Selection of Component in Component Based Software Development using Analytical Network Process

Mahima Gupta
Research Scholar
Department of Software Engineering
Poornima College of Engineering
Jaipur, India

Mr. D.K. Somwanshi
Associate Professor
Department of Electrical Engineering
Poornima College of Engineering
Jaipur, India

Abstract: **Software companies are developing interest in making product that is reliable, need short development time and cost effective which moves them toward Component Based Development. In Component Based Development the selection of component is the most crucial part due to its 'Black-Box' nature. Many researches have been done in this field but they have their shortcomings like inadequate address of Non-functional requirements, use of Hierarchical Process, no learning from previous selection, no practical implementation. This paper presents the Component selection method using Analytical Network Process and its implementation for the selection of a component namely Payment Gateway for a commercial website. It not only makes the process automatic but also allows us to give preference to particular specification depending upon our situation and make the decision accordingly.**

Keywords: **Component Based Software Development (CBSD), Component Based Software Engineering (CBSE), Selection of Component, Analytical Network Process**

I. INTRODUCTION

Software Industries are witnessing expansion in their business in everyday life. Software is becoming the key factor for success even in traditionally non-software areas. The software companies are moving towards Component Based Software Development (CBSD)[12]. CBSD not only promises efficient and effective reuse of software component but also increases reliability and quality because the component is already tested by the vendor [12].It reduces the challenges like cost, unsatisfying requirements, and delayed schedules which occur due to the increasing complexity of software and adaptability to change [7].

Component based software development approach is based on the idea to develop software systems by selecting appropriate off-the-shelf components and then to assemble them with well-defined software architecture [15].These components are commonly named as COTS (Commercial-Off-the-shelf) Software Component. COTS Software Component are generally licenced or sold which does not include source code [10]. COTS Software Component can be connected with main software with the help of 'Glue-Code'. The Glue-Code can be API (Application Programming Interface), DLL (Dynamic Link Libraries) or any executable file [9]. Companies are going with COTS Software Components rather than developing the complete software on their own [1].

It looks very promising to use COTS but it also introduces new problems and risk. The major risk is selection of COTS upon which the entire software depends [9].The wrong selection may cause increased time to implement and additional costs and can also lead to failure of main software. For example, the COTS Software Component is made for different organisations and we have limited access to the internal design, it is possible that higher version has certain issues. The evaluation and selection of COTS software Component is still performed using ad-hoc manners in most organisations, such as depending on the experiences of developer team or their intuition, or depending on the relationship with particular vendor[7]. Therefore, lack in systematic, repeatable, and well-defined process for evaluating and selecting COTS software in the industry keeps the organizations under the pressure. Furthermore, the development team has lack of experiences to plan for the selection process in detail. Even though many methods have been proposed to evaluate and select COTS Software Component but still there are certain problems and issues that are not being considered.

II LITERATURE SURVEY

For the selection of COTS Software Components there is no commonly accepted or standard method [5]. The COTS selection is the complex decision making problem and has become a challenge for software development industry due to the following reason:

1) Non Functional Requirements

   While selecting the COTS Software Component, experts analyse the functionality of the Component but Non-functional requirements have always been neglected [11]. It is important to identify the quality of Component which includes reliability, flexibility, usability etc.

2) The lack of learning from previous methods[2]

   The previous information that can be used:

   i. Previous Software Components Chosen

ii. Successful criteria and techniques

iii. Information about Vendor

3) Iterative Learning

Iteration means the act of repeating process in order to reach the desired goal. The Selection of Component method should be iterative one[13].

4) Hierarchical Decision Making

The hierarchical decision making decompose the problems into different hierarchy and the bottom hierarchy depends upon the top one. The top elements cannot be affected by the bottom so there is a need of network between different criteria and elements.

5) Problems in previous methods.

TABLE I. SHORTCOMING OF PREVIOUS COTS SELECTION METHOD

Method Name	Year	Shortcoming [2,3,5,23]
OTSO [16]	1995	Assumes requirements, Does not support Multiple Selection, and does not handle mismatches. Use of AHP.
IusWare [17]	1997	Same CF value for all vendors. Assumes requirements, Does not support Multiple Selection, and does not handle mismatches. Use AHP.
PRISM [5]	1997	Generic Architecture, Assumes requirements, Does not support Multiple Selection, does not handle mismatches.
CISD [20]	1997	Assumes requirements, does not handle mismatches.
PORE [13]	1998	Does not support Multiple Selection, Use of AHP, laborious, when to stop requirement acquisition
CEP[18]	1999	Same CF value, Assumes requirements, Does not support Multiple Selection, does not handle mismatches. Use AHP.
STACE [19]	1999	Does not support Multiple Selection, does not handle mismatches. Use AHP, require more efforts
CRE [10]	1999	Does not support Multiple Selection, does not handle mismatches. Use AHP, NFR add extra effort
CAP[19]	2000	Assumes requirements, Does not support Multiple Selection, do not handle mismatches. Use AHP.
CARE [5]	2001	Not clear how to define requirements, No process that identify the influence the mismatch and handle mismatch
PECA [5]	2002	No clear guideline on how to tailor, Does not support Multiple Selection, does not handle mismatches.
BAREMO [21]	2002	Assumes requirements, Does not support Multiple Selection, and does not handle mismatches. Use AHP.
Combined Selection Approach [5]	2002	Does not handle mismatches, NFR not addressed, no formal evaluation process
WinWin Approach [22]	2003	No formal process to give multiple COTS selection, handle risk of mismatches but does not tells how to handle it.
DesCOTS [8]	2005	Does not handle mismatches, does not handle multiple COTS selection
MiHOS [6]	2007	does not handle multiple COTS selection, does not address quality requirement
Repository [2]	2006	Maintaining and updating the repository, manual maintenance
GAP Analysis [3]	2008	No formal process to give multiple COTS selection
The state of Art [2]	2011	Basically gives framework.

III PROPOSED METHODOLOGY

The COTS evaluation and selection is a crucial part of CBSE. Here we present a solution using Analytical Network Process (ANP). It consists of 5 steps:

i) Searching of Components

ii) Filter Components

iii) Evaluation Criteria

a. Creating Baseline and High Preference

b. Functional Requirement

c. Non-functional Requirement

iv) Getting Weights and Apply ANP

v) Analyse and Final Decision using ANP

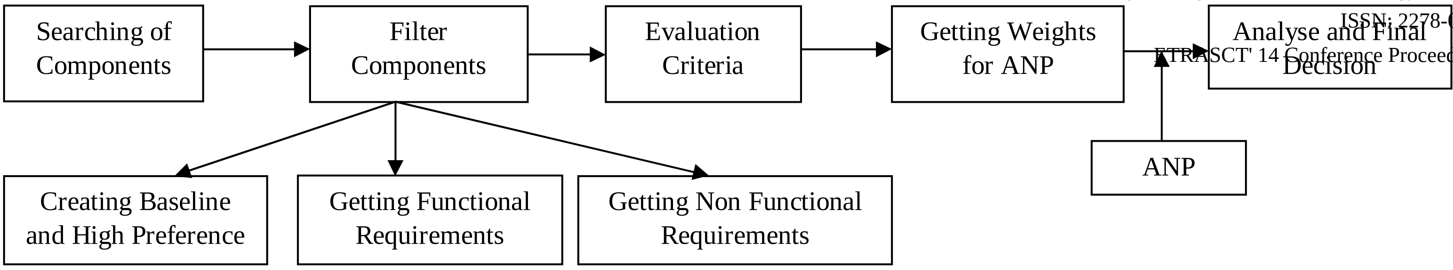

Fig. 1 Flow Diagram of the propose Model

IV PROCEDURE FOR SELECTION OF COMPONENT

Step 1: Searching of Component

This process attempts to identify and find all the potential candidates [16]. The search process is driven by the evaluation criteria defined. The alternatives should have the information related to name, source (name and address of company or website) and main characteristics and features. One can get the alternatives from In-house Libraries, Internet, Advertisements in Magazines, Trade Shows, Popular Vendors or others.

Step 2: Filter Component

This step is to filter the components according to the requirements. The result of this step should shortlist the most promising COTS candidates which are to be evaluated in detail. This can be done by using Survey, Interview or Questionnaire. Here the components are shortlisted on the basis of customer's reviews that are people who have used it.

Step 3: Evaluation Criteria

a. Creating Baseline and High Preference
A baseline is defined as set of characteristics that each alternative must meet or exceed. It can be considered as minimum needed functionality. On the other hand high preference is the maximum functionality needed. If the component's characteristics exceed the high preference it should not create any problem [16].

b. Functional Requirement
Functional requirement is what system is supposed to accomplish. We have left with 4 payment gateways. Respondents were asked to select the best among the functional criteria.

c. Non-functional Requirement
A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. They are often called Quality of the system. Respondents were asked to select the best among the non-functional criteria.

Step 4: Getting Weights and Applying ANP

The Analytical Network Process (ANP) is a decision making process [14]. It is generalization of Analytical Hierarchy Process (AHP). It can model complex decision making problem and allows loops and feedback connections. In Hierarchy process pair wise comparison of Criteria and Sub criteria is done. It is top down structure from the overall objective to criteria, criteria to sub criteria down to alternatives [14].

In ANP criteria, sub criteria and alternatives are treated equally as node and each node can be compared to any other node. So here, the ranking of alternative not only depends upon the weighing of criteria but alternatives can also influence the ranking of criteria [14].

Steps of ANP:

a) Model Construction and Problem Formation
Initially we get Goal of our problem that is what we want to achieve in the end, the criteria and/or sub-criteria which are to be taken care in achieving Goal and Alternatives that is the possible selections we have. The Criteria and Alternatives are represented as Elements or Nodes in ANP. Here we get the control hierarchy which consists of network relationship between goal, criteria and sub-criteria. Next we get the network hierarchy that is the relationship between elements and clusters which is the interdependence (both inner and outer dependence) and feedback among clusters and elements.

b) We then divide our problem into clusters. Clusters are the group of criteria/sub-criteria with common characteristics.

c) Getting Influence Matrix
We get influence matrix which consist of all the nodes or elements horizontally and vertically. The element in the matrix is 1 if column header node influence row header node else it is 0.

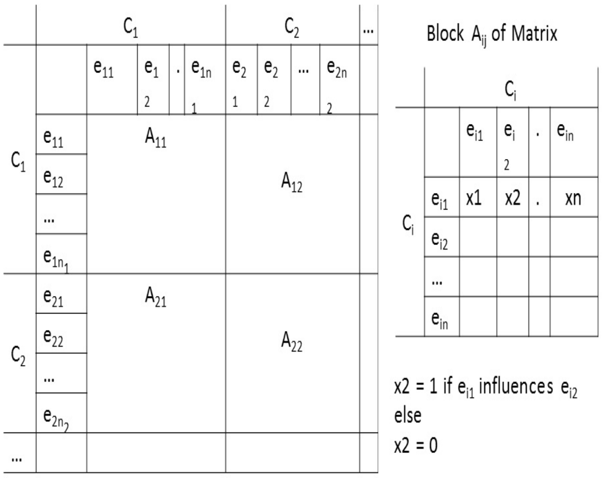

Fig. 2 Influence Matrix

d) Pairwise Comparison Matrix

Perform paired comparisons on the clusters as they influence each cluster and on those that it influences, with respect to that criterion. We get pairwise comparison matrix that is the 1 in the influence matrix is replaced by weights according to Saaty Scale [13] that is how much the first element influence second. Zero is assigned where there is no influence. Pairwise comparison is again performed on the elements within the clusters themselves according to their influence on each element in another cluster they are connected to (or elements in their own cluster). The weights can also be assigned with the help of Questionnaire, Surveys or Expert Opinion. 'A' is the pairwise Comparison Matrix.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{mn} \end{pmatrix}$$

n is the order of matrix

e) Calculate principal Eigen Vectors and Eigen Values

$$W = \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{pmatrix} \quad \text{Where } w_i = \frac{1}{n} \sum_{j=1}^{n} \frac{a_{ij}}{\sum_{i=1}^{n} a_{ij}}$$

Where w_i = Eigen Value (Weights)

n = No. of Criteria

Here W is Eigen Vector

f) Consistency Check

Initially we built a normalised Pairwise Comparison Matrix.

$$A_1 = \begin{pmatrix} a_{11}' & a_{12}' & \dots & a_{1n}' \\ a_{21}' & a_{22}' & \dots & a_{2n}' \\ \dots & \dots & \dots & \dots \\ a_{n1}' & a_{n2}' & \dots & a_{nn}' \end{pmatrix}$$

$$\text{Where } a_{ij}' = \frac{a_{ij}}{\sum_{i=1}^{n} a_{ij}} \qquad \text{For } i,j = 1,2..n$$

Here A_1 is Normalised Pairwise Comparison Matrix

Calculate W' = AW where W is the Eigen Vector

$$\lambda_{max} = \frac{1}{n}\left(\frac{w_1'}{w_1} + \frac{w_2'}{w_2} + \dots\dots + \frac{w_n'}{w_n}\right)$$

λ_{max} is the largest Eigen Value of the Pairwise Comparison Matrix

Calculate Consistency Ratio (CR) and Consistency Index (CI).

$$CR = \frac{CI}{\text{Random Index (RI)}} \qquad CI = \frac{\lambda_{max} - n}{n-1}$$

Random Index (RI) is given by Satty [13]. If CR <= 0.1 then evaluation process satisfies the consistency else we will be having conflicting judgement.

g) Formation of Super matrix

Super Matrix is formed which consist of all nodes horizontally and vertically. The super matrix lists down all the sub-matrixes consisting of all the clusters and necessary elements in order on the left and upper sides of the matrix, where each matrix segment represents a relationship between two nodes (components or clusters) in a system. The element in the matrix represents weights from node of column header to node of row header.

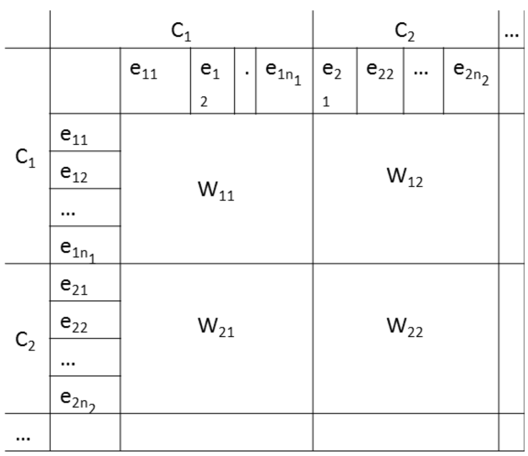

Fig. 3 Super Matrix

ISSN: 2278-0181
ETRASCT' 14 Conference Proceedings

Make the aggregate of column vectors of Super Matrix to 1 and get weighted Super Matrix. The weighted Super Matrix is taken to the power of k+1, where k is an arbitrary number , to get Limit Matrix.

h) The control hierarchy may or may be of more than one layer. Generally the two layers BOCR model is made that is Benefit, Opportunities, Cost and Risk. The evaluation Formula can be:

$$(B*O)/(C*R) \text{ or } \quad (1)$$

$$(B+O)-(C+R) \quad (2)$$

Step 5: Analyse and Final Decision

The Limit Super Matrix itself gives us the result regarding which component is best suitable for us. We also get the result according to the different criteria. The final decision is taken by the decision making authority.

CONCLUSION

This paper illustrated the selection process of component in Component Based Software Development. This research presents the model based on Analytical Network Process for decision making. It eliminates the shortcomings of previous methods like use of hierarchical process, ignorance of non-functional requirements, mismatch handling etc. Unlike other approaches it emphasis on functional requirements, non-functional requirements, Cost and risk involved. The model is based on to select the component which fulfils all the functional and non-functional requirements, reduces risk of using and has minimum cost. It is worth noting that this process is cost effective, automatic and also consumes less time. It gives multiple component selection option and sensitivity and preference (priority) can be given to any criteria.

However as the limitation of this process the decision of large COTS using ANP becomes a complex task and filter process is necessary to use. In terms of future work it is necessary to automatize the process completely and it should support large COTS.

REFERENCES

[1] Feras Tarawneh, Fauziah Baharom, Jamaiah Hj. Yahaya and Azida Zainol “COTS Software Evaluation and Selection: a pilot Study Based in Jordan Firms”, IEEE International Conference on Electrical Engineering and Informatics Bandung, Indonesia, pp. 1-5, July 2011.

[2]Feras Tarawneh, Fauziah Baharom, Jamaiah Hj. Yahaya, Faudziah Ahmad “Evaluation and Selection COTS Software Process: The State of the Art”, International Journal on New Computer Architecture and their Application, ISSN No: 2220-9085, Vol-2, No-1, 2011.

[3]Jinfang Sheng, Bin Wang “Evaluating COTS Components Using Gap Analysis”, The 9th IEEE International Conference for Young Computer Scientists Bandung, Indonesia, pp-1248-1253, November 2008.

[4] Tom Wanyama, Behrouz H. Far “An Empirical Study to Compare Three Methods For Selecting Cots Software Components”, International Journal of Computing and ICT Research, Vol. 2, No. 1, pp. 34 – 46, June 2008.

[5] Abdallah Mohamed, Guenther Ruhe, Armin Eberlein “COTS Selection: Past, Present, and Future” 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, pp-103-114, March 2007.

[6] Abdallah Mohamed, Guenther Ruhe, Armin Eberlein “MiHOS: an approach to support handling the mismatches between system requirements and COTS products”, Springer Volume-12, No-3, pp: 127-143, July 2007.

[7]Tom Wanyama, Agnes F. N. Lumala “Decision Support for the selection of COTS”, IEEE Conference on Electrical and Computer Engineering, Canada, pp: 908-911, May 2005.

[8] Carme Quer, Xavier Franch, Xavier Lopez-Pelegrin “DesCOTS-EV: A Tool for the Evaluation of COTS Components” 13th IEEE International Conference on Requirements Engineering, pp: 457-458, Aug-Sept 2005.

[9] Ivica Crnkovic “Component-based Software Engineering – New Challenges in Software Development”, IEEE 25th International Conference on Information Technology Interfaces, pp: 9-18, June-2003.

[10]C. Alves and J. Castro, "CRE: A Systematic Method for COTS Components Selection", SBES'01Rio de Janeiro, Brazil, 2001.

[11] Dukic L. “Non-Functional Requirements for COTS Software Components”, International Conference of Software Engineering 2000 Workshop on Continuing Collaborations for Successful COTS Development, Linerick, Ireland, June 2000.

[12]Baoming Song “A General Model for Component-based Software”, A Thesis submitted to the Faculty of Computer Science In Partial Fulfillment of the Requirements for the Degree of Master of Computer Science in Dalhousie University – DalTech 2000.

[13] C. Ncube and N. A. Maiden, "PORE: Procurement-Oriented Requirements Engineering Method for the Component Based Systems Engineering Development Paradigm", International Workshop on Component Based Software Eng., Los Angeles, CA, 1999

[14] Thomas L. Saaty “Fundamentals of the Analytic Network Process” International Symposium on Analytical Hierarchy Process 1999, Kobe, Japan, August 1999.

[15] Dr. Gilda Pour “Moving toward Component-Based Software Development Approach” IEEE Proceedings of Technology of Object Oriented Languages, Beijing, and pp: 296-300, September 1998.

[16] Jyrki Kontio “OTSO: A Systematic Process for Reusable Software Component Selection”, ACM, University of Maryland, MD, USA, December 1995.

[17] M. Morisio and A. Tsoukis, "IusWare: a methodology for the evaluation and selection of software products," IEE Software Engineering vol. 144 (3), June 1997.

[18] B. C. Phillips and S. M. Polen, "Add Decision Analysis to Your COTS Selection Process," The Journal of Defense Software Engineering, April 2002.

[19] M.Ochs, D.Pfahl, G.Chrobok-Diening, and B. Nothhelfer-Kolb, "A COTS Acquisition Process: Definition and Application Experience," *ESCOM-COPE'00*, Munich, Germany, 2000.

[20] V.Tran & D.Liu,"A Procurement-centric Model for Engineering Component-based Software Systems," SAST '97, 1997, pp.7079.

[21] A. Lozano-Tello and A. Gomez-Pérez, "BAREMO: how to choose the appropriate software component using the analytic hierarchy process," in SEKE'02, Italy, 2002, pp. 781-788.

[22] B.Boehm, D.Port, and Y.Yang,"WinWin Spiral Approach to Developing COTS-Based Applications,"*EDSER-5*, Oregon, 2003.

[23] Tom Wanyama and Behrouz Homayoun Far “Repositories for Cots Selection” IEEE Canadian Conference on Electrical and Computer Engineering Ottawa, pp: 2416-2419, May 2006