scenario generation and prioritization. The usability of the proposed concepts are demonstrated through two case studies and results are analyzed to study the impact of different kinds of prioritization and selection techniques.