

بسم الله الرحمن الرحيم

**Sudan University of Science and Technology  
College of Graduate Studies**

***Investigation of Aspect Oriented  
Programming Support for Crosscutting  
Concerns  
Case Study: Account Reconciliation  
System***

**برمجة السمات لتطبيق المفاهيم المتقاطعة  
دراسة حالة : نظام تسوية الحسابات**

**A thesis submitted in partial fulfillment of the  
Requirements for the degree of MSc in Computer Science**

**Prepared by:**  
Hind Awad Farajallah Ahmed

**Supervisor**  
Dr. Mohamed Alhafiz Mustafa Musa

**October 2008**

{اللَّهُ لَا إِلَهَ إِلَّا هُوَ الْحَيُّ الْقَيُّومُ لَا تَأْخُذُهُ سِنَّةٌ وَلَا نَوْمٌ لَهُ مَا فِي السَّمَاوَاتِ وَمَا فِي الْأَرْضِ مَنْ ذَا الَّذِي يَشْفَعُ عِنْدَهُ إِلَّا بِإِذْنِهِ يَعْلَمُ مَا بَيْنَ أَيْدِيهِمْ وَمَا خَلْفَهُمْ وَلَا يُحِيطُونَ بِشَيْءٍ مِنْ عِلْمِهِ إِلَّا بِمَا شَاءَ وَسِعَ كُرْسِيُّهُ السَّمَاوَاتِ وَالْأَرْضَ وَلَا يَئُودُهُ حِفْظُهُمَا وَهُوَ الْعَلِيُّ الْعَظِيمُ}

البقرة: ٢٥٥

## Acknowledgement

This thesis is the result of my strong commitment to study a new field that brought me so many challenges and valuable experiences. It was Dr. Nael Advanced Software Engineering classes that inspired me and led me to start this thesis.

I would like to express my sincere gratitude to my advisor Dr. Mohamed Alhafiz, for all his valuable guidance, teaching and encouragement during the realization of this research.

This work could not be done without the unconditional love and support from my family, to whom my thesis is dedicated. My love for my family provides endless motivation and inspiration to help me overcome obstacles and fulfill my goals.

Words fail me to express my appreciation to my husband Abdelrahman whose dedication, love and persistent confidence in me, has taken the load off my shoulder. I owe his for being unselfishly let his intelligence, passions, and ambitions collide with mine.

## المستخلص

تهدف هندسة البرمجيات الى انتاج انظمة حاسوبية ذات جودة عالية. في البرمجة كائنية المنحى " Object Oriented Programming (OOP)" يمثل كل مفهوم بوحدة منفصلة تسمى (صنف) تحوي كل هياكل البيانات والإجراءات المتعلقة بالصنف. ولكن هناك بعض المفاهيم يستعصى حصرها داخل صنف واحد. هذا النوع من المفاهيم يسمى المفاهيم المتقاطعة "Crosscutting concerns" و الذي يؤثر سلبا على جودة المنتج البرمجي الكائني و يجعل تطويره معقدا. لتلافي هذه المشكلة استحدثت برمجة السمات "Aspect Oriented Programming (AOP)" لتساعد في برمجة المفاهيم المتقاطعة.

هذه الدراسة هي دراسة تطبيقية تختبر صلاحية ال "AOP" لحل مشكلة المفاهيم المتقاطعة عن طريق تطبيقه في نظام معاملة الحسابات البنكية والذي يحوي العديد من المفاهيم المتقاطعة كالتحكم بالدخول للنظام, التحقق من مستخدميه , معالجة الاخطاء و التحقق من المدخلات و المخرجات من والى النظام.

أوضحت هذه الدراسة أن "AOP" تمكنت من تحقيق زيادة في جودة المنتج البرمجي في بعض الخصائص كقابليته للفهم والقراءة والصيانة واعادة الاستخدام, بينما في الجانب الاخر اثرت سلبا على صحته و اختباريته.

## Abstract

Software engineering seeks the realization of concerns in computerized systems. In object-oriented programming (OOP) each concern is realized by a separate entity (class). But there are some concerns such as logging, security, and so forth, which need to be realized using many entities. These types of concerns are called "Crosscutting concerns". The implication of the crosscutting concerns compromise software modularity in terms of lower productivity, poor quality, and that they make the evolution of the designed systems very complicated.

Aspect-oriented programming (AOP) overcomes these problems by modularizing the crosscutting concerns through explicit abstractions called aspects and composition mechanisms for composing the aspects with the software components.

This thesis investigates to what extent AOP is a practical solution for the mentioned problems. The thesis contains a case study for implementing AOP approach in developing ARS (Account Reconciliation System). The Aspect oriented based ARS represents a foundation for a modular version for bank accounts which contains crosscutting concerns treated as aspects. These concerns are Logging, Access control (Authentication and Authorization), Error handling, Transaction management and Input/output validation.

Our case study shows that the AOP has the potential to increase the quality of a software implementation with regard to its modularity, maintainability, readability, understandability, and reusability. However, AOP may cause problems with the structural complexity, correctness and testability of a software implementation.

## **LIST OF ACRONYMS**

<b>AOP</b>	Aspect Oriented Programming
<b>ARS</b>	Account Reconciliation System
<b>DBMS</b>	Database Management System
<b>IDE</b>	Integrated Development Environment
<b>OOP</b>	Object Oriented Programming

<b>POP</b>	Post-object programming
<b>SOC</b>	Separation of Concerns
<b>SQL</b>	Structured Query Language

## **LIST OF FIGURES**

<b>Figure</b>	<b>Title</b>	<b>Page</b>
Figure 2.1	Simple transfer method	10
Figure 2.2	Tangling and Scattering problems	11
Figure 3.1	Logging Example using OOP Approach	18
Figure 3.2	Logging Example using AOP Approach	19
Figure 3.3	Differences between tangled code and AOP solution	20

Figure 3.4	Weaver or aspect compiler	20
Figure 3.5	Comparing Object Oriented Model and Aspect Oriented Model	22
Figure 3.6	Phases of AOSD for an existing project	23
Figure 4.1	Account Reconciliation System Architecture Design without Aspects	29
Figure 4.2	ARS Class Diagram	30
Figure 4.3	Account Reconciliation System Architecture Design with Aspects	31
Figure 4.4	Completed Class Diagram for ARS	32
Figure 4.5	ARS Package Diagram as Core System + Extensions	33
Figure 4.6	An overview of composition in the AO approach	34
Figure 4.7	Authentication Aspect	39
Figure 4.8	ValidateAmount Aspect	40
Figure 4.9	ExceptionHandling Aspect	40
Figure 4.10	Trace Aspect	41
Figure 4.11	Logging Aspect	41
Figure 4.12	DatabaseAccess Aspect	42
Figure 4.13	InsufficientBalance Aspect	42

## **TABLE OF CONTENTS**

Acknowledgment .....	VII
Abstract .....	VII
List of Acronyms.....	VII
List of Figures.....	VII
Table of Contents .....	VII

### **Chapter 1: Introduction**

1.1 Introduction .....	2
1.2 Problem Statement .....	3
1.3 Motivation.....	4
1.4 Thesis Objectives.....	4
1.5 Methodology.....	5
1.6 Organization of the Thesis.....	5
<b>Chapter 2: Crosscutting Concerns</b>	
2.1 Introduction .....	7
2.2 Crosscutting Problem.....	8
2.2.1 The separation of concerns.....	8
2.2.2 Crosscutting Concerns .....	9
2.2.3 Symptoms of nonmodularization .....	11
2.2.4 Implications of nonmodularization .....	13
2.3 Introducing AOP .....	14
<b>Chapter 3: Aspect Oriented Programming (AOP)</b>	
3.1 Overview .....	16
3.2 The need for AOP paradigm.....	17
3.3 Aspects, Join points and Pointcuts.....	18
3.4 Aspect-Oriented vs. Object-Oriented Programming.....	22
3.5 Software engineering with aspects.....	23
3.6 Language Mechanisms for Capturing Crosscutting Concerns.....	25
3.6.1 AspectJ.....	25
3.7 AOP Applications.....	26
3.8 Adoption risks.....	26
<b>Chapter 4: Aspect Oriented Account Reconciliation System</b>	
4.1 Overview .....	29
4.2 System Analysis .....	29
4.3 System Design .....	30
4.3.1 <i>Design Overview</i> .....	30
4.3.2 <i>Account Reconciliation System Design</i> .....	33
4.3.2.1 <i>Tools and Languages</i> .....	33
4.3.2.2 <i>Class Diagram</i> .....	33
4.3.2.3 <i>Package Diagram</i> .....	34
4.4 Implementation .....	34
4.4.1 <i>Composition Overview</i> .....	34



4.4.2 <i>Tools and Programming Languages</i> .....	35
4.4.3 <i>Implementation Methodology</i> .....	36
4.4.3.1 <i>Aspect decomposition</i> .....	36
4.4.3.1.1 <i>Primary model</i> .....	36
4.4.3.1.2 <i>Aspect model (crosscutting concerns)</i> .....	37
4.4.3.2 <i>Concern implementation</i> .....	39
4.4.3.3 <i>Aspect re-composition</i> .....	39
4.4.4 <i>System Description</i> .....	39
<b>Chapter 5: Results Discussion</b>	
5.1 <i>Software Qualities and Aspect-Oriented Programming</i> .....	45
5.1.1 <i>Software Modularity</i> .....	45
5.1.2 <i>Software Complexity</i> .....	46
5.1.3 <i>Software Correctness</i> .....	47
5.1.4 <i>Software Reusability</i> .....	47
5.1.5 <i>Software Testability</i> .....	48
5.1.6 <i>Software Maintainability</i> .....	48
5.2 <i>Summary</i> .....	49
<b>Chapter 6: Conclusion</b>	
6.1 <i>Introduction</i> .....	51
6.2 <i>Contribution</i> .....	51
6.3 <i>Areas for Further Investigation</i> .....	51
Appendix A - <i>AOP TOOLS</i> .....	53
Appendix B - <i>USECASE DIAGRAM</i> .....	54
Appendix C - <i>SAMPLE SEQUENCE DIAGRAMS</i> .....	56
REFERENCES.....	58