

ABSTRACT

A Software Tool for the Reuse of Object-Oriented Classes

Colin Anthony Depradine

One of the most alluring promises of object-oriented programming is the ability to perform code reuse in a simple and efficient manner. Currently, many programmers continue to depend on the syntax of an object-oriented language to perform code reuse. For a small number of objects, this form of code reuse is easy to control. But as the number of objects increases, the managing and searching of information becomes increasingly difficult and unwieldy. To support the storage, querying and viewing of software reuse information, the collection-view model is introduced as an architectural pattern for the design of code reuse systems and is implemented as the ROOC system for the reuse of object-oriented classes. The ROOC system is aimed at those programmers that work alone or in small teams of five or less. This group of developers makes up a sizeable proportion of the software development community and therefore, cannot be ignored. It allows a programmer during the implementation phase of the software life cycle to readily update and retrieve class reuse information. The class information is standardized in the D-ROOC format and accessed via the S-ROOC scripting language. The S-ROOC scripts are executed via an object-oriented search engine whose structure is based on design patterns. Its object-oriented nature reduces the maintenance effort and facilitates the reuse of search engine code in several other ROOC utilities. Finally, the ROOC system is language and platform independent, allowing the developer to capture reuse information for a variety of object-oriented languages. Although the system was originally designed for small development teams in the Caribbean region, it is nonetheless hoped that the ROOC system will make class reuse an integral part of software development effort in any environment.

Keywords: Software reuse; Object-oriented; Classes; Design patterns.