Investigation of Aspect-Oriented Metrics for Stability Assessment: A Case Study

Mahmoud O. Elish, Mojeeb Al-Khiaty, Mohammad Alshayeb
Information & Computer Science Department
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia
Emails: {elish, alkhiaty, alshayeb@kfupm.edu.sa

***Abstract* — Stability assessment provides software managers with early insight into trends in software evolution, and thus assists them in managing and controlling long-lived software systems. However, there are few empirical studies that have been conducted to relate software metrics with external quality attributes of aspect-oriented software in general, and metrics have not been evaluated as indicators of aspect stability in particular. This paper investigates the relationships between 13 aspect-oriented metrics and aspect stability. These metrics measure different structural properties of an aspect: size, coupling, cohesion, and inheritance. A case study was conducted using an open source aspect-oriented software consisting of 76 aspects. The results obtained from this study indicate statistically significant correlation between most of the size metrics and aspect stability. The cohesion metric was also found to be significantly correlated with aspect stability. In addition, different prediction models were built using different combinations of metrics' categories. It was observed that the best accuracy was achieved as a function of some of the size and inheritance metrics.**

***Index Terms* — software metrics, software stability, aspect-oriented software.**

I. INTRODUCTION

Separation of concerns is one of the vital principles in software engineering for achieving quality software [4]. Although different approaches — including object-oriented programming, component-oriented programming, and design patterns — provide useful modularity mechanisms, none of them satisfactorily modularize all concerns of complex software systems. Some of the concerns still inherently crosscut the modularity of multiple modules and are difficult to be captured by these techniques [6]. Aspect-Oriented Software Development (AOSD) provides an explicit concepts and mechanisms for separating the crosscutting concerns [6]. It is increasingly getting popularity as useful practice to improve the modularization of software artifacts.

It is vital to quantitatively assess the quality of software produced using AOSD. In this regard, software metrics are needed to do such assessment. The external quality attributes of aspect-oriented software are usually assessed, using modeling techniques, as a function of metrics that measure the internal structural properties of the aspect-oriented software. However, there are few empirical studies [1, 7, 8, 10] that have been conducted to relate software metrics with external quality attributes of aspect-oriented software in general, and metrics have not been evaluated as indicators of aspect stability in particular.

Motivated by the foregoing issues, in this paper, we empirically investigate the relationships between a suite of aspect-oriented metrics and aspect stability. In other words, this paper investigates whether or not the metrics under investigation are good early indicators of aspect stability in aspect-oriented software. Aspect stability, in this paper, refers to the extent to which the revisions made to an aspect are infrequent. The availability of adequate metrics for stability assessment provides software managers early insight into trends in software evolution, and thus assists them in managing and controlling long-lived software systems.

The rest of this paper is organized as follows Section II defines the aspect-oriented metrics under investigation. Section III discusses the case study and its results. Section IV reviews related work. Section V concludes the paper.

II. ASPECT-ORIENTED METRICS

In this study, 13 aspect-level metrics are investigated as indicators of aspects' stability. These metrics were chosen because: (i) they measure different structural properties of an aspect: size, coupling, cohesion, and inheritance; and (ii) they refine classical object-oriented metrics, such as C&K metrics [3], which are well-established and based on sound measurement theory. The metrics are defined next.

A. *Size Metrics*

- Number of Attributes (NA) [7]: The NA metric of an aspect is defined as the number of attributes defined in the aspect.
- Number of Methods (NM): The NM metric of an aspect is defined as the number of methods defined in the aspect.

doi:10.4304/jsw.6.12.2508-2514

- Number of Advices (NAD) [9]: The NAD metric of an aspect is defined as the number of advices in the aspect.
- Number of Pointcuts (NP) [9]: The NP metric of an aspect is defined as the number of pointcuts in the aspect.
- Number of Introductions (NI) [9]: The NI metric of an aspect is defined as the number of introductions (intertype declarations) in the aspect.
- Lines of Code (LOC) [7]: The LOC metric of an aspect is the number of lines of code, excluding comment and blank lines, in the aspect.

B. *Coupling Metrics*

- Afferent Coupling through Introductions (ACTI): The ACTI metric of an aspect is defined as the number of classes and aspects that are affected by the aspect through introductions.
- Afferent Coupling through Pointcuts (ACTP): The ACTP metric of an aspect is defined as the number of classes and aspects that are affected by the aspect through pointcuts.
- Efferent Coupling through Introductions (ECTI): The ECTI metric of an aspect is defined as the number of aspects that affect the aspect through introductions.
- Efferent Coupling through Pointcuts (ECTP): The ECTP metric of an aspect is defined as the number of aspects that affect the aspect through pointcuts.

C. *Cohesion Metric*

- Lack of Cohesion in Operations (LCO) [7]: The LCO metric of an aspect is defined as the number of method/advice pairs that do not access the same instance variable.

D. *Inheritance Metrics*

- Depth of Inheritance (DIT) [7]: The DIT metric of an aspect is defined as the level of the aspect in its inheritance hierarchy, i.e. the length of the longest path from the aspect to the root of inheritance tree.
- Number of Children (NOC) [2]: The NOC metric of an aspect is defined as the number of immediate sub-aspects of the aspect, i.e. the number of aspects that inherit directly from the aspect.

III. THE CASE STUDY

The main objective of this case study is to investigate the relationships between the 13 aspect-level metrics, described in the previous section, and aspect stability. For this purpose, a large open source aspect-oriented software system called Glassbox[1] was chosen because of the availability of its revision history. The system, which is written in Java and ApectJ, is an automated troubleshooting and monitoring agent for Java applications. It consists of 76 aspects and about 420 classes.

[1] http://sourceforge.net/projects/glassbox/

A. *Descriptive Statistics*

The 13 aspect-oriented metrics were collected from each aspect in the first version of the Glassbox system. Table 1 provides their descriptive statistics. It can be observed that aspects vary in size in terms of the number of attributes, methods, advices, pointcuts, introductions, etc. Furthermore, there is a relatively good utilization of inheritance, and aspects' afferent couplings are more than their efferent couplings.

The dependent variable is aspect stability. The number of revisions made to an aspect was used as a proxy measure for its stability, i.e. the more revisions the less stable it is. It was collected using the Concurrent Versions System (CVS) repository of the Glassbox system. The log data (i.e. revision history) for each aspect was obtained by using the 'log' subcommand of 'cvs' command, which contains the total number of revisions made to it. These revisions might be corrective, perfective, adaptive, and/or preventive. In Glassbox system, the number of revisions made to each aspect varies from 1 to 9 with an average of 1.7.

TABLE 1.
DESCRIPTIVE STATISTICS

Metric	Min	Max	Avg.	Std. Dev.
NA	0	9	1.29	1.91
NM	0	31	3.03	5.62
NAD	0	12	1.87	2.45
NP	0	29	2.91	4.01
NI	0	13	0.99	2.49
LOC	2	283	44.95	57.95
ACTI	0	381	7.09	45.61
ACTP	0	500	12.08	59.72
ECTI	0	28	1.39	3.11
ECTP	0	3	2.28	0.74
LCO	0	219	4.93	28.36
DIT	0	5	1.45	1.72
NOC	0	10	0.36	1.45

B. *Correlation Analysis*

The correlation analysis aims to determine if each individual aspect-oriented metric is significantly related to aspect stability. For this purpose, Spearman's rank correlation was performed due to the nonparametric nature of the metrics. The significance of the correlation was tested at 99% confidence level (i.e. p-level ≤ 0.01). The results obtained by applying this analysis are given in Table 2, where bold values indicate statistically significant correlations. All size metrics, except the NM metric, were found to be significantly correlated with aspect stability. In addition, the cohesion metric (LCO) was also found to be significantly correlated with aspect stability. This is not the case, however, for the coupling and inheritance metrics.

TABLE 2.
SPEARMAN CORRELATION ANALYSIS RESULTS

Metric	Correlation Coefficient (r)	p-value
NA	**0.375**	**<0.01**
NM	0.177	0.125
NAD	**0.474**	**<0.01**
NP	**0.479**	**<0.01**
NI	**0.361**	**<0.01**
LOC	**0.629**	**<0.01**
ACTI	0.159	0.169
ACTP	0.167	0.150
ECTI	-0.099	0.394
ECTP	0.112	0.335
LCO	**0.305**	**<0.01**
DIT	0.130	0.264
NOC	0.034	0.768

C. *Multivariate Regression Analysis*

Multivariate Linear Regression (MLR) is the most commonly used technique for modeling the relationship between two or more independent variables and a dependent variable by fitting a linear equation to observed data. The main advantages of this technique are its simplicity and that it is supported by many popular statistical packages. The multivariate analysis was performed to construct different MLR prediction models for predicting aspect stability as a function of the 13 metrics (independent variables) under investigation. Since there are four categories of these metrics (size, coupling, cohesion, and inheritance), 15 different prediction models were built that represent all possible combinations of these categories (see Table 3).

TABLE 3.
CATEGORY(S) OF METRICS IN EACH MODEL

Model	Size Metrics	Coupling Metrics	Cohesion Metric	Inheritance Metrics
M1	√			
M2		√		
M3			√	
M4				√
M5	√	√		
M6	√		√	
M7	√			√
M8		√	√	
M9		√		√
M10			√	√
M11	√	√	√	
M12	√	√		√
M13	√		√	√
M14		√	√	√
M15	√	√	√	√

1) *Variables Selection*

Variables selection is a preliminary step used in multivariate data analysis. When there are many independent variables, there is a possibility that some of these variables contain redundant or noisy information. Additionally, there can be a high correlation between independent variables which can adversely affect the results meanwhile it does not add new information. Thus, it is useful to reduce the number of independent variables and remove the collinearity in each model. In this study, we performed the best first search method in WEKA machine learning toolkit for variables selection [11]. It searches the space of variable subsets by greedy hill climbing augmented with a backtracking facility.

The selected independent variables and their regression coefficient for all the 15 prediction models are provided in Table 4. An empty cell indicates that the corresponding metric is not one of the independent variables in the corresponding model. It can be observed that some models are the same (M1 and M6; M5 and M11; M7 and M13; M12 and M15) due to the variables selection step. This means that there are 11 different models out of the 15 models, which will be considered in the subsequent analyses.

There are some interesting observations that can be obtained from Table 4. First, only NP and LOC metrics were selected from the size metrics category. Moreover, the impact of the NP metric is stronger than the LOC metric because the regression coefficient of the NP metric is higher than the regression coefficient of the LOC metric. Second, the only coupling metric that was selected is the ACTI metric, but its regression coefficient is weak. Third, the LCO metric was selected with coupling and inheritance metrics, but not with size metrics. Finally, the NOC metric had the strongest impact (i.e. highest regression coefficient) on the prediction models.

2) *Models' Goodness of Fit*

In order to measure and evaluate the goodness of fit for each model, we used R^2 (R-squared). It indicates what percentage of the variability in the dependent variable can be explained by the independent variables in each model. The R^2 value for each prediction model is provided in Table 5 and visualized by Figure 1. The M12 model (based on size, coupling and inheritance metrics) has the highest R^2 value (0.625), whereas the M4 model (based on the inheritance metric only) has the lowest R^2 value (0.042). Another interesting observation is that those models that include size metrics have better R^2 values compared to those that do not include them. It is also interesting to observe that improved R^2 value could be achieved by considering other structural properties in addition to size. For instance, improved R^2 value was achieved by M5, M7 and M12 models compared to the M1 model.

3) *Cross Validation*

A leave-one-out cross-validation procedure was used to evaluate and compare the accuracy of the prediction models. In this procedure, one observation is removed from a dataset with n observations, and then each prediction model is built with the remaining $n-1$ observations and evaluated in predicting the value of the observation that was removed. The process is repeated n times; each time removing a different observation.

TABLE 4.
SELECTED METRICS AND THEIR REGRESSION COEFFICIENT FOR THE REGRESSION MODELS

Model		Size Metrics						Coupling Metrics				Cohesion Metric	Inheritance Metrics	
	Constant	NA	NM	NAD	NP	NI	LOC	ACTI	ACTP	ECTI	ECTP	LCO	DIT	NOC
M1	1.154				0.158		0.014							
M2	2.154							0.010						
M3	2.074											0.030		
M4	2.138													0.241
M5	1.155				0.147		0.014	0.005						
M6 (same as M1)	1.154				0.158		0.014							
M7	1.072				0.148		0.014							0.237
M8	1.998							0.010				0.031		
M9	1.791							0.011					0.191	0.221
M10	1.976											0.031		0.267
M11 (same as M5)	1.155				0.147		0.014	0.005						
M12	1.070				0.135		0.014	0.005						0.244
M13 (same as M7)	1.072				0.148		0.014							0.237
M14	1.673							0.011				0.031	0.158	0.253
M15 (same as M12)	1.070				0.135		0.014	0.005						0.244

TABLE 5.
MODELS' GOODNESS OF FIT

Model	R^2
M1 (also M6)	0.568
M2	0.069
M3	0.255
M4	0.042
M5 (also M11)	0.583
M7 (also M13)	0.608
M8	0.331
M9	0.150
M10	0.306
M12 (also M15)	0.625
M14	0.410

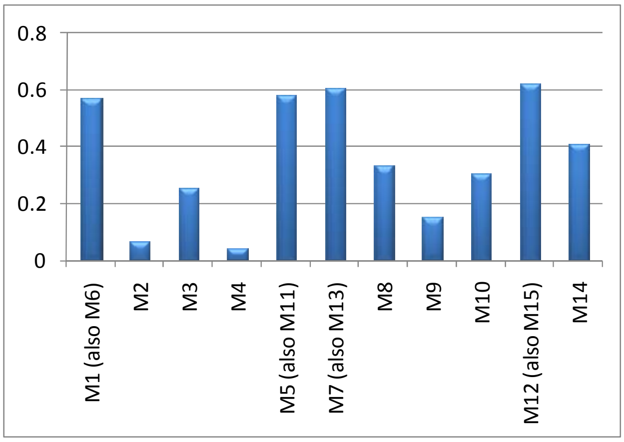

Figure 1. Models' R-squared.

The accuracy of the prediction models were evaluated based on de facto standard and commonly used measures: mean magnitude of relative error (MMRE) and prediction at level *q* (Pred(*q*)) measures. MMRE over a dataset of *n* observations is calculated as follows:

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i$$

where MRE_i is a normalized measure of the discrepancy between the actual value (x_i) and the predicted value ($\hat{x}_i$) of observation *i*. It is calculated as follows:

$$MRE_i = \frac{|x_i - \hat{x}_i|}{x_i}$$

Pred(*q*) is a measure of the percentage of observations whose MRE is less than or equal to *q*. It is calculated as follows:

$$Pred(q) = \frac{k}{n}$$

where *k* is the number of observations whose MRE is less than or equal to a specified level *q*, and *n* is the total number of observations in the dataset. In this study, we used Pred(0.25), which is commonly used in the literature.

Table 6 shows the MMRE and Pred(0.25) values for the prediction models, and Figure 2 and Figure 3 visualize them respectively. It can be noticed that the M1 model (based on size metrics only) achieved the best MMRE value (0.483), followed by the M7 model (based on size and inheritance metrics) which achieved very competitive MMRE value (0.489). The best model in terms of Pred(0.25) is M7, which achieved a value of 34.2%. Based on these results and from Table 4, we can conclude that the best prediction of aspect stability can be achieved as a function of the NP, LOC and NOC metrics. This means that the higher the number of pointcuts, lines of code, and number of children of an aspect, the less stable it will be (i.e. undergo more revisions). The inclusion of the coupling and cohesion metrics and other size metrics under investigation is not useful for aspect stability prediction.

TABLE 6.
EVALUATION OF THE PREDICTION MODELS

Model	MMRE	Pred(0.25)
M1 (also M6)	0.483	30.3%
M2	0.731	22.4%
M3	0.673	22.4%
M4	0.840	21.1%
M5 (also M11)	0.514	26.3%
M7 (also M13)	0.489	34.2%
M8	0.637	23.7%
M9	0.739	26.3%
M10	0.699	22.4%
M12 (also M15)	0.619	30.3%
M14	0.835	26.3%

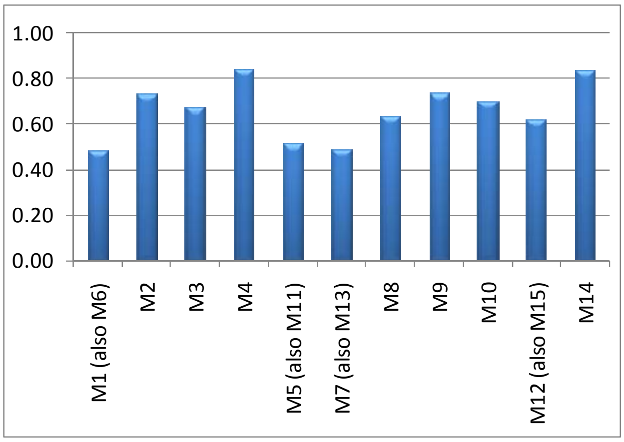

Figure 2. Models' MMRE.

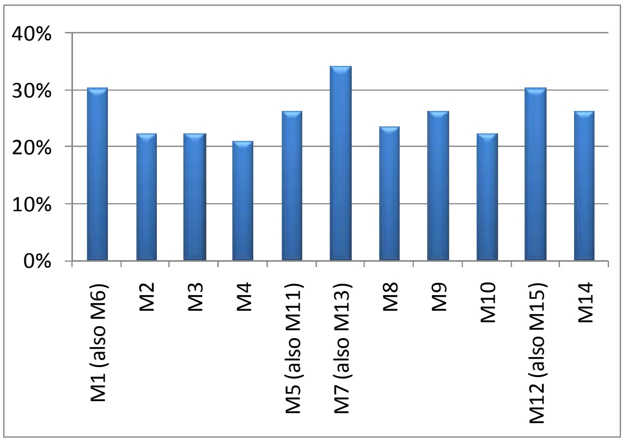

Figure 3. Models' Pred(0.25).

D. *Threats to Validity*

This case study evaluated one open source software. The SourceForge[2] open source software repository was searched for suitable aspect-oriented software but it was challenging because: (1) the number of aspect-oriented software was limited since it is a relatively new paradigm; (2) most of the existing software have either limited number of aspects or no detailed revision history. More future case studies should be conducted, as more open source software with detailed revision history become available, to further support the findings of this paper and to accumulate knowledge.

In this study, aspect stability was measured as the total number of revisions made to it, i.e. the frequency of revisions. Other dimensions of stability can be addressed as future work. In addition, this study was a regression and correlation study. Association between some of the investigated metrics and aspect stability was observed but causality of the association cannot be claimed.

IV. RELATED WORK

There are few research studies that have explored the relationships between aspect-oriented metrics and external software quality attributes [1, 7, 8, 10]. Table 7 summarizes those studies by listing the relevant metrics and the external software quality attributes against which they were empirically evaluated. Cells marked with (√) indicate that significant correlations were observed, whereas cells marked with (X) indicate no significant correlations. Unmarked cell indicates that the corresponding metric was not empirically evaluated against the corresponding external software quality attribute.

Greenwood et al. [5] investigated the impact of aspectual decompositions on design stability. They evaluated the design stability of two different implementations (aspect-oriented and object-oriented) of a real-life web-based information system by applying different types of maintenance tasks. The stability was assessed using traditional suites of modularity and change impact metrics. That is, the more stable design is the one that minimizes the undesirable variation in the values of the metrics after applying the maintenance tasks. The overall conclusion of that study is that "aspect decomposition narrows the boundaries of concern dependencies, however, with more tight and intricate interactions" [5].

The objective of this paper is different from the work of Greenwood et al. [5]. We have investigated a set of metrics as early indicators (predictors) of aspect stability, whereas they compared the design stability of aspect-oriented implementation vs. object-oriented implementation.

[2] http://sourceforge.net/

TABLE 7.
RELATIONSHIPS BETWEEN ASPECT-ORIENTED METRICS AND EXTERNAL SOFTWARE QUALITY ATTRIBUTES

Metrics	External Software Quality Attributes						
	Maintainability			Understandability	Reusability	Fault proneness	Stability
	[10]	[7]	[8]	[10]	[7]	[1]	This Paper
Lines of Code	X			X			√
Weighted Operations in Module	X			X		X	
Operation Cohesion	√			√			
Attribute Cohesion	X			X			
Interface Coupling	√			√			
Coupling Between Components		√			√	√	
Depth of Inheritance Tree		√			√	X	X
Concern Diffusion over Components		√			√		
Concern Diffusion over Operations		√			√		
Concern Diffusions over LOC		√			√		
Number of Attributes		√			√		√
Vocabulary Size		√			√		
Coupling on Method Call			X			√	
Coupling on Field Access			X			√	
Response For a Module			X			X	
Coupling on Advice Execution			X			X	
Crosscutting Degree of Aspects			X			√	
Lack of Cohesion in Operations						X	√
Number of Children						X	X
Base-Aspect Coupling						√	
Number of Methods							X
Number of Advices							√
Number of Pointcuts							√
Number if Introductions							√
Afferent Coupling through Introductions							X
Afferent Coupling through Pointcuts							X
Efferent Coupling through Introductions							X
Efferent Coupling through Pointcuts							X

V. CONCLUSION

This paper has investigated the relationships between 13 aspect-oriented metrics and aspect stability. These metrics measure different structural properties of an aspect: size, coupling, cohesion, and inheritance. A case study was conducted using an open source aspect-oriented software consisting of 76 aspects. The results obtained from this study indicate statistically significant correlation between most of the size metrics and aspect stability. The cohesion metric was also found to be significantly correlated with aspect stability. In addition, different prediction models were built using different combinations of metrics' categories. It was observed that the best accuracy was achieved as a function of some of the size and inheritance metrics: number of pointcuts, lines of code, and number of children. More future case studies should be conducted, as more open source software with detailed revision history become available, to further support the findings of this paper and to accumulate knowledge.

This case study contributes interesting preliminary and novel empirical knowledge about the relationships between some aspect-oriented metrics and aspect stability. Future works include exploring more metrics; conducting more case studies; investigating the impact of author styles and other factors on aspect stability; exploring the relationships between aspect-oriented metrics and other software quality attributes; and building computational intelligence models to improve the prediction accuracy.

ACKNOWLEDGMENT

The authors acknowledge the support of King Fahd University of Petroleum and Minerals (KFUPM). This research work was funded by KFUPM under Research Grant No. IN080420.

REFERENCES

[1] R. Burrows, F. Ferrari, A. Garcia, and F. Taïani, "An Empirical Evaluation of Coupling Metrics on Aspect-Oriented Programs," in *ICSE Workshop on Emerging Trends in Software Metrics*, pp. 53-58, 2010.

[2] N. Cacho, C. Sant'Anna, E. Figueiredo, A. Garcia, T. Batista, and C. Lucena, "Composing Design Patterns: A Scalability Study of Aspect-Oriented Programming," in *5th International Conference on Aspect-Oriented Software Development*, pp. 109-121, 2006.

[3] S. Chidamber and C. Kemerer, "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol. 20, pp. 476-493, 1994.

[4] R. Filman, T. Elrad, S. Clarke, and M. Aksit, *Aspect-Oriented Software Development*: Addison-Wesley, 2005.

[5] P. Greenwood, T. Bartolomei, E. Figueiredo, M. Dosea, A. Garcia, N. Cacho, C. Sant'Anna, S. Soares, P. Borba, U. Kulesza, and A. Rashid, "On the Impact of Aspectual Decompositions on Design Stability: An Empirical Study," in *21st European Conference on Object-Oriented Programming (ECOOP 07), LNCS 4609, Springer*, pp. 176-200, 2007.

[6] R. Laddad, *AspectJ in Action: Practical Aspect-Oriented Programming*: Manning Publications, 2003.

[7] C. Sant'Anna, A. Garcia, C. Chavez, C. Lucena, and A. Staa, "On the Reuse and Maintenance of Aspect-Oriented Software: An Assessment Framework," in *Brazilian Symposium on Software Engineering*, 2003.

[8] H. Shen, S. Zhang, and J. Zhao, "An Empirical Study of Maintainability in Aspect-Oriented System Evolution Using Coupling Metrics," in *2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering*, pp. 233-236, 2008.

[9] K. Srivisut and P. Muenchaisri, "Bad-Smell Metrics for Aspect-Oriented Software," in *6th IEEE/ACIS International Conference on Computer and Information Science*, pp. 1060-1065, 2007.

[10] P. Tonella and M. Ceccato, "Refactoring the aspectizable interfaces: an empirical assessment," *IEEE Transactions on Software Engineering,* vol. 31, pp. 819-832, 2005.

[11] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

Mahmoud O. Elish is an assistant professor of Software Engineering in the Information and Computer Science Department at King Fahd University of Petroleum and Minerals, Saudi Arabia. He received the PhD degree in Computer Science from George Mason University, USA. His research interests include software metrics and measurement, software design, software maintenance, empirical software engineering, and software quality predictive models.

Mojeeb Al-Khiaty is a PhD student in computer science and engineering at King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia. He received his Master of Science (MS) degree in Computer Science from KFUPM in June 2009. He obtained his Bachelor of Science (B.S.) degree with honors in Mathematics/Computer from Sana'a University, Yemen in June 1999.

Mohammad Alshayeb is an assistant professor in Software Engineering at the Information and Computer Science Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. He received his MS and PhD in Computer Science and certificate of Software Engineering from the University of Alabama in Huntsville. He worked as a senior researcher and Software Engineer and managed software projects in the United States and Middle East. He is a certified project manager (PMP). His research interests include Software quality, software measurement and metrics, and empirical studies in software engineering.