



**AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER AND SYSTEMS DEPARTMENT**

Mining of Software Design Patterns

A THESIS
SUBMITTED FOR THE PARTIAL FULFILLMENT OF
MASTER DEGREE IN ELECTRICAL ENGINEERING
(COMPUTER AND SYSTEMS ENGINEERING)

By

Eng. MOHAMED ABOU-BAKR MOHAMED SOLIMAN
B.Sc. in Computer and Systems Engineering
Faculty of Engineering, Ain shams University, 2005

Supervised by

Prof. Dr. AYMAN MOHAMED WAHBA	Dr. ISLAM AHMED ELMADDAH
Faculty of Engineering	Faculty of Engineering
Ain shams University	Ain shams University

Cairo 2010

APPROVAL SHEET

Thesis title: Mining of Software Design Patterns

Presented by: Eng. Mohamed Abou-Bakr Mohamed Soliman

Degree Name: Master of Electrical Engineering
(Computer and system Engineering)

Thesis Defense Committee

Name, Title and Affiliation

Signature

1. **Prof. Dr. Mohamed Gamal El-Din Darwish**
Faculty of Computers and Information, Cairo University

2. **Prof. Dr. Hani Mohamed Kamal Mahdi**
Faculty of Engineering, Ain shams University

3. **Prof. Dr. Ayman Mohamed Wahba**
Faculty of Engineering, Ain shams University

STATEMENT

This dissertation is submitted to Ain Shams University in fulfillment of the requirements for master degree in Electrical Engineering.

This work included in the thesis was made by the author during the period from October 2005 to October 2010 at Computer and Systems Engineering Department, Ain Shams University.

No part of this thesis has been submitted for degree or qualification at any other university or institute.

Date:

Signature:

Name: Mohamed Abou-Bakr Soliman

Contents

List of Figures	ix
List of Tables	xi
Acknowledgments	xiii
Abstract	xv
1 Introduction	1
2 Background	5
2.1 Understanding Design Patterns	5
2.1.1 Design Patterns Foundation	5
2.1.2 Design Patterns in Reverse Engineering	10
2.2 Design Pattern Mining Challenges	12
2.3 Classification of Related Work	14
2.3.1 Detect Pattern Structural Aspects	15
2.3.2 Detect Pattern Behavioral Aspects	16
2.3.3 Search Pattern Implementation Variants	17
2.3.4 Improve Pattern Mining Performance	17
2.3.5 Detect Partial/Distorted Patterns	18
2.4 Variety of Design Patterns' Implementations	18
3 Design Pattern Description	25
3.1 Formalizing Design Patterns	25
3.2 Proposed Design Pattern Description	28
3.2.1 Design Pattern Description (DPD) Model	29
3.3 GoF Patterns Description	30
4 Design Pattern Miner (DPM)	37
4.1 DPM Mining Process	37
4.2 ASG Creation	39
4.3 Sub-Graph Matching CSP	43
4.4 DPD to CSP Transformation	48
4.5 DPM Implementation	53

5	DPM Evaluation and Results	55
5.1	Pattern Mining Evaluation Aspects.....	55
5.2	DPM Evaluation and Results	57
5.3	Design Pattern Miner Tools Comparison.....	60
6	Conclusion and Future Work	65
A	ASG Specification	67
A.1	AST Specification	68
A.2	ASG Enrichments.....	71
B	CSP Foundation	75
B.1	Introduction	75
B.2	CSP Solving	78
C	DPM QVT Rules	83
D	DPM Design Specification	107
D.1	DPM Architecture	108
D.2	DPM Packages Design.....	110
D.3	DPM GUI.....	112
E	Benchmark Design Patterns	115
E.1	JUnit 3.8.....	116
E.2	JUnit 4.1.....	118
E.3	AWT 1.3	121
	References	135

List of Tables and Figures

List of Figures

2.1	OOA/D process and the role of patterns.....	6
2.2	The GoF Singleton pattern structure	8
2.3	An Example of a Singleton pattern implementation code snippet from the Java AWT framework.....	8
2.4	The GoF Composite pattern structure	9
2.5	An Example of a Composite pattern implementation code snippet from the JUnit 3.8 framework.	10
2.6	Design Pattern Mining Challenges Hierarchy	12
2.7	Template Method pattern GoF OMT structure, and examples of Template Method Java implementation variants in JUnit framework.	20
2.8	Bridge pattern GoF OMT structure, and examples of Bridge Java implementation variants in Java AWT.	21
2.9	An Example for a Decorator pattern implementation.	22
2.10	Two examples for different one to many association implementations for the Composite pattern	23
3.1	FUJABA's specification for the Singleton and Abstract Factory Patterns	26
3.2	MUSCAT's specification for the Bridge and Abstract Factory Patterns	27
3.3	Design Pattern Description (DPD) domain model	29
3.4	Composite DPD structure	31
3.5	Decorator DPD structure	33
3.6	Bridge DPD structure	35
3.7	Template Method DPD structure.....	36
4.1	The design pattern mining process implemented by DPM.....	38
4.2	Code snippet from the AWT Java framework and the AST result from parsing.....	40

4.3	The type nodes to declaration nodes relationships enrichment, the blue lines represent the new declaration relationships.....	41
4.4	Classes relationships enrichment, the red lines represent the new classes relationships.	42
4.5	The pattern matching CSP domain model	45
4.6	The pseudocode for the AC-3 CSP Reduction algorithm	47
4.7	The pseudocode for the Chronological Backtracking CSP search algorithm	48
4.8	The transformation of the Singleton pattern DPD to a CSP based on the DPM defined QVT rules.	51
4.9	The transformation of the Bridge pattern DPD to a CSP based on the DPM defined QVT rules.	51
5.1	DPM accuracy vs. PINOT accuracy, based on the AWT 1.3 evaluation results.	62
A.1	ASG Package node sub-tree	68
A.2	ASG TypeDeclaration node sub-tree	69
A.3	ASG Type node sub-tree	69
A.4	ASG Method node sub-tree.....	70
A.5	ASG Formal Parameter node sub-tree	70
A.6	ASG Field node sub-tree	71
A.7	ASG TypeDeclaration node static relations and properties	72
A.8	ASG Method node static relations and properties.....	73
A.9	ASG Method node dynamic relations and properties	73
B.1	Control of the chronological backtracking (BT) algorithm.....	81
B.2	Cost of problem reduction vs. cost of backtracking.	82
D.1	DPM Package Diagram.....	108
D.2	DPDToCSP Adapter classes	110
D.3	ASGCreator package class diagram.....	111
D.4	DPM Parser class diagram	112
D.5	DPM GUI snapshot for the JUnit 4.1 Packages	113
D.6	DPM GUI snapshot for the JUnit 4.1 classes and design patterns	114

List of Tables

5.1	DPM mining results on JUnit 3.8	58
5.2	DPM mining results on JUnit 4.1	58
5.3	DPM mining results on AWT 1.3.....	58
5.4	DPM performance evaluation results in seconds	60
5.5	PINOT mining results on AWT 1.3.....	62
E.1	JUnit 3.8 Design Patterns' Instances.....	116
E.2	JUnit 4.1 Design Patterns' Instances.....	118
E.3	AWT 1.3 Design Patterns' Instances.....	121

Acknowledgments

My thesis would not have been possible without the expert guidance and everlasting patience of my advisors, Prof. Dr. Ayman Wahba and Dr. Islam El-Maddah. I thank them for their comments which were always insightful and appropriate. Many thanks go to my thesis committee for revising my thesis. I thank them for their efforts and comments in perfecting my thesis.

The writing of a thesis can be a lonely and isolating experience, yet my research journey to an MSc degree would not have been possible without the help and support of numerous people. Thus, my sincere gratitude goes to my mother, father, and all my friends for their love and their personal and practical support over the last few years.

Special thanks to my leaders in Hewlett Packard – HP Company for their continuous support and their invitation to attend the HP Tech Conference.

Mohamed A. Soliman
Computer and Systems Engineering Dept.

Mining of Software Design Patterns

Abstract

One of the most repeatable and time consuming tasks in the software development process is trying to analyze and understand the technical implementation and design of an existing program source code. Design patterns encapsulate solutions to common object oriented design problems which make them an aid to understand different design decisions taken by software designers in existing software. Thus, by detecting design patterns from the source code, we can understand the main intent of the design. Current approaches for design patterns detection described design patterns based on a specific pattern implementation without considering the pattern semantics which prevents the detection of many existing pattern implementation variants. This thesis presents a new, fully automated source code pattern mining approach. The new approach is based on a new design pattern description which describes design patterns based on the pattern semantics. We claim that design pattern description has a great impact on the accuracy of a design pattern mining tool. Our approach is capable of detecting a variety of design patterns' implementations based on the proposed pattern description. This thesis also describes our tool, DPM, that implements the new pattern description based on a model transformation approach. DPM is using graph matching and constrain satisfaction problem (CSP) algorithms to detect patterns' instances. DPM has been tested against benchmark applications, including JUnit and Java AWT. Comparison to