ABSTRACT

Active Object Diagrams as a Tool to

Support Object Oriented Programming Comprehension

Jason Annil Thompson

Object-oriented programming is a key part of the computer science curriculum because of its wide-scale use in the software development industry and because of the benefits the paradigm possesses. However, students generally have difficulty comprehending several of its concepts. This has lead to much research to identify students areas of difficulties and to determine ways to improve students' comprehension of its concepts. We developed Active Object Diagrams (AODs), which depict an object's state, its behaviour, and the relationship of an object to other objects in a system. An extension of UML object diagrams, AODs incorporate the object behaviour at a point in time and the object reference. The effectiveness of AODs as an instructional tool was investigated by testing if there was an improvement in students' performance after they were taught with AODs. However, the concepts that students found difficult and a viable instructional approach to be used with AODs had to be identified before the study was done. Both the difficulties and instructional approach were identified by examining research literature, observing and surveying first year OOP students and interviewing OOP instructors.

The results of the investigation showed that the use of AODs in instruction resulted in a marginally significant ($p<0.055$) improvement in the students' overall performance, when tested on several OOP concepts. However, there was a significant improvement in the students' performance on an object communication question ($p<0.026$). In addition, the Object Visualization System for Novices (OVSN), a prototype programming visualisation system that utilizes the AODs technique. It provides a concrete representation of objects in the form of AODs that are controlled by the executing program so that its users can observe the changes in object state during program execution.

Keywords: object-oriented programming, object diagrams, program visualization