ABSTRACT

This paper evaluates the features of mainstream programming paradigms. Imperative, object oriented programming and functional programming concepts are considered here. This is an effort to identify the programming paradigms which consume less resource from mobile devices. Designers usually depend on the programming languages, language oriented programming design is in current practice. Choosing appropriate programming paradigms during the mobile application design is not in practice now; Failure to use the best approaches for mobile computing from programming paradigms will cause mobile applications to consume more mobile resources. Imperative paradigm concepts such as inheritance, creating redundant objects, unnecessary constructors, recursion, strings concatenation, thread synchronization, using global variables and abstract methods results in redundancy, memory leaks, stack overflow, low execution speed and consumes more memory. These features are relatively not suitable for mobile software development. Functional paradigm concepts such as higher order functions, tail recursion, lazy evaluation, referential transparency, parametric polymorphism, and list comprehension principles are suitable for mobile software development as they consume less memory and or use less processing power. Using appropriate paradigms will optimize the resource utilisation of mobile applications in mobile devices.

Keywords: Functional Paradigm, Green Computing, Mobile Application Development, Mobile Computing, Programming Paradigms

DOI: 10.4018/ijide.2014010105
1. INTRODUCTION

Presently, mobile devices are able to do things that are normally done with computers, including desktops and laptops except for software development. The advantage of mobile devices is that it consumes much less power compared to computers, hence if used for common computing need; it will save energy and contribute towards keeping our planet green.

In programming, there are many paradigms that have been suggested and this keeps evolving, in order to suit the requirements of software development of the respective times. A paradigm defines a fundamental approach to design a solution for a given problem. Choosing an optimal paradigm will enhance the accuracy and performance of the solution. Mostly procedural, structural and object oriented features are common when we use languages like Java, C#, C++, Objective C etc. for mobile software development. Inappropriate use of paradigms consuming more resources while running the mobile applications in mobile devices.

“Optimization is the process of making your application run more effectively. You can optimize for many things like speed, memory space usage, disk space usage, etc.” (Bartlett, & Bruno Jr., 2004).

The optimization of a program design is highly dependent on the programming paradigm. As optimization can occur at a number of levels, the system can make the best use of the available resources through design level optimization.

The performance of a system is greatly affected by its architectural design. Therefore, among all other items of design, the choice of paradigms affects efficiency the most (Bentley, 1982).

The simple JavaMe code shown in Example 1 demonstrates the role and dominancy of programming paradigms in a program design. The output of the 12- lines code is just to display a message “Hello mobile world” as shown in Figure 1. But the usage of programming paradigms is vast, it has most of the object oriented and other programming concepts such as class, objects, encapsulation, packages, inheritance, abstract methods, static, constructors, etc., while it executes it will take more mobile resources which is not worthwhile for the required output. This is the main flaw of the current approaches.

Mobile devices have limited resources compared to desktop machines; hence the application of the desktop development ap-

```java
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class Example 1 extends MIDlet {
    private Form form;
    public Example 1() {
        form = new Form("Example 1");
        form.append(new StringItem (null, "Hello Mobile World");
    }
    public void startApp() {
        Display.getDisplay (this).setCurrent (form);
    }
    public void pauseApp() {}
    public void destroyApp (Boolean unconditional) {
        notify Destroyed();
    }
}
```

"Example 1."
Related Content

Biometrics Technology and the New Economy: A Review of the Field and the Case of the United Arab Emirates
[www.irma-international.org/article/biometrics-technology-new-economy/74063/](www.irma-international.org/article/biometrics-technology-new-economy/74063/)

Future Directions
[www.irma-international.org/chapter/future-directions/23571/](www.irma-international.org/chapter/future-directions/23571/)

Attributes for Executing Change in an Agile Information System
[www.irma-international.org/article/attributes-for-executing-change-in-an-agile-information-system/130677/](www.irma-international.org/article/attributes-for-executing-change-in-an-agile-information-system/130677/)

Measuring Lecturers’ Perception of Transition to E-Learning Systems and Digital Divide: A Case Study in School of Business Administration of Istanbul University
[www.irma-international.org/chapter/measuring-lecturers-perception-transition-learning/72410/](www.irma-international.org/chapter/measuring-lecturers-perception-transition-learning/72410/)
Web Service Composition, Optimization and the Implications for Developing Economies

Leveraging Developing Economies with the Use of Information Technology: Trends and Tools (pp. 57-78).

www.irma-international.org/chapter/web-service-composition-optimization-implications/65927/