Chapter III

Object Oriented Testing in Software Development

Samuel K. A. Agyemang
IBM Global Services

Object oriented technology is still growing and has not yet matured. Many articles have been written on object oriented software development processes, particularly in the area of testing. Most of the publications seem to agree with the fact that object oriented testing is a challenging aspect of the software development process. The main reason for this view seems to revolve around the fact that the objects and the code are inseparable and also, the idea of inheritance. Despite these views, the publications all seem to agree on one aspect, that object oriented when successfully tested leaves a better-maintained product compared to the traditional non-object oriented software. Object oriented software makes better maintainable software; it also has an added advantage over traditional software development because in the final analysis, it will cost less by shortening the development time as well as cutting down the cost of maintenance.

Object Oriented Testing

Object oriented software involves the methods to organize both information and the process that manipulates that information in accordance with the real world objects that the information describes.

The objects have attributes whose values define the state of the object and these also determine the value of the objects. (Kung, 1995) Object oriented software could be described to be just about objects. The concept behind object orientation is that it is closer to a person’s perception of reality. Object orientation gives a new and powerful formula for developing computer software which makes it unique from the traditional form of software development. In a world where new products are emerging all the time, particularly in this Internet age, object orientation could help to improve the reliability and speed with which new software is developed. There could also be an added advantage of maintaining such software which is one of the problems in software development. There has already been a lot of interest generated about object oriented technology and the advantages it brings to the software development process which may be lacking in the traditional process. The powerful features of object orientation which are encapsulation, re-usability and inheritance also introduce problems for testing and maintenance, the anchor of a good software product. Most developers have tried to circumvent this problem by advocating the use of conventional testing tools for object oriented software. (Kung, 1995).

Object oriented development process provides some powerful and interesting features for software development (Hsia et al., 1998). It also brings to light some problems in the areas of quality of the software, particularly in the areas of testing and maintenance.

This will be the focus of this chapter, looking at some of the testing techniques and the problems of testing in determining if object oriented technology has any advantages over the traditional process. The chapter will look at aspects of testing in the object oriented technology, the issues involved and how they make object a quality product and the problems that may be encountered. Due to the growing importance of this technology, there has been a lot of testing strategies each trying to explain the advantages of object orientation. In this chapter, a critical view of some of the articles on the topic will be discussed to show the advantages object orientation has over traditional process. By the use of encapsulation, the designer could narrow the possible interdependencies with other components by way of interface without affecting the other units (Perry, 1990). Even though object oriented programming has some advantages over traditional, particularly by the use of the inheritance and encapsulation techniques, it also raises
User Requirements Validation and Architecture Discovery through Use Case Invariants and Model Animation
[www.igi-global.com/chapter/user-requirements-validation-architecture-discovery/25635?camid=4v1a](www.igi-global.com/chapter/user-requirements-validation-architecture-discovery/25635?camid=4v1a)

Object-Oriented Modeling in UML2
[www.igi-global.com/chapter/object-oriented-modeling-uml2/25630?camid=4v1a](www.igi-global.com/chapter/object-oriented-modeling-uml2/25630?camid=4v1a)