Implementation of CONCEIVER++ : an object-oriented program understanding system.

ABSTRACT

Problem statement: Understanding on computer program is a complex cognitive activity. It is ability and also a difficult task especially for novice programmer. The object-oriented languages has widely used in education and industry recently. In programming it is important to have such software which can aid programmers or students to code the program. But, available program understanding systems using the plan based approach usually are developed for non-object-oriented programming languages. Reviewed from the available system also showed that none of the plan formalisms used is for an object-oriented language. Specifically, problem arises when the existing system is not usable for teaching programming purposes. Program understanding system with plan for object-oriented does not exist was the main reason why this research is being carried out. Approach: Method used on developed the program understanding system named CONCEIVER++ is Unified Approach (UA). The process involved from UA for developing and testing the system is iterative development and continuous testing. The process must be iterate and reiterate until satisfied with the system. In order to test the quality assurance of the system is by choosing the black box testing strategies. Results: The object-oriented program understanding system has been successfully implemented. The implementation is tested with an example of Java programming code. The binary search tree for control flow graph and linked list for plan has been generated. Results of understanding the meaning or semantic of the program codes also has been produced. The black box testing had shows that all statements of line of code of the example program have been recognized and the correctness output has been checked. Conclusion: The understanding module of CONCEIVER++, which are code/CFG processor, plan processor and recognition engine has been tested. All line of codes (or nodes) has been recognized and got correct meaning using the developed module.