Using Case-Based Reasoning and Genetic Algorithm in Framework Documentation Approach

HAJAR MAT JANI
College of Information Technology
Universiti Tenaga Nasional
Km. 7, Jalan Kajang-Puchong
43009 Kajang, Selangor Darul Ehsan,
MALAYSIA
hajar@uniten.edu.my

LEE SAI PECK
Department of Software Engineering
Faculty of Computer Science and Information Technology
University Malaya
50603 Kuala Lumpur,
MALAYSIA
saipeck@um.edu.my

Abstract - *Framework developers normally use several approaches in documenting object-oriented application frameworks, which include tutorials, reference manuals, design patterns, cookbooks, and minimalist documentation. Basically, the main objective of a framework is to significantly reduce the time and effort needed in developing application software within a family of applications domain. This paper implements the case-based reasoning (CBR) and genetic algorithm (GA) techniques in documenting a framework. The main objective is to introduce an effective framework documentation approach using CBR and GA that makes learning to use a specific object-oriented application framework less troublesome to new framework users. In CBR, reasoning is based on remembering and recalling cases from past experience. Here, a case is a complete example of how to use a particular component or set of components within a given framework. In other words, a CBR system learns through examples. GA is used in implementing the CBR's "retrieve", "reuse" and "revise" processes. During the "revise" and "retain" processes of the CBR, a pattern matching algorithm is applied to ensure that correct adaptation is performed in getting solutions to new framework usage cases. To implement the proposed documentation approach, a prototype that uses this approach is developed in Java. The evaluation of the prototype shows that the proposed approach is able to reuse existing cases, adapt past cases to suit new cases, and learn new cases correctly, and effectively.*

Keywords: Framework documentation, case-based reasoning, genetic algorithm.

1 Introduction

Object-oriented application framework presents one of the most successful approaches to developing reusable software components and libraries of software by providing a general skeleton of classes and behavior patterns for a given application domain [1]. An application framework may also be considered as a generic solution for a class of problems. There must be points of flexibility within the framework that can be customized to suit the required application.

To generate an executable application using a framework, several components of the framework must be instantiated by implementing application specific code for each feature that is changeable. Some features of the framework are stable and cannot be easily altered. Effective and complete framework documentation is required in order to instantiate the framework correctly in meeting the new application's requirements.

Basically, there are two main objectives of framework documentation. First, information about framework design and other related information must be made available and communicated during the framework development. And, second, information on how to use the framework must be conveniently available to all users and need to be delivered along with the framework [2].

Based on our previous study [3], the following documentation approaches have been identified: tutorials, cookbooks and recipes, design patterns, frequently asked questions, contracts, run-time errors documents, web approach, formal specification, hooks model, and case-based reasoning.

The results of the survey and interviews performed in [4] showed that most framework users are still not satisfied with the *effectiveness* and *usability* of the documentation that is provided along with the framework. About 50% of them took more than 1 year to master the features of a specific framework.

2 Objective and Motivation

The main purpose of using a framework is to minimize the amount of time required for developing applications