Software Architecture Evaluation using Multivariate Statistical Analysis

Dr. G. Zayaraz
Associate Professor
Department of CSE, Pondicherry Engg. College, Puducherry, India

J. Syed Ibrahim Shah
Technical Consultant and Business Analyst, Infosys, Chennai, India

C. Dhaya
Research Scholar
Department of CSE, Pondicherry Engg. College, Puducherry, India

Dr. V. Vijayalakshmi
Assistant Professor
Department of ECE, Pondicherry Engg. College, Puducherry, India

ABSTRACT

Organizations often need to choose Software Architecture for future development from several competing candidate architectures. The various stakeholders' quality requirements need to be considered collectively to describe the quality requirements of the envisioned system and therefore build the basis for the comparison and selection criteria. The involvement of many stakeholders with different preferences of quality requirements poses a challenge for evaluation. In this paper, Multivariate Statistical Analysis approach has been employed to model the differences in preferences of the quality requirements of the stakeholders. Based on this model the different candidate architectures are evaluated for the conformance to the stakeholders' quality requirements.

Keywords: Software Architecture; Quality Requirements; Multivariate Statistical Analysis;

1. INTRODUCTION

Quantitative evaluation techniques employ quantitative framework [1] or metrics to select an appropriate architecture from the given set of alternatives. Using metric value to assess architectures not only depends on the evaluation approach used but also on the validity of the assumptions on architecture structures [2]. Architectural metrics for measuring cohesion, complexity and coupling through COSMIC Full Function Points is given in [3]. The other ways of quantitative architecture evaluation is by employing comparison frameworks like Analytic Hierarchy Process (AHP) [4] or by using Hypothetical Equivalents [5].

Multivariate statistics provide the ability to analyze complex sets of data. It provides statistics for analysis of independent and possible dependent variables. Software architecture evaluation which involves many stakeholders with different preferences of quality requirements can be modeled naturally with multivariate statistical analysis [6]. This paper focuses on proposing a new evaluation framework using multivariate statistical analysis. In this section a brief introduction on software architecture evaluation has been presented. Section 2 briefly reviews the existing quantitative techniques. Section 3 lists the limitations of existing quantitative techniques which is the motivation behind this work. Section 4 discusses on the proposed evaluation framework and validates it through a case study. Section 5 concludes the paper by highlighting the benefits of the proposed evaluation technique.

2. RELATED WORK

A quantitative framework and models using AHP for comparison of different candidate architectures for a specific quality attribute and vice versa is proposed in [7]. A quantitative selection framework based on multi-attribute decision making using Hypothetical Equivalents is proposed in [8]. This framework provides a way for an architecture selection process by comparing the fitness of competing candidate architectures based on the quality requirements of different stakeholders

3. MOTIVATION

The captivating benefits of quantitative techniques over scenario based techniques have been the main motivation behind this work. Although, the results of existing methods are quantitative, they failed to provide enough statistical evidence for the choices being made.

The use of multivariate statistics in analyzing complex data sets have laid the foundation for developing an evaluation framework based on multivariate statistical analysis.

4. PROPOSED QUANTITATIVE FRAMEWORK

In order to overcome the limitations discussed in section 3, a new evaluation framework based on multivariate statistical analysis is proposed. This evaluation framework adopts the properties of multivariate normal distribution to model variation in preferences according to changes in requirements. Changes in preferences with requirements are modeled into multivariate distribution. Inputs for the evaluation framework are set of quality attributes and set of alternative architectures along with its quality characteristics measured. A case study of real-time stock monitoring system [10] has been used to demonstrate the usage of the proposed evaluation framework.

The primary goal of this real-time stock monitoring system is to capture, analyze and broadcast stock events information in real-time. It is a soft real-time system where some of the events may miss their deadline without affecting the whole system behavior. The system is a real-time data provider for monitoring stocks of small and medium size stock exchanges for brokers and independent investors. An antenna (feed server), external to the system, provides the data (feed) to the data server. A feed contains the relevant information of a stock exchange transaction. Feeds are supposed to be reliable and available. The clients namely the brokers are distributed in different geographical locations are subscribed to the data server. When a change on the feed to which a client has

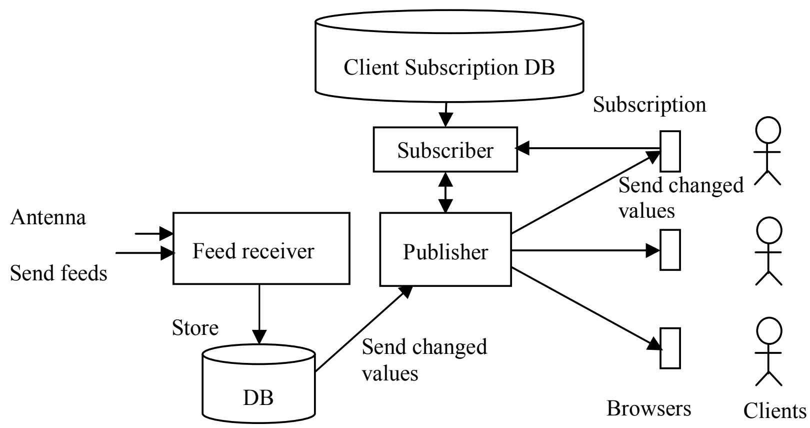

Figure 1: Publisher/Subscriber Pattern (A)

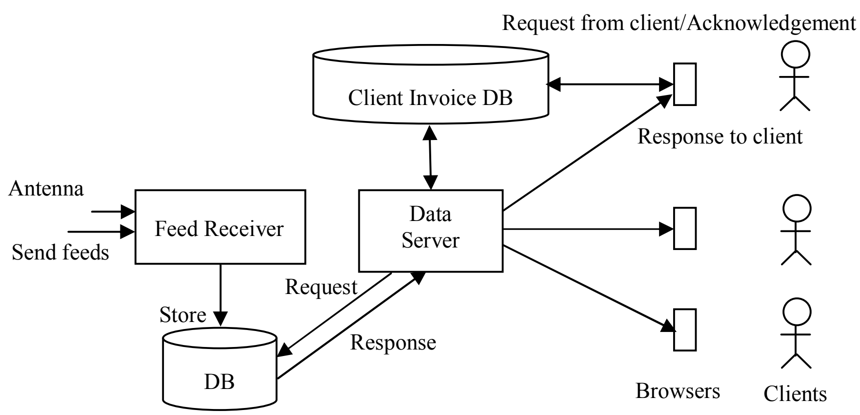

Figure 2: Repository Pattern (B)

has subscribed occurs, the feed is broadcasted to him/her by the data server according to a strict time delay. The time delay will depend on the network structure used to send the information to the clients. The type of service offered depends on this delay. Internet facilities through commercial browsers are required for the system. The Publisher/Subscriber stores the client subscriptions, the actual values in the client subscription DB and the data server respectively.

4.1 Types of Patterns

Three different architectural solutions are available for real-time stock monitoring system namely: Publisher/Subscriber Pattern, Repository Pattern and Broadcast Pattern. Quality attributes considered for evaluation are Response Time, Learnability, Recoverability, Reusability, Cost, Team Size, Maintainability and Development Time.

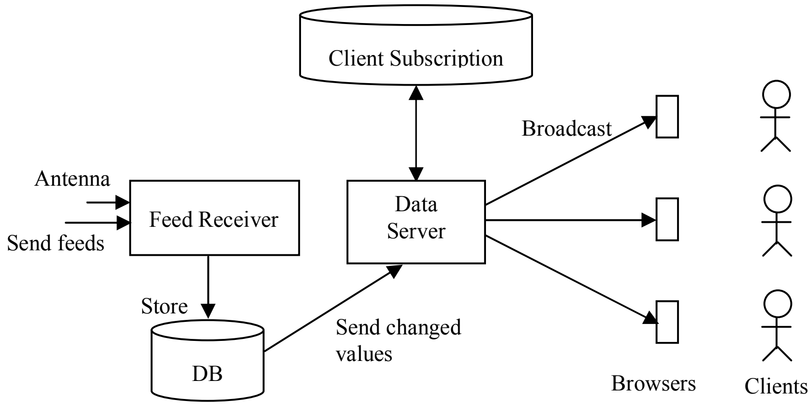

Figure 3: Broadcast Pattern (C)

4.1.1 Publisher/Subscriber Pattern (A)

In this type of candidate architecture, clients register their interest for stocks with the subscriber. The subscriber records the details of the clients in the database. A change in stock prices causes the publisher to notify these changes to the interested clients. Publisher/Subscriber pattern is shown in Figure 1.

4.1.2 Repository Pattern (B)

In this type of candidate architecture, clients request the server for data about the stocks. Requests by clients may or may not be done periodically. Usage of proper queuing mechanism helps to avoid conflicts among requesting clients. This pattern is shown in Figure 2.

4.1.3 Broadcast Pattern (C)

In this type of candidate architecture, a change in stock prices causes the server to broadcast these changes to their clients. Communication between clients and server is unidirectional. This pattern is shown in Figure 3.

Candidate architectures along with quality characteristics measured are taken as input to the proposed evaluation framework. How these quality characteristics are measured is beyond the scope of this paper. Measured values of candidate architectures are listed in Table 1[1].

4.2 Steps in Evaluation Framework

Evaluation process has to be systematic and easily understood by participants. Considering these aspects the proposed evaluation framework is presented in six steps. These steps facilitate the evaluators to choose the right architecture. The series of steps in the evaluation framework are presented as follows:

i. *Identification of stakeholders*
ii. *Identification of quality requirement ranges and preferences*
iii. *Determination of preference co-efficients*
iv. *Determination of realization values*
v. *Determination of rational values and*
vi. *Selection of architecture.*

Table 1: Measured Values

Architecture	Response Time (milliseconds)	Maintainability (hours)	Learnability (hours)	Reusability (numbers)	Recoverability (seconds)	Cost (Rupees in Lacs)	Team size (numbers)	Development Time (weeks)
A	10	20	5	1	65	8	20	60
B	20	10	8	5	20	4	10	30
C	12	5	3	1	70	6	5	20

4.2.1 Identification of Stakeholders

Identifying stakeholders and classifying them is done using segmentation [11]. Segmentation refers to grouping of people based on similar characteristics. Participating stakeholders are given the choice for selecting their relevant quality attributes. Stakeholders and their associated quality attributes for the case study considered are tabulated below in Table 2.

4.2.2 Identification of Quality Requirement Ranges and Preferences

Existing requirement gathering techniques viz., WinWin requirement gathering method [12] and Goal-oriented requirement gathering method [13] can be employed to identify the stakeholders' quality requirements. Preference rating of each quality attributes are identified from individual stakeholders.

Table 2: Stakeholders and their Associated Quality Attributes

Stakeholder	Quality Attributes
Broker	Response time, Learnability, Recoverability and Cost
Small investor	Response time, Learnability, Recoverability and Cost
Technical person	Response time, Maintainability and Reusability
Manager	Cost, Team Size and Development Time

Table 3: Quality Requirements of the Stakeholders

Role	Quality Attributes	Preferences	Min. Range	Max. Range	Orientation
Broker	Response time (ms)	6.5	8	20	10
	Learnability (hrs)	7.0	3	6	-
	Recoverability (secs)	6.0	10	50	20
	Cost (rs in Lacs)	8.0	6	10	-
Small investor	Response time (ms)	8.0	15	25	-
	Learnability (hrs)	7.5	5	10	6
	Recoverability (secs)	7.0	30	60	40
	Cost (rs in Lacs)	8.5	3	6	4
Technical person	Response time (ms)	9.0	10	30	10
	Maintainability (hrs)	8.5	5	20	12
	Reusability (nos)	7.0	2	8	-
Manager	Cost (rs in Lacs)	8.5	4	7	4
	Team size (nos)	8.0	5	20	8
	Development time (wks)	6.5	30	60	35

Ratings may range from 0 to 10. When more than one stakeholder participates for the same quality attribute then the peak preference *P* of that quality attribute is computed as the average of all preferences. Therefore a quality attribute consists of a peak preference and a requirement range. Quality requirements for the case study considered are listed in Table 3.

Orientation refers to the most preferred value in the specified range of requirements. From the requirements gathered from stakeholders each quality attribute is identified with its minimum range, maximum range, peak preference and median. If more than one stakeholder is participating for the same quality attribute then minimum range is the minimum value of all the requirements and maximum range is the maximum of all the requirements for that quality attribute. Similarly, the peak preference of a quality attribute is taken as the average of all the preferences of stakeholders participating for that quality attribute. Median is the mid-point of the requirement range. If the stakeholder specifies orientation then the orientation is assigned to median. If there is more than one stakeholder participating for a quality attribute then median of that particular quality attribute is taken as average of all medians.

Similarly peak preference for a quality attribute having more than one stakeholder participating is given by taking average of preferences. Peak preference of response time is 7.83. Table 4 summarizes the minimum range, maximum range, peak Preference and median of response time

From Table 4 it is obvious that response time has the requirement range of 8 milliseconds to 30 milliseconds. Its peak preference is 7.83 (average of all preferences) and median is 13.33(average of all medians). Similarly the requirement range, peak preference and median of other quality attributes are computed and listed in Table 5.

4.2.3 Determination of Preference Co-efficients

The next step in evaluation is to determine the preference co-efficients. Preference co-efficient is the ratio of actual preference *p* to peak preference *P*. Determination of the actual preference value *p* of each measured value of quality attributes requires the mapping to multivariate curve. For that the distribution value *u* is determined using (1). The value 10 in the denominator is the value of maximum preference with the same unit of measure as the measured values. Therefore, the distribution value *u* is unit less.

Table 4: Requirement Range, Preference and Median of Response Time

Stakeholder	Response time in milliseconds		Preference	Median
	Min. Range	Max. Range		
Broker	8	20	6.5	10(orientation)
Small investor	15	25	8.0	20
Technical person	10	30	9.0	10(orientation)

For response time there are three stakeholder groups participating namely broker, small investor and technical person. The requirement range of broker for response time is 8 milliseconds to 20 milliseconds, the small investor requirement range for response time is 15 milliseconds to 25 milliseconds and technical person requirement range for response time is 10 milliseconds to 30 milliseconds. The minimum requirement is 8 milliseconds and maximum requirement is 30 milliseconds, therefore the requirement range of response time is 8 milliseconds to 30 milliseconds. Median of response time is computed by taking the average of medians of requirement range of all participating stakeholders. Median for response time of broker requirement is 10(orientation), median for response time of small investor requirement is 20 and that of technical person is 10(orientation). Therefore, average of medians of all stakeholders' requirement range for response time gives the median for the response time.

$$u = (x - \mu) / 10 \quad \text{.................. (1)}$$

where x is the value in the given range of requirements, μ is the median (orientation or midpoint of requirement range) and *10* is the value of maximum preference in the same unit of measure as x or μ.

Distribution value u of response time for candidate architecture A is computed as follows: value of x = 10 (Table 1) and value of μ = 13.33 (Table 5).

$$u = (10 - 13.33) / 10 \quad \text{..........using (1)}$$

$$u = -0.33$$

The change in preference in accordance with the change in requirements is given by *f (u)* in (2).

$$f(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \quad \text{............... (2)}$$

where u is the distribution value of the quality requirement in the given range.

At peak preference, since the value of x is equal to the value of μ, the value of u becomes zero. The value of *f(u)* at peak preference is given by (3.3)

$$f(u) = \frac{1}{\sqrt{2\pi}}, \because e^{-\frac{u^2}{2}} = 1 \quad \text{............. (3)}$$

Actual preference p of response time for candidate architecture A is computed as follows: value of P=7.83 (Table 5) and value of u = -0.33.

$$p = 7.83 * \left(e^{\frac{-(-0.33)^2}{2}} \right) \quad \text{...using (4)}$$

$$p = 7.4$$

Table 5: Requirement Range, Peak Preference and Median of Quality Attributes

Quality Attributes	Min. Range	Max. Range	Peak Preference P	Median μ
Response time (in ms)	8	30	7.83	13.33
Maintainability (in hrs)	5	20	8.5	12
Learnability (in hrs)	3	10	7.25	5.25
Reusability (in nos)	2	8	7.0	5
Recoverability (in secs)	10	60	6.5	30
Cost (rs in lacs)	3	10	8.33	5.33
Team size (in nos)	5	20	8.0	8
Development time (in wks)	30	60	6.5	35

When $x = \mu$, the value of *f(u)* becomes $1/\sqrt{2\pi}$ and actual preference p is P. when $x \neq \mu$, the value of p is determined by applying rule of three. For any given instance of value in the range of requirements, corresponding preference value p is given by (4).

$$p = P\left(e^{-\frac{u^2}{2}}\right) \quad \ldots\ldots\ldots\ldots\ldots (4)$$

where P is the peak preference value and u is the distribution value of requirement in given range.

Preference co-efficient P_c is obtained by computing the ratio of actual preference p to that of peak preference P and is given by (5).

$$P_c = \frac{p}{P} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

where P is the peak preference value and p is the actual preference value.

Preference co-efficient P_c of response time for candidate architecture A is computed as follows: value of p = 7.4 and value of P = 7.83.

$$P_c = \frac{7.4}{7.83} \quad \ldots\ldots\ldots\ldots \text{using (5)}$$

$$P_c = 0.95$$

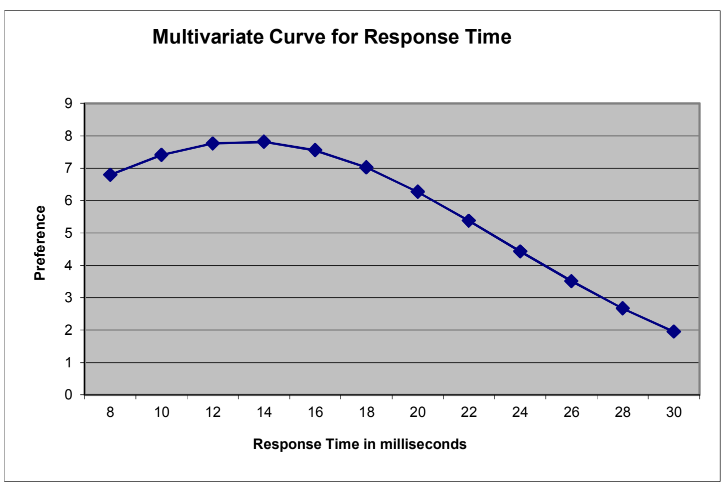

Figure 4: Multivariate Curve for Response Time

Quality Attributes	Actual Preference p			Preference Co-efficient P_c		
	A	B	C	A	B	C
Response time	7.40	6.27	7.76	0.95	0.80	0.99
Maintainability	6.17	8.33	6.65	0.73	0.98	0.78
Learnability	7.25	6.98	7.07	0.99	0.96	0.98
Reusability	6.46	7.00	6.46	0.92	1	0.92
Recoverability	0.01	3.94	0.002	0.002	0.61	0.0003
Cost	8.04	8.26	8.31	0.96	0.99	0.99
Team size	3.89	7.84	7.65	0.49	0.98	0.96
Development time	0.29	5.74	2.11	0.04	0.88	0.32

Table 6: Actual Preferences and Preference Co-efficients

Similarly actual preference for all the values in the range of requirements is computed and multivariate curve is constructed. This curve reflects the trend in preferences for quality attributes and this serves as a statistical evidence for the choices being made. There will be eight multivariate curves, one for each quality attribute. Multivariate curve for response time is constructed by determining the actual preference for all the values between 8 milliseconds and 30 milliseconds. Figure 4 shows the multivariate curve of response time.

Preference co-efficient of response time for candidate architecture (A) is 0.95. These preference co-efficient shows how close the measured value is to the preference of stakeholder. Preference co-efficient of other candidate architectures (B) and (C) is 0.8 and 0.99 respectively. From the multivariate curve it can be clearly seen that architecture (C) has high preference co-efficient which is close to one. Preference co-efficient for all measured values of three candidate architectures are computed in Table 6.

4.2.4 Determination of Realization Values

Realization value R_a of a particular architecture is given by density of distribution under the multivariate curve. Realization value determines the flexibility in the architecture. Lesser the realization value more is the architecture adaptable for future changes. Density of distribution under the curve is given by the area under the multivariate curve from minimum requirement to the measured value and is given by (6).

$$R_a = \int_{v_{min}}^{m} f(u)\,du \quad \text{............ (6)}$$

where f(u) is the function of u given by (2), m is the distribution value of measured quantity of an architecture and v_{min} is the distribution value of the minimum instance of requirement range.Realization value of response time for candidate architecture (A) is computed as follows: value of m = -0.33 and value of v_{min} = -0.53.

$$R_a = \frac{1}{\sqrt{2\pi}} \int_{-0.53}^{-0.33} e^{\frac{-u^2}{2}}\,du \quad \text{..........using (6)}$$

$$R_a = 0.075$$

Realization value of response time for candidate architecture (A) is 0.075, whereas for architecture (B) and (C) it is 0.38 and 0.15 respectively. Candidate architecture (A) has more flexibility when compared to others. Realization value for all the measured values of candidate architectures are listed in Table 7.

4.2.5 Determination of Rational Values

Rational value gives the conformance of a quality attribute to the requirement preference of stakeholders. If the actual preference p equals peak preference P, then the value of preference co-efficient P_c becomes one. Similarly the realization value R_a will be nearly zero when the measured value is very close to the minimum value. The product of preference co-efficient and (1-R_a) gives the rational value of a quality attribute of a particular architecture structure. The maximum value of λ is one and the minimum value is not less than zero. Rational value λ of a quality attribute of a particular architecture structure is given by (7).

$$\lambda = P_c\left(1 - R_a\right) \quad \text{................... (7)}$$

where P_c is the preference co-efficient and R_a is the realization value of a quality attribute. Rational value of response time for candidate architecture (A) is computed as follows: value of P_c=0.95 (Table 6) and value of R_a = 0.075(Table 7)

$$\lambda = 0.95\left(1 - 0.075\right) \quad \text{.........using (7)}$$

$$\lambda = 0.875$$

Rational value of response time for candidate architecture (A) is 0.875; similarly rational values for all the measured values of candidate architecture are computed using (7) and listed in Table 8.

Table 7: Realization Value

Quality Attributes	Realization value R_a		
	A	B	C
Response time	0.075	0.383	0.158
Maintainability	0.434	0.196	0
Learnability	0.079	0.192	0
Reusability	0.258	0.119	0.258
Recoverability	0.004	0.242	0.001
Cost	0.192	0.039	0.119
Team size	0.291	0.196	0
Development time	0.053	0	0.129

4.2.6 Selection of Architecture

Total satisfaction value of candidate architecture is the summation of rational values of all the quality attributes. It is given by (8).

$$\text{Total satisfaction value} = \sum_{i=1}^{n} \lambda \quad \text{………} (8)$$

Table 8: Rational Values

Quality Attributes	Rational Value λ		
	A	B	C
Response time	0.875	0.494	0.834
Maintainability	0.410	0.789	0.783
Learnability	0.919	0.778	0.975
Reusability	0.685	0.88	0.685
Recoverability	0.002	0.459	0.003
Cost	0.779	0.952	0.879
Team size	0.345	0.789	0.956
Development time	0.042	0.882	0.283

By summation of rational values of all the quality attributes, the total satisfaction value of candidate architectures A, B and C are obtained and listed in Table 9. It is obvious that candidate architecture (B) has the highest total satisfaction value 6.023 with 75.29% as the percentage of conformance. Therefore, the repository pattern (B) is chosen as the appropriate architecture for Stock monitoring system.

Table 9: Percentage of Conformance to Requirements

Candidate Architectures	Total Satisfaction Value	Percentage of Conformance to Requirements
A	4.058	50.73 %
B	6.023	75.29 %
C	5.394	67.43 %

5. CONCLUSION

On the analysis of the experimental evaluation it is found that the proposed evaluation framework offers the following advantages:

- In the proposed evaluation framework complexity is linearly proportional to the quality attributes considered and the number of participating candidate architectures. There complexity is reduced to the order of m*n when compared with the existing quantitative techniques [10,14].
- Architecture selection is based on its conformance to quality requirements, stakeholders' quality preferences and its capability to adapt to structural changes in the architecture.
- In the proposed evaluation framework, variation in preference for changes in measured value of quality characteristics is modeled using two dimensional multivariate distributions. This enables the evaluators to quickly verify the small changes for conformance with stakeholders' requirements, thereby avoiding the need for repeating the entire evaluation process.
- The proposed evaluation framework provides statistical evidences for the choices being made and the results are quantitatively expressed.

The developed evaluation framework uses two dimensional multivariate distributions in which all the stakeholders were assigned equal priorities and the quality attributes are independent. However, in reality all stakeholders are not treated equal and the quality attributes have interdependencies. A possible improvement in the developed evaluation framework is to use n-dimensional multivariate distribution by which the architecture can be analyzed in multiple dimensions.

6. REFERENCES

[1] G. Zayaraz and P. Thambidurai, "Quantitative Model for the Evaluation of Software Architectures," *Journal of Software Quality Professional*, *American Society for Quality,* vol. 9, no. 3, pp. 28-40, June 2007

[2] S. Richard and R. Ronald, "Interconnectivity Analysis for Large Software Systems," *Proceedings of the California Software Symposium*, pp. 3-17, Mar.1995.

[3] G. Zayaraz, P. D. Thambidurai, M. Srinivasan and P. D. Rodrigues, "Software quality assurance through COSMIC FFP," *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 5, pp. 1-5, Sep. 2005.

[4] T. L. Saaty, *The Analytic Hierarchy Process*, McGraw Hill, Inc., 1980.

[5] T. K. See and K. Lewis, "Multi attribute Decision Making Using Hypothetical Equivalents," *Proceedings of the ASME Design Technical Conference,*pp. 176-182, Oct. 2002.

[6] C. Chatfield and A. J. Collins, "Introduction to Multivariate Analysis," *Journal of the American Statistical Association*, vol. 78, no. 381, pp. 212-213, Mar. 1983.

[7] M. Svahnberg, C. Wohlin, L. Lundberg and M. Mattsson, "A Method for Understanding Quality Attributes in Software Architecture Structures," *Proceedings of the 14th International Conference on Software*

Engineering and Knowledge Engineering, pp. 819-826, July 2002.

[8] G. Zayaraz and P. Thambidurai, "Software Architecture Selection Framework Based On Quality Attributes," *Proceedings of the IEEE Conference INDICON*, pp. 167-170, Dec. 2005.

[9] S Vijayalakshmi, G Zayaraz and V Vijayalakshmi, "Multicriteria Decision Analysis Method for Evaluation of Software Architectures," International *Journal of Computer Applications*, PP. 22–27, February, 2010

[10] F. Losavio, L. Chirinos, N. Levy and A. Ramdane, "Quality Characteristics of Software Architecture," *Journal of object Technology*, vol. 2, no. 2, pp. 133-150, Mar. 2003.

[11] A. Volker and J. Giesen, "Requirements Interdependencies and Stakeholders Preferences," *Proceedings of the 10th Anniversary IEEE Joint international Conference on Requirements Engineering*, pp. 206-212, Sep. 2002.

[12] B. Boehm and A. Egyed, "WinWin Requirements Negotiation Process: A Multi Project Analysis," *Proceedings of the 5th International Conference on Software Process,* pp. 125-136, Aug. 1998.

[13] A. V. Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," *Proceedings of the 5th IEEE International Symposium on Requirements Engineering,* pp. 249-263, Aug. 2001.

[14] Dr. G. Zayaraz, Dr. V. Vijayalakshmi, M. Venkatesan and S. Mathivanan, "Quantitative Analysis of Service Oriented Architectures," *CIIT International Journal of Engineering and Technology,* pp. 1-5, May 2011.