Interdisciplinary collaboration between engineering, mathematics and science

SEMS Research Highlights

Knowledge Management Framework for Software Reuse

Dr. Jameela Al-Jaroodi*, Mr. Simone Maccanti**, Dr. Arif Sirinterlikci***

***Associate Professor of Software Engineering, **Graduate Student,
***University Professor and Head of Engineering Department**

This newsletter presents the research conducted within the School of Engineering, Mathematics and Science (SEMS) at Robert Morris University (RMU). It covers various relevant topics including: interdisciplinary efforts, successful research grants, student research, posters and papers, journal publications, presentations at national and international conferences, contribution to professional societies, STEM educational research, industrial consulting collaborations and applied research.

Adjoining figure shows a new project 'B' being developed. Some software components for Project 'B' may come from previously developed software for Project 'A' along with some new requirements specific to Project 'B'. The existing and new software components then need to be merged to complete Project 'B'.
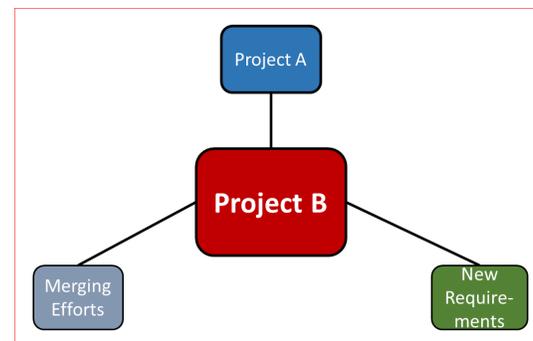
When considering software development projects, it is easy to see when there are parts or features in the requested software that have been developed before and consider reusing these in the new project. However, finding the right information and components from previous projects for reuse is a very difficult task. Software reuse is an important and effective way to help reduce the cost and duration of the software development projects. However, it is underutilized due mainly to the lack of the necessary knowledge of previous work. Locating previous work, features and components sometimes takes longer than actually building these parts from scratch.

In this work, we propose to build a knowledgebase that can transform, store and locate all necessary information about previous projects; allow software developers to shape and organize this information in a way that is useful to them; create learning mechanisms such that the knowledgebase will be continuously evolving and refining its capabilities to better respond to the software developers' queries.

The framework relies on an Extract, Transform, and Load (ETL) model that goes through multiple stages to first extract the information from current sources such as code, design diagrams, documentation, flow charts, etc. Then it uses specific rules and patterns (learning mechanism) to transform this information into the format used within the knowledgebase. Finally, it will load the created knowledge onto the knowledgebase along with all the necessary information that will help others use this information effectively. The learning mechanism is the central part of this knowledgebase and it uses four parts: the rules, patterns, applications and user formats.

Successful implementation of the knowledgebase will create a central repository of all previous and current projects, where software developers can query and find the necessary information that could lead them to successfully reuse existing software in new projects.