

## Chapter 13

# Productivity Evaluation of Self-Adaptive Software Model Driven Architecture

**Basel Magableh**

*Trinity College Dublin, Ireland*

**Stephen Barrett**

*Trinity College Dublin, Ireland*

### **ABSTRACT**

*Anticipating context changes using a model-based approach requires a formal procedure for analysing and modelling context-dependent functionality and stable description of the architecture which supports dynamic decision-making and architecture evolution. This article demonstrates the capabilities of the context-oriented component-based application model-driven architecture (COCA-MDA) to support the development of self-adaptive applications; the authors describe a state-of-the-art case study and evaluate the development effort involved in adopting the COCA-MDA in constructing the application. An intensive analysis of the application requirements simplified the process of modelling the application's behavioural model; therefore, instead of modelling several variation models, the developers modelled an extra-functionality model. COCA-MDA reduces the development effort because it maintains a clear separation of concerns and employs a decomposition mechanism to produce a context-oriented component model which decouples the applications' core functionality from the context-dependent functionality. Estimating the MDA approach's productivity can help the software developers select the best MDA-based methodology from the available solutions. Thus, counting the source line of code is not adequate for evaluating the development effort of the MDA-based methodology. Quantifying the maintenance adjustment factor of the new, adapted, and reused code is a better estimate of the development effort of the MDA approaches.*

DOI: 10.4018/978-1-4666-2157-2.ch013

## **INTRODUCTION**

Mobile computing environments are heterogeneous and dynamic. Everything from the devices used and resources available to network bandwidths and user context can change drastically at runtime (Belaramani, Wang, & Lau, 2003). This presents the software developers with the challenge of tailoring behavioural variations both to each specific user need and to the context information. Context-dependent behavioural variations can be seen as a collaboration of individual features expressed in requirements, design, and implementation. Before encapsulating the crosscutting context-dependent behaviours into a software module, the developers must first identify them both in the requirements document and in the software model. This is difficult to achieve because, by their nature, context-dependent behaviours are entangled with other behaviours, and are likely to be included in multiple parts (scattered) of the software modules. Using intuition or even domain knowledge is not necessarily sufficient for identifying the behavioural variations; instead, it requires a formal analysis procedure for the software requirements and a separation of their individual concerns. Moreover, a formal procedure for modelling these variations is needed. This kind of analysis and modelling procedure can reduce the complexity in modelling self-adaptive applications and encapsulate the context-dependent part of the distinct architecture module (component).

In this sense, a context oriented component model (COCA-component) (Magableh & Barrett, 2009) is used to encapsulate behavioural variations and decouple them from the application's core functionality. In this way, dynamic component composition is achieved. Additionally, from the software developer's perspective, it is vital to know the productivity of the development paradigm which might be used in constructing the self-adaptive application. Productivity evaluation of model-driven approaches can assist the developers in selecting among the proposed methodologies in

the literature which approach dynamic behavioural variations of self-adaptive software. Context-oriented component-based application-model-driven architecture (COCA-MDA) emerged as a development paradigm which facilitates the development of self-adaptive context-oriented software (Magableh & Barrett, 2011b, 2011c).

This article evaluates the development effort involved in adopting COCA-MDA when constructing a self-adaptive application for an indoor wayfinding application (IWayfinder) targeting individuals with cognitive impairments. The development effort of COCA-MDA is compared to other model-driven approaches proposed in the literature.

The remainder of the article is structured as follows. The next section provides a comparative analysis of related studies. A case study of a self-adaptive application is then demonstrated. The COCA-MDA phases are also described. The following section provides a COCA-MDA evaluation using constructive cost model II (COCOMO II). The last section summarizes the research findings and describes directions for future work.

## **RELATED WORK**

In the literature, there are several MDA approaches which target the development of self-adaptive applications for mobile computing environments which produce component-based applications; this study borrows from the following methodologies: MUSIC, proposed by Wagner, Reichle, Khan, and Geihs (2011); U-MUSIC (Khan, 2010); and Paspallis MDA (Paspallis, 2009).

The MUSIC development methodology (Wagner et al., 2011) adopts a model-driven approach to constructing the application variability model. The applications are built as a component framework with component types as variation points. Middleware is used to resolve the variation points, which involves the election of a concrete component as a realization of the component type. Using

18 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/productivity-evaluation-self-adaptive-software/72762](http://www.igi-global.com/chapter/productivity-evaluation-self-adaptive-software/72762)

## Related Content

---

### Experiences with Software Architecture Analysis of Usability

Eelke Folmerand Jan Bosch (2010). *Web Engineering Advancements and Trends: Building New Dimensions of Information Technology* (pp. 177-202).

[www.irma-international.org/chapter/experiences-software-architecture-analysis-usability/40427](http://www.irma-international.org/chapter/experiences-software-architecture-analysis-usability/40427)

### A Hybrid Approach for Web Change Detection

Sakher Khalil Alqaaidi (2013). *International Journal of Information Technology and Web Engineering* (pp. 46-69).

[www.irma-international.org/article/a-hybrid-approach-for-web-change-detection/89329](http://www.irma-international.org/article/a-hybrid-approach-for-web-change-detection/89329)

### Quality of Service Management by Third Party in the Evolved Packet System

Ivaylo Atanasovand Evelina Pencheva (2012). *Models for Capitalizing on Web Engineering Advancements: Trends and Discoveries* (pp. 276-305).

[www.irma-international.org/chapter/quality-service-management-third-party/61911](http://www.irma-international.org/chapter/quality-service-management-third-party/61911)

### Quality Measures for Semantic Web Application

Adiraju Prasanth Rao (2016). *Design Solutions for Improving Website Quality and Effectiveness* (pp. 130-139).

[www.irma-international.org/chapter/quality-measures-for-semantic-web-application/143370](http://www.irma-international.org/chapter/quality-measures-for-semantic-web-application/143370)

### Energy Efficient Scheduling for Multiple Workflows in Cloud Environment

Ritu Gargand Neha Shukla (2018). *International Journal of Information Technology and Web Engineering* (pp. 14-34).

[www.irma-international.org/article/energy-efficient-scheduling-for-multiple-workflows-in-cloud-environment/204357](http://www.irma-international.org/article/energy-efficient-scheduling-for-multiple-workflows-in-cloud-environment/204357)