A Quantitative Evaluation of Reusability for Aspect Oriented Software using Multi-criteria Decision Making Approach

Pradeep Kumar Singh, Om Prakash Sangwan, Amar Pal Singh and Amrendra Pratap

Amity University, Noida, India
Gautam Buddha University, Greater Noida, India

Abstract: Aspect oriented software development offer several benefits compared to the existing object oriented approach and modular approach. Aspect Oriented (AO) approach combines concerns that crosscut the modularity of the traditional programming like modular programming and object oriented programming. The reusability assessments for software practitioners and researchers have projected the various software quality models. Based on these existing quality models for software, we have proposed a reusability assessment model for AO software using Multi Criteria Decision Making (MCDM) approach. In this paper, the software reusability model for AO system is proposed and validated using Analytic Hierarchy Process (AHP) and cross validated by Fuzzy AHP (FAHP) technique. This proposed software reusability model for AO software may be used to evaluate AO software reusability and also helps software professionals to choose best reusable components and software among the available choices.

Key words: Software quality . software reusability . Aspect Oriented Programming (AOP) . Analytic Hierarchy Process (AHP) . Fuzzy AHP (FAHP) and Multi Criteria Decision Making (MCDM) . Aspect Oriented Software Development (AOSD)

INTRODUCTION

AOSD is reasonably forthcoming technique which combines concerns that crosscut the modularity of conventional programming approaches such as Module Oriented Approach (MOA) and Object Oriented Approach (OOA) [1, 2]. Cross-cutting concerns includes logging, tracing, resource pooling, etc. and may also help in reducing the development and maintenance efforts, if used effectively. AO development is considered as the extension to object oriented languages and enhance the capability of existing OO languages. There are many languages that belongs to family aspect oriented systems such as AspectJ (Java Extension), AspectC (C Extension), AspectC++ (C++ Extension), AspectXML (XML Extension), CaserJ and HyperJ [3, 4]. Significant characteristics of AspectJ are as (i) It used the concept of cross cutting concerns most efficiently on Java Programs (ii) Supports all essential characteristics of AO languages such as joint-points, pointcuts, advice and introduction [5, 6] (iii) It provides synchronization, consistency checking, protocol management and other services effectively as compare to existing technologies.

Software reusability is process of reusing the software with very little or no modification. Software reusability has high demand in industry as it reduces time, cost and effort. There are many approaches that develop reusable software such as Module-Oriented (MO), Object-Oriented (OO) and Component-Based Software Development (CBSD). By the help of these approaches, various extents of reusability like function reusability, code design reusability, sub-system reusability and system reusability can be achieved [1, 2, 7].

MO approach provides functional reusability, OO approach provides functional as well as code design reusability by its inheritance property and Component Based Software Development (CBSD) provides system or sub-system level reusability.

Many researchers [2] have proposed various models and frameworks for assessing reusability of software’s at various levels i.e. at function level [8], at design/code level [9-12], at system/sub-system level [13-16].

The AHP is proposed by Saaty [17, 18]. It use only pair wise matrix to determine the indistinctness in MCDM problems. In this process fuzzy pairwise comparison are done to make decision process to
Utility or Priority based methods consists of Weighted Average Method and Fuzzy AHP. Explanation of each method is given below.

a. Weighted Average Method: Overall utility for a given alternative \( i \), \( U_i \) is expressed as weighted sum of utility values of each criterion \([27, 28]\).

b. Fuzzy AHP: It is an extension of Analytical Hierarchy Process (AHP). In this method elements of pairwise comparison matrix lie between 1 and 9 where as in fuzzy AHP value lie in fuzzy number. This method in more detail is used in section four of the paper.

Miscellaneous methods: There are other methods that use in fuzzy based MCDM approaches as: (i) Fuzzy programming based method (ii) Fuzzy PROMETHEE-2: Geldermann \textit{et al.} in 2000 had taken fuzzy PROMETHEE-2 with trapezoidal based fuzzy number and defuzzification approaches \([29]\).

The rest part of paper splits into three sections. In second part related work in area of software reusability and MCDM approaches is reported. In next section, we will discuss the proposed reusability model. In section four, we will discuss the evaluation of proposed reusability model by analytical hierarchy process. In section five, evaluation of proposed reusability model using fuzzy analytical hierarchy process is reported. Finally, conclusion followed by future work is mentioned in last section.

**RELATED WORK**

Many researchers reported various software reusability frameworks; however empirical validation is not yet reported for AO software \([2]\). In this section of paper, the related work has been discussed in context to reusability of aspect oriented software and their metrics.

Sant’Anna \textit{et al.} projected a framework for assessing the software reusability and maintainability for AO software \([30]\). They established a relationship between reusability and maintainability internal metrics such as coupling, cohesion, crosscutting concern and size.

Cunha \textit{et al.} discussed the high level concurrency patterns and mechanisms coded in AspectJ \([31]\). They explored some advantages of reusability, modularity and understandability..

Zhang \textit{et al.} proposed AO approach with connectors in reusable design and implementation of connectors \([32]\). They uncover the affect of crosscutting concerns in reusability of connectors.

Ajajseret \textit{et al.} discussed the extension of AspectJ programming languages known as ParaAJ \([33]\). This language has ability to parameterize aspects.

Chaudhary \textit{et al.} reported the reusability model which states that predicting reusability over maintainability is easier \([34]\).

Zhao proposed a coupling metrics which is based on number of dependencies among aspects and classes like module-class, attribute-class, aspect-inheritance and module-method dependencies \([35]\).
Table 1: Summarization of quality attributes by researchers that affect reusability of AO software

<table>
<thead>
<tr>
<th>Researcher</th>
<th>Understandability</th>
<th>Modularity</th>
<th>Maintainability</th>
<th>Adaptability</th>
<th>Flexibility</th>
<th>Unplugability</th>
<th>Testability</th>
<th>Analyzability</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sant’Anna et al. [30]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cunha et al. 2006 [31]</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kumar et al. 2009 [45]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chaudhary et al. 2013 [34]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Erlend et al. 2004 [46]</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tsang et al. [47]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zakaria et al. [48]</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A coupling metrics is defined by Ceccato et al. for Aspect Oriented Programming (AOP) languages [36]. They present a framework for aspects, classes and define aspect advices, introductions.

Bartsch et al. [37] extended the framework for OO systems [39]. In their extended framework they contain six diverse coupling schemes such as locus of impact types of connection, firmness of server, granularity, straight or undirected connections, instantiation along with inheritance.

Bartolomei et al. projected a framework for coupling in AO system [38]. They extended their work from the existing framework of Briand’s and Arisholm [39, 40]. They considered two AOP languages, AspectJ and CaesarJ and showed that the how these two languages can be instantiated on Java. In their work they haven’t proposed any new metric.

Kumar et al. proposed a framework of coupling metrics for AO software [41]. They recognized the various types of connection between couplings such as attribute type, parameter type, Attribute reference, operation invocation, inheritance and high level association.

Zhao projected a cohesion measure which is based on dependency graphs [42]. They defined number of dependencies among aspect and classes and cohesion as the degree of relatedness among modules and attributes.

Gelinas et al. define a metrics to measure the cohesion in AO system [43]. This metrics is known as “ACoh Metrics”. Two aspect cohesion criteria are defined in Acoh metrics namely “Modules-Modules Connection Criterion” and “Modules-Data Connection Criterion”.

Kumar et al. defined the cohesion metric for generic/unified AO system [44]. This framework considered Java, AspectJ and CaesarJ languages. Later, they extended their work and correlate cohesion metric value with changeability metric for unified AO system.

An empirical study is conducted by National Institute of Standard and Technology (NIST) [49] which provides the information about the quality attributes that affect the reusability of software. This study contains some indirect attributes namely Understandability, Modularity, Portability, Reliability, Completeness, Maintainability, Adaptability, Efficiency, Learn-ability and Helpfulness.

Even ISO has developed four quality attribute which affects the reusability of software namely Understandability, Learn-ability, Operability, Attractiveness and Usability Compliance [50].

In order to propose the reusability model for AO system, we carried out a survey. In this survey we considered all the 10 characteristics that are provided by National Institute of Standards and Technology (NIST) [49] and ISO which contributes to reusability of AO system [50]. This survey was conducted on 36 different participants. These participants are consisting of 10 industry expert (Software Professionals), 16 (Academician or Professors), 6 PhD scholars and 4 M.Tech. Experts having experience of Object Oriented and Aspect Oriented Technology and having relevant publications are considered for opinion. In case of students, those who have passed the courses of OO and AO programming language were considered.

From this survey we found essential quality attributes and attributes having negligible effect on reusability of AO software. Characteristics that majorly affect the reusability of AO systems are given below:

a. Understandability (C1)
b. Learn-ability (C2)
c. Adaptability (C3)
d. Operability (C4)
e. Maintainability (C5)
f. Modularity (C6)

From these external quality characteristics we can also measure some internal characteristics (architectural metrics) such as Separation of Concern (SoC), coupling, cohesion, size and complexity. Then a reusability model is proposed to establish the
relationship among external quality characteristics, internal attributes and the metrics to measure reusability for aspect oriented software. However, similar model has been proposed by Singh et al. for maintainability assessment of AO software [24].

**EVALUATION OF PROPOSED REUSABILITY MODEL USING AHP METHOD**

Saaty propose the AHP in 1980 [17]. It use only pair wise matrix to determine the indistinctness in MCDM problems. In this work, n elements were considered like C1, C2, …, Cn, which are likely to be compare the relative weight of Ci w.r.t. Cj denoted as aij and a square matrix $A = [a_{ij}]$ of order n as given below.

$$A = \begin{bmatrix}
C_1 & C_2 & \ldots & C_n \\
C_1 & 1 & \alpha_2 & \ldots & \alpha_n \\
C_2 & \frac{1}{\alpha_2} & 1 & \ldots & \alpha_n \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
C_n & \frac{1}{\alpha_n} & \frac{1}{\alpha_n} & \ldots & 1
\end{bmatrix}$$

Here $a_{ij}=1/a_{ji}$, $i \neq j$ and $a_{ii}=1$ for all $i$.

AHP involve human judgment which may be consistent or inconsistent. In this case find vector $\omega$ will satisfy the equation.

$$A \omega = \lambda_{\text{max}} \omega, \lambda_{\text{max}} = n$$

Here

- $\omega$ = Eigen vector,
- $\lambda_{\text{max}}$ = Eigen values

Saaty also projected consistency index (CI) and consistency ratio (CR) to verify the inconsistency in comparison matrix. The equation for evaluating the CI and CR are given below.

$$CI = \frac{\lambda_{\text{max}} - n}{n - 1}$$

$$CR = \frac{CI}{RI}$$

Saaty also stated that CR is greater than 0.1 than our set of judgment is too inconsistent. In that situation new comparison matrix will require for CR=0.1.

**A. Allocating the weights to characteristics:** In order to assign weights to characteristics, we conducted the survey on 36 participants and a form consist 10 tables for filling pair-wise relative weight values of characteristics is provided to each individual. First table is to fill pair-wise relative weight values of six characteristics C1 to C6. We have finalized based on survey, six factors (Understandability, Learnability, Adaptability, Operability, Maintainability and Modularity) as mentioned in Fig.1 to get the values from C1 to C6. The mean of collected samples of pair-wise relative weights are given in square matrix $A = [a_{ij}]$. Now next step is to calculate the eigen vector and eigen value for C1 to C6.
Table 2: Factor values using AHP technique [17]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.58</td>
<td>0.9</td>
<td>1.12</td>
<td>1.24</td>
<td>1.32</td>
<td>1.41</td>
<td>1.45</td>
<td>1.49</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Eigen vector and values

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>Nth root of product</th>
<th>Eigen vector (w)</th>
<th>Eigen value (Aw)</th>
<th>( \lambda = Aw/w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.750</td>
<td>1.499</td>
<td>1.252</td>
<td>1.821</td>
<td>2.210</td>
<td>1.746</td>
<td>0.275</td>
<td>1.829</td>
<td>6.650</td>
</tr>
<tr>
<td>C2</td>
<td>0.266</td>
<td>1</td>
<td>1.847</td>
<td>1.319</td>
<td>1.419</td>
<td>2.389</td>
<td>1.140</td>
<td>0.179</td>
<td>1.158</td>
</tr>
<tr>
<td>C3</td>
<td>0.667</td>
<td>0.541</td>
<td>1</td>
<td>1.335</td>
<td>1.069</td>
<td>1.984</td>
<td>1.003</td>
<td>0.158</td>
<td>0.971</td>
</tr>
<tr>
<td>C4</td>
<td>0.798</td>
<td>0.758</td>
<td>0.749</td>
<td>1</td>
<td>1.681</td>
<td>1.502</td>
<td>1.022</td>
<td>0.162</td>
<td>0.936</td>
</tr>
<tr>
<td>C5</td>
<td>0.549</td>
<td>0.704</td>
<td>0.935</td>
<td>0.665</td>
<td>1</td>
<td>2.152</td>
<td>0.895</td>
<td>0.141</td>
<td>0.856</td>
</tr>
<tr>
<td>C6</td>
<td>0.452</td>
<td>0.418</td>
<td>0.514</td>
<td>0.594</td>
<td>0.464</td>
<td>1</td>
<td>0.544</td>
<td>0.085</td>
<td>0.521</td>
</tr>
<tr>
<td>Total</td>
<td>6.350</td>
<td>1.000</td>
<td>37.205</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2: Hierarchy for evaluating software reusability

B. Determine Eigenvector and Eigen-Value: In order to determine the value of Eigen vector, following five steps are followed:

Step 1: Determine Nth root of product by multiplying all the row values. For C1, \( (1*3.750*1.499*1.252*1.821*2.210)^{1/6} = 1.746 \). Similarly, determine Nth root of product of values for other factors. The values we get are 1.140, 1.003, 1.022, 0.895 and 0.544. Summation of these values is 6.35.

Step 2: Determine eigen vector (w) by dividing the Nth root of product by sum of Nth root of product of values. For C1, \( (1.746/6.35) = 0.275 \). Similarly, calculate the eigen vector value designed for other criteria. The values we get are 0.179, 0.158, 0.161, 0.141 and 0.086. According to Saaty rule, the sum of these eigen vector should be 1.00. Therefore the comparison of values for the criteria is correct for the calculated values.

Step 3: To make sure if the analysis went correct or not? We determine eigen value (Aw) by multiply row values of the factor with the column values of Eigen vector (w). For C1, \( (1*0.275 + 3.750*0.179 + 1.499*0.158 + 1.252*0.161 + 1.821*0.141 + 2.210*0.544) = 1.829 \).

Similarly, calculate eigen values for other criteria and get the values 1.158, 0.971, 0.936, 0.856 and 0.521.

Step 4: Estimate \( \lambda \). For a consistent matrix, \( \lambda_{\text{max}} >= n \). Here, \( n=6 \) hence \( \lambda_{\text{max}} >= 6 \) where \( \lambda_{\text{max}} \) is mean of \( \lambda \) values. For C1, \( \lambda = 1.829/0.275=6.650 \).

Similarly, we get values given below (6.650+6.469+6.145+5.813+6.070+6.058)/6= 6.200 > 6 for characteristics. Based on the calculated values, our matrix is consistent.

Step 5: Calculate CI and CR. The CR must be less than 0.1.

Here CI is 0.04.

To calculate Consistency Ratio we take the random judgment given in Table 2 derived by Saaty [17].

\[
\text{CR} = \frac{\text{CI}}{\text{RI}} \text{(value from the below Table 2)} = 0.04/1.24 = 0.032 < 0.1
\]

Hence, judgments are acceptable and consistent. All calculations are also shown in Table 3.
Table 4: Brief description about projects [30]

<table>
<thead>
<tr>
<th>Project name</th>
<th>Metrics</th>
<th>Size and complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Soc Cohesion Coupling</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CDC CDO LCOO CBC DIT VS</td>
<td>LOC NOA WOC</td>
</tr>
<tr>
<td>Spacewar (P1)</td>
<td>0 0 0.25 2 29 21 1472 50</td>
<td>196</td>
</tr>
<tr>
<td>Telecom (P2)</td>
<td>0 0 1.07 5 10 7 105 6 30</td>
<td></td>
</tr>
<tr>
<td>Tracing (P3)</td>
<td>0 0 0.58 2 18 13 261 21 94</td>
<td></td>
</tr>
</tbody>
</table>

D. Evaluate quality of AO projects: Now we have applied the proposed model on three AspectJ Projects. We named these three AspectJ projects P1, P2, P3. We have pair wise comparisons for P1, P2 and P3 in terms of the six criteria, C1, C2, C3, C4, C5 and C6.

All these three project are AspectJ programs and their details are mentioned in Table 4. For the selected AspectJ projects, we have shown their metrics values which are gathered from AOP Metric. This tool is capable to plug-in with eclipse and Java 1.5 enabled. Three software named, Spacewar, Tracing and Telecom are open source code available on sourceforge.net. Spacewar is having the highest number of line of code. Considered projects were taken to assess their reusability based on proposed scheme. Further, real life projects can also be evaluated and cross validated against the proposed model. Here the prime motivation is to shown the applicability of the proposed scheme, irrespective of the size of the considered project.

Relative values for C1 (Understandably) to rank the three projects is given below. The Eigen vector for this matrix is (0.64, 0.10, 0.26), very much as expected and the CR is 0.001, so the judgments are acceptable and consistent as required for the used approach proposed by Saaty [17].

Understandability

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>P2</td>
<td>1/5</td>
<td>1</td>
<td>1/3</td>
</tr>
<tr>
<td>P3</td>
<td>1/3</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

The values for the next five characteristics for the considered projects are shown below from C2 to C6 respectively along with their Eigen Vector and CR values.

Learn-ability

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>P2</td>
<td>1/5</td>
<td>1</td>
<td>1/5</td>
</tr>
<tr>
<td>P3</td>
<td>1/3</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Adaptability:

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>P2</td>
<td>1/7</td>
<td>1</td>
<td>1/5</td>
</tr>
<tr>
<td>P3</td>
<td>1/5</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Eigen vector (0.73, 0.08, 0.19), CR=0.03

Operability:

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>P2</td>
<td>1/7</td>
<td>1</td>
<td>1/5</td>
</tr>
<tr>
<td>P3</td>
<td>1/3</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Eigen vector (0.65, 0.07, 0.28), CR=0.03

Maintainability:

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>3</td>
<td>1/3</td>
</tr>
<tr>
<td>P2</td>
<td>1/3</td>
<td>1</td>
<td>1/5</td>
</tr>
<tr>
<td>P3</td>
<td>3</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Eigen vector (0.26, 0.10, 0.64), CR=0.01

Modularity:

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>P2</td>
<td>1/5</td>
<td>1</td>
<td>1/5</td>
</tr>
<tr>
<td>P3</td>
<td>1/3</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Eigen vector (0.61, 0.09, 0.30), CR=0.06

In final stage we construct a matrix of the eigenvectors for P1, P2 and P3 as mentioned below.

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
</tr>
</thead>
<tbody>
<tr>
<td>W_i</td>
<td>0.275</td>
<td>0.179</td>
<td>0.158</td>
<td>0.162</td>
<td>0.141</td>
<td>0.085</td>
</tr>
<tr>
<td>P1</td>
<td>0.64</td>
<td>0.62</td>
<td>0.73</td>
<td>0.65</td>
<td>0.26</td>
<td>0.61</td>
</tr>
<tr>
<td>P2</td>
<td>0.10</td>
<td>0.09</td>
<td>0.08</td>
<td>0.07</td>
<td>0.10</td>
<td>0.09</td>
</tr>
<tr>
<td>P3</td>
<td>0.26</td>
<td>0.29</td>
<td>0.19</td>
<td>0.28</td>
<td>0.64</td>
<td>0.30</td>
</tr>
</tbody>
</table>

AO Project Reusability Index (RI)

\[
RI = \sum \text{Weight Value of } P_i \times \text{Weight Value of } C_i
\]

where

\( n \) = Number of characteristics.

\( C_i \) = Characteristics \( i \)

\( P_i \) = Projects \( i \)
Table 5: Quality of AO projects by AHP

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>RI</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>0.275</td>
<td>0.179</td>
<td>0.158</td>
<td>0.162</td>
<td>0.141</td>
<td>0.085</td>
<td></td>
</tr>
<tr>
<td>P1</td>
<td>0.64</td>
<td>0.62</td>
<td>0.73</td>
<td>0.65</td>
<td>0.26</td>
<td>0.61</td>
<td>1.286</td>
</tr>
<tr>
<td>P2</td>
<td>0.10</td>
<td>0.09</td>
<td>0.08</td>
<td>0.07</td>
<td>0.10</td>
<td>0.09</td>
<td>0.28</td>
</tr>
<tr>
<td>P3</td>
<td>0.26</td>
<td>0.29</td>
<td>0.19</td>
<td>0.28</td>
<td>0.64</td>
<td>0.30</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Our prime objective is to estimate the quality of considered aspect oriented projects, we have empirical calculated their values as reported in Table 5. Project 1 is better in terms of quality as compare to other two projects in respect to Reusability Index (RI), whose value is 1.286 for P1 and minimum (0.28) for P2. In actual situation, comparative values of characteristics can be gathered from running projects, which are developed using AO technology. The projects, which are compared here, are medium size projects. However, our motive is to show the applicability of proposed scheme for the reusability estimation of Aspect Oriented Software. Proposed schemes can be applied on real life software based on the values of identified six characteristics and it will determine the Reusability Index (RI) for the considered software. It can be applied on each module (class, aspect) in order to know their reusability or it can also be applied on whole developed system to know its overall reusability. In next section, FAHP is applied on the same projects in place of AHP to cross validate the estimated results by AHP.

EVALUATION OF REUSABILITY USING FUZZY ANALYTICAL HIERARCHY PROCESS (FAHP)

AHP use only pair wise matrix to determine the indistinctness in multi criteria decision making problems. In FAHP, fuzzy pair-wise comparisons are done to make decision process to construct the reciprocal decision matrix. In this way, qualitative data can be transformed into crisp ratios. We have applied the FAHP on all three projects considered for AHP. Implementation of the FAHP is mentioned below for the considered projects.

a. Triangular fuzzy numbers: A fuzzy number with triangular membership function A is characterized by (a1, a2, a3), a1<a2<a3 as shown in figure 3. Its membership function qualities are written in equation 1[26].

\[
\mu_A(x) = \begin{cases} 
\frac{x-a_1}{a_2-a_1}, & a_1 \leq x \leq a_2 \\
\frac{a_3-x}{a_3-a_2}, & a_2 < x \leq a_3 \\
0, & \text{otherwise}
\end{cases}
\]  

(1)

Table 6: Fuzzy numbers for qualitative assessments [26]

<table>
<thead>
<tr>
<th>Fuzzy numbers</th>
<th>Membership function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,1,3)</td>
</tr>
<tr>
<td>X</td>
<td>(x-2, x, x+2) for x=3, 5, 7</td>
</tr>
<tr>
<td>9</td>
<td>(7, 9, 11)</td>
</tr>
</tbody>
</table>

Fig. 3: Graph for triangular membership function

b. Fuzzy Synthetic Extent Analysis: Fuzzy synthetic extent analysis helps to determine the fuzzy value of Saaty [17] scale which will estimate the quality of considered projects.

\[
S_j = \sum_{i=1}^{n} \frac{w_j a_{ij}}{\sum_{i=1}^{n} w_j a_{ij}}
\]  

(2)

Here, we describe the algorithm in ten steps to apply fuzzy AHP for evaluating the best quality of project in terms of reusability as mentioned below; however the reference of calculation and formulas is based on the paper given by Deng [26].

**Step 1:** First construct the data by using AHP process and fuzzy synthesis analysis and then construct the hierarchical structure for evaluation of quality of AO Projects.

**Step 2:** Construct Decision Matrix (X) as shown in equation (6) by using equation (3) to equation (5) [26].

\[
C=I-W= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\
\vdots & \ddots & \vdots & \vdots \\
a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix}
\]  

(3)

Where

\[
a_{ij} = \begin{cases} 
1, & i = j \\
\frac{i}{s}, & i < s \\
\frac{s}{i}, & i > s
\end{cases}
\]  

(4)

With the help of fuzzy extent analysis of equation (3) and using equation (1) and (2), the corresponding weights (w_j) with respect to specific criterion C_j can be determined as
\[ X_{ij} \text{ or } w_j = \frac{\sum_{k=1}^{m} a_{ik}}{\sum_{k=1}^{m} a_{kj}} \]  

(5)

\[
X = \begin{bmatrix}
    a_{11} & a_{12} & \cdots & a_{1n} \\
    a_{21} & a_{22} & \cdots & a_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}
\]

(6)

\[
X = \begin{bmatrix}
    (0.18,0.60,1.79) & (0.17,0.53,1.43) & (0.28,0.69,1.52) & (0.22,0.58,1.36) & (0.07,0.29,1.05) & (0.17,0.53,1.43) \\
    (0.04,0.10,0.52) & (0.04,0.08,0.34) & (0.04,0.08,0.34) & (0.03,0.07,0.28) & (0.04,0.10,0.50) & (0.04,0.08,0.34) \\
    (0.07,0.29,1.08) & (0.14,0.37,1.04) & (0.06,0.22,0.67) & (0.13,0.33,0.88) & (0.17,0.60,1.75) & (0.14,0.37,1.04)
\end{bmatrix}
\]

Step 3: Measure the weighting vector from equation (7) using equations (3) to (5).

\[
W = (w_1, w_2, \ldots, w_m)
\]

(7)

\[
W_1 = (0.11, 0.33, 0.91) \\
W_2 = (0.06, 0.20, 0.91) \\
W_3 = (0.06, 0.21, 0.65) \\
W_4 = (0.01, 0.02, 0.13) \\
W_5 = (0.02, 0.07, 0.26) \\
W_6 = (0.04, 0.14, 0.45)
\]

Step 4: Construct the fuzzy performance matrix as shown in equation (8).

\[
Z = \begin{bmatrix}
    w_1 f_1 & w_1 f_2 & \cdots & w_1 f_n \\
    w_2 f_1 & w_2 f_2 & \cdots & w_2 f_n \\
    \vdots & \vdots & \ddots & \vdots \\
    w_n f_1 & w_n f_2 & \cdots & w_n f_n
\end{bmatrix}
\]

(8)

\[
Z = \begin{bmatrix}
    (0.019,0.198,1.62) & (0.010,0.106,0.82) & (0.016,0.144,0.98) & (0.002,0.011,0.176) & (0.001,0.002,0.273) & (0.006,0.074,0.643) \\
    (0.004,0.033,0.47) & (0.002,0.016,0.197) & (0.002,0.01,0.221) & (0.0003,0.0001,0.036) & (0.0008,0.007,0.13) & (0.001,0.011,0.153) \\
    (0.007,0.09,0.98) & (0.008,0.074,0.603) & (0.003,0.04,0.435) & (0.001,0.006,0.114) & (0.003,0.042,0.455) & (0.005,0.05,0.468)
\end{bmatrix}
\]

The \( \alpha \)-cut on the performance matrix as given in equation (8) can also be used to determine interval performance matrix equation (9). Here \( \alpha \) show the decision making extent of assurance in fuzzy assessments regarding projects as well as criteria. The higher value of \( \alpha \) means further confident decision making (DM).

Step 5: Construct the interval performance matrix as given in equation (9) by means of \( \alpha \)-cut scheduled for performance matrix.

\[
Z_{\alpha} = \begin{bmatrix}
    [Z_{11}^\alpha, Z_{11}^{\alpha}] & [Z_{21}^\alpha, Z_{21}^{\alpha}] & \cdots & [Z_{m1}^\alpha, Z_{m1}^{\alpha}] \\
    [Z_{12}^\alpha, Z_{12}^{\alpha}] & [Z_{22}^\alpha, Z_{22}^{\alpha}] & \cdots & [Z_{m2}^\alpha, Z_{m2}^{\alpha}] \\
    \vdots & \vdots & \ddots & \vdots \\
    [Z_{1n}^\alpha, Z_{1n}^{\alpha}] & [Z_{2n}^\alpha, Z_{2n}^{\alpha}] & \cdots & [Z_{mn}^\alpha, Z_{mn}^{\alpha}]
\end{bmatrix}
\]

(9)
Step 6: Construct the crisp performance matrix [26] as given in equation (10)

\[ Z^i_{ij} = \frac{z^i_{ij}}{\sqrt{\sum \alpha \lambda (Z^i_{ij} \cdot Z^i_{min})}} \]  

Step 7: Determine the normalized performance matrix as shown in equation (12) using equation (11).

\[ Z^i_{ij} = \frac{Z^i_{ij}}{\sqrt{\sum \alpha \lambda (Z^i_{ij} \cdot Z^i_{min})}} \]  

Step 8: Determine the ideal and negative ideal values for every criterion from normal performance matrix using equation (13) and equation (14).

\[ A^+ = (z^i_{12} \cdot z^i_{22} \cdot z^i_{32} \cdot \ldots \cdot z^i_{n2}) \]  

\[ A^- = (z^i_{11} \cdot z^i_{21} \cdot z^i_{31} \cdot \ldots \cdot z^i_{n1}) \]  

Here

\[ Z^+_{\text{max}} = \max (z^i_{1i}, z^i_{2i}, z^i_{3i}, \ldots, z^i_{ni}) \]  

\[ Z^-_{\text{min}} = \min (z^i_{1i}, z^i_{2i}, z^i_{3i}, \ldots, z^i_{ni}) \]  

Step 9: Identify the extent of resemblance among each project and the ideal as well as negative ideal values via equation (15) and equation (16)

\[ S^+_{\text{max}} = \frac{A^+}{\max (A^+, A^-)} \]  

\[ S^-_{\text{max}} = \frac{A^-}{\max (A^+, A^-)} \]  

Here, the higher value of positive ideal and lower value of negative ideal is desirable for estimation and shows the larger degree of resemblance among each project.

Step 10: Identify the entire performance index for every project via equation (17) and rank the each project in according to performance index values as shown in Table 7.

\[ P^i_{\alpha} = \frac{z^i_{\alpha \text{max}}}{z^i_{\alpha \text{max}} + z^i_{\alpha \text{min}}} \]  

Finally, the ranking of the considered AO projects have been estimated using both AHP and FAHP and both have shown the same result.

### CONCLUSION AND FUTURE SCOPE

In this paper we have proposed AO software reusability quality model and also shown the relationship among external attributes and its related metrics. This model has considered six qualities attribute namely understandability, learn-ability, adaptability, operability, maintainability and modularity based on the conducted survey and reported literature. In order to evaluate reusability using the proposed model as a single unit, Analytic Hierarchy Process (AHP) and FAHP techniques are applied. The survey on 36 participant have been carried out and value of pair-wise relative weights for the characteristics is taken. The mean of the collected samples have been considered as pair-wise relative weights. Reusability of three AO projects has been compared using these weights. This proposed reusability model can be used to estimate the reusability of selected projects developed in AO language.

This AHP method is majorly used for handling various MCDM problems in real situations. In spite of its easiness in conception and competence in calculation, it has various shortcomings. To improve the AHP method, this paper presents a fuzzy logic based discrete MCDM method for solving the wide-range of Multi Analysis decision problem involving qualitative data. In future, overall quality estimation model for AO software can be developed using other MCDM methods. Later, these models can be cross validated using other techniques and help the software professional in estimating the quality of software at hand. It will also help in reducing the developing and maintenance efforts and used an indicator of quality characteristics for the analyzed software.
ACKNOWLEDGMENT

We would like to express our sincere gratitude toward the faculties of Amity University and Gautam Buddha University for helping us in refine the objectives and both universities for providing us research environment and facilities.

REFERENCES


