Chapter 59
Analysis and Evaluation of Software Artifact Reuse Environments

Sajjad Mahmood
King Fahd University of Petroleum and Minerals, Saudi Arabia

Moataz Ahmed
King Fahd University of Petroleum and Minerals, Saudi Arabia

Mohammad Alshayeb
King Fahd University of Petroleum and Minerals, Saudi Arabia

ABSTRACT

Software reuse enables the development of software that is of better quality and at lower cost. Software reuse environments are sought to enhance the reuse of software artifacts especially when done at early-stage of the software life cycle. A number of software reuse environments have been proposed, however, there is no framework that helps in analyzing and evaluating such environments. In this paper the authors provide an attribute-based framework to analyze, evaluate, classify and compare the reuse environments in order to aid practitioners and researchers to select the appropriate reuse environments for their use. The authors first present a survey of existing reuse environments for systematic reuse of software artifacts. Then, they use the framework to analyze those reuse environments. The evaluation of existing environments provides an understanding of current reuse approaches and identifies gaps for future research.

INTRODUCTION

Software reuse is the process of creating software using existing software rather than building software from scratch. Software reuse tends to reduce overall development costs, save significant development effort, improve product quality, reduce process risks, and make development faster. Reuse can take place at the architecture, design and source code levels. Object-oriented (OO) features such as abstraction and encapsulation encourage reuse of software by enabling building reusable blocks of code. However, software reuse is more beneficial if types of software artifact other than code such as domain models, requirement specification, design, documentation, and test

DOI: 10.4018/978-1-4666-8200-9.ch059

data can be reused as well. If an analysis model for a previous project is found to be similar to the analysis model of a current project, then the previous project's artifacts: design, code, test data, and relevant documentation may be reused in the current project.

Software reuse effort suffers some few challenges though (Blok & Cybulski 1998, Mili, Mili et al. 1998, Li, Zhang et al. 2008, Hummel 2010). These challenges include increased maintenance costs, lack of tool support, difficulty of maintaining a library of reusable artifacts, and the cost of locating and adapting reusable artifacts.

Researchers investigated the development of effective integrated reuse environment (IRE) (Mascena, Meira et al. 2010, Ahmed 2011) to address the challenges of software reuse. IRE maximizes the designer's productivity by offering tightly-knit tools integrated with some prominent CASE tools to allow finding and reusing exiting design artifacts and what follows based on matching requirements specification. IREs are usually capable of maintaining and indexing repositories of previous project diagrams, finding similar artifacts for reuse, ranking and comparing potential similar existing models for best fit and assessing the impact of the difference on effort.

To the best of our knowledge, there is no framework that helps in analyzing and evaluating such environments. In this paper we present an attribute-based framework to analyze, evaluate, classify and compare the reuse environments in order to aid practitioners and researchers to select the appropriate reuse environments for their use. Furthermore, the framework can also guide researchers interested in proposing new reuse environments. The paper discusses nine representative reuse environments against the framework. We also present our initial effort towards the development of architecture of an integrated reuse environment to allow early-stage object oriented software artifact reuse. A preliminary version of this paper can be found in (Mahmood et al. 2013).

The rest of this paper is organized as follow: Section II presents the analysis framework. Section III reviews the software reuse environments. Section IV discusses open research challenges with reference to software reuse environments. Section V presents an initial architecture of a sought integrated reuse environment. We present the related work in Section VI. We finally conclude the paper and discuss the future work in Section VII.

ANALYSIS FRAMEWORK

We studied and contrasted environments available in the literature. Based on our observations on these environments along with some wish items, we were able to identify a set of attributes to analyze different software reuse environment and approaches. The attributes are as follows:

- **Artifact Type (AT)**: To better understand a reuse approach, it is important to specify the supported software artifacts (for example, UML class diagram, UML sequence diagram, feature models etc.).
- **Reuse Phase (RP)**: In addition to artifact type, it is also important to specify what stage (i.e. early-stage or later-stage) these artifacts are developed during a software development life cycle.
- **Artifact Documentation (AD)**: How software artifacts are stored and indexed for reuse retrieval. Furthermore, are there any specific meta-models supported by the reuse approach to facilitate assessment between different artifacts belong to a range of projects within a domain?
- **Similarity Assessment (SA)**: For effective reuse of available software artifacts, how a reuse environment carries out a similarity matching between different software artifacts. Does it support specific model-