A design level optimization approach for functional paradigm software designs considering low resource devices development

Selvakumar Samuel, A. Kovalan

Affiliations
Periyar Maniammai University, Vallam, Thanjavur - 613403, Tamil Nadu, India

DOI: 10.17485/ijst/2016/v9i21/95208

Abstract

Objectives: The main objective of this paper is to identify suitable programming concepts from Functional Programming paradigm concerning low resource devices development and eventually contribute an approach for design level optimization. Methods: Experiments have been conducted, CPU time and memory consumptions (Private Bytes) were measured. Findings: The research results indicated that Pattern Matching, Lazy, Curried, Tail Recursion, Functional Composition, Referential Transparency and Higher Order Functions with functions as parameters concepts consumed less CPU and memory resources compared to their alternative concepts. This paper suggests that the above mentioned concepts can be applied by any software engineering practitioners in designing resource efficient constructs for software applications. Applications: Using these guidelines substantial performance growth can be formed and at the same time, performance degradation issues can be easily avoided. Eventually, this paper contributes a way to optimize the Functional Programming design at design level.

Keywords

Full Text:

References

predictability across the software life cycle. A software factory schema is a graph of viewpoints used to separate concerns, relating work
done at one level of abstraction, in one part of a system, or in one phase of the life cycle, to work done at other levels, or in other parts and
phases, and about how the schema can be used to deliver guidance and to support its enactment. A Disciplined Agile Delivery Handbook
for Optimizing Your Way of Working, is an indispensable guide for agile coaches and practitioners to identify what techniques - including

DOI: http://dx.doi.org/10.17485/ijst%2F2016%2Fv9i21%2F95208

Refbacks
There are currently no refbacks.
practices, strategies, and lifecycles - are effective in certain situations and not as effective in others. One of such paradigms is called a Component Based Software Development (CBSD) and it relies on the concept of building an application components that are meant to be the independent, reusable pieces of code. In this post I will present the component based approach for large-scale software development, discuss its advantages and argue its superiority over traditional approaches in modern large-scale software. What is Component-Based Software Engineering? Component Based Software Engineering aims at reducing the cost of software production and improving the quality of a system by building it using... Request PDF on ResearchGate | A Design Level Optimization Approach for Functional Paradigm Software Designs Considering Low Resource Devices Development | Objectives: The main objective of this paper is to identify suitable programming concepts from Functional Programming paradigm concerning low resource devices development and eventually contribute an approach for design level optimization. Methods: Experiments have been... This paper suggests that the above mentioned concepts can be applied by any software engineering practitioners in designing resource efficient constructs for software applications.