Empir Software Eng
DOI 10.1007/s10664-008-9094-4

Software architecture evaluation methods based on cost benefit analysis and quantitative decision making

Jihyun Lee · Sungwon Kang · Chang-Ki Kim

Editor: Kenichi Matsumoto

Abstract Since many parts of the architecture evaluation steps of the Cost Benefit Analysis Method (CBAM) depend on the stakeholders' empirical knowledge and intuition, it is very important that such an architecture evaluation method be able to faithfully reflect the knowledge of the experts in determining Architectural Strategy (AS). However, because CBAM requires the stakeholders to make a consensus or vote for collecting data for decision making, it is difficult to accurately reflect the stakeholders' knowledge in the process. In order to overcome this limitation of CBAM, we propose the two new CBAM-based methods for software architecture evaluation, which respectively adopt the Analytic Hierarchy Process (AHP) and the Analytic Network Process (ANP). Since AHP and ANP use pair-wise comparison they are suitable for a cost and benefit analysis technique since its purpose is not to calculate correct values of benefit and cost but to decide AS with highest return on investment. For that, we first define a generic process of CBAM and develop variations from the generic process by applying AHP and ANP to obtain what we call the CBAM+AHP and CBAM+ANP methods. These new methods not only reflect the knowledge of experts more accurately but also reduce misjudgments. A case study comparison of CBAM and the two new methods is conducted using an industry software project. Because the cost benefit analysis process that we present is generic, new cost benefit analysis techniques with capabilities and characteristics different from the three methods we examine here can be derived by adopting various different constituent techniques.

Keywords Software architecture evaluation · CBAM · AHP · ANP

J. Lee (✉) · S. Kang
Software Technology Institute, Information and Communications University, Seoul, South Korea
e-mail: puduli@icu.ac.kr

S. Kang
e-mail: kangsw@icu.ac.kr

C.-K. Kim
Electronics Telecommunications Research Institute(ETRI), Gajong-dong, Daejeon, South Korea
e-mail: ckkim1@etri.re.kr

1 Introduction

Software architecture design and evaluation are closely related activities. Software architecture should be designed such that it reflects business goals and Quality Attributes (QAs) such as performance, security, modifiability, usability and etc. of the system for which the architecture is being designed, and architecture evaluation methods enable us to evaluate how well they are reflected in the designed architecture. Two most widely used architecture evaluation methods are the Architectural Tradeoff Analysis Method (ATAM) (Clements et al. 2002) and the Cost Benefit Analysis Method (CBAM) (Kazman et al. 2002). ATAM assists architects in identifying tradeoffs among QAs and identifying risks when we design architecture. CBAM helps architects select Architectural Strategies (ASs) that are optimal for a given system during its evolution by calculating economical tradeoffs based on the benefits, costs, schedule, and inherent risks for each candidate AS.

CBAM quantifies benefits using utility values which represent design preference determined by votes or consensus among stakeholders. The benefit that will be added to the system through the implementation of an AS is referred to as *utility*. Each AS provides a specific level of utility to the stakeholders, and each AS has cost and schedule implications (Kazman et al. 2002). But since the utility value of a design is determined independently, we cannot guarantee that the value is a reasonable one. Besides, since decision making by consensus cannot reflect the knowledge of each stakeholder, much uncertainty gets into the design decision. Moreover, since the stakeholders should consider various kinds of decision information (i.e. scenarios and ASs) simultaneously and independently the possibility of making misjudgment is high.

In order to overcome this limitation of CBAM, we propose in this paper two new CBAM-based methods to software architecture evaluation, which respectively adopt the Analytic Hierarchy Process (AHP) and the Analytic Network Process (ANP), which are decision making methods widely used and validated in industry and government. Since they use pair-wise comparison instead of comparing several decision elements concurrently and also because the result values are relative preference they are suitable for a cost and benefit analysis technique since in such a technique the purpose is not to calculate correct values of benefit and cost of ASs but to determine AS with high expected benefit. For that, in this paper we first define a generic process of CBAM and develop variations from the generic process by applying AHP and ANP to obtain what we call the CBAM+AHP and CBAM+ANP methods, which then can be selected according to the importance of decision making. Through utilizing AHP or ANP, we can not only reflect the knowledge of experts in a more sophisticated way but also can reduce the possibility of subjective misjudgment. A case study comparison of CBAM and the two new methods is conducted using an industry software project. Because the cost benefit analysis process that we presented in this paper is generic, new cost benefit analysis techniques with capabilities and characteristics different from those we developed in this paper can be developed.

This paper is organized as follows: In Section 2, CBAM, AHP and ANP are explained as they are basic building blocks of the proposed methods. In Section 3, a generic process for software architecture evaluation methods based on cost benefit analysis is presented together with the CBAM+AHP and CBAM+ANP methods. Then in Section 4 we conduct a case study comparison of CBAM, CBAM+AHP and CBAM+ANP. In Section 5, we discuss related works. Finally in Section 6, we conclude our paper by mentioning the contributions of the paper and the future research directions.

2 Background

In this section we give an overview of the CBAM, AHP, and ANP approaches that are the building blocks of the methods we propose in the following sections.

2.1 CBAM

CBAM is a method for selecting the best AS by considering the costs accompanied by ASs such as the effort, the schedule, and the risks as well as their benefits. Since many steps of the CBAM process depend on the stakeholders' empirical knowledge and/or intuition, it is very important to make the knowledge of experts accurately reflected in determining ASs.

According to Kazman et al. (Kazman et al. 2002), "software architecture design activity is in maximizing the difference between the benefit derived from the system design and the cost of implementing the design." The Cost Benefit Analysis Method (CBAM) process can be seen as being composed of two main phases, the ASs Development Phase and the Cost benefit Analysis Phase, as shown in Fig. 1 (Kazman et al. 2002). The first phase serves to reducing the size of the decision space through refining and prioritizing the quality requirements scenarios. Also in the first phase, collected scenarios are collated with the scenarios elicited during the ATAM and then the top one-thirds are chosen based on their

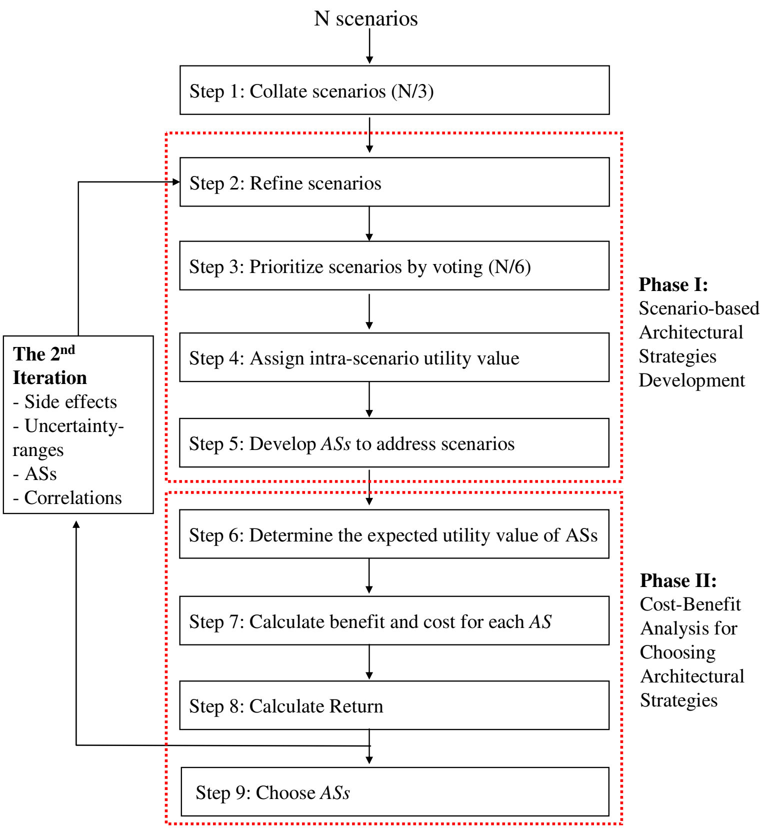


Fig. 1 Overview of the CBAM process

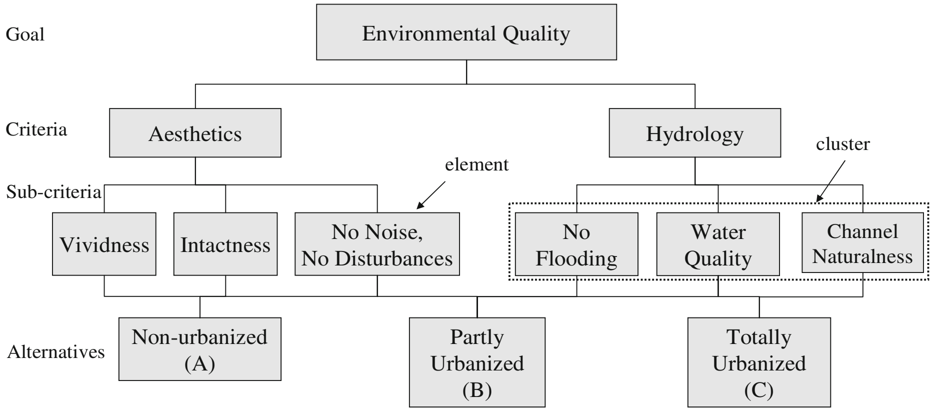


Fig. 2 Hierarchy for Brandywine River Region example from (Saaty 1999)

importance for the business goals of the system. The chosen scenarios are refined by focusing on their stimulus/response measures and their worst, current, desired, and best-case QA response levels are elicited. QA response level provides a tangible representation for reducing room for subjective interpretation and also is used to interpolate utility values of each AS later on. Stakeholders vote for each scenario and choose top 50% of them.[1] After assigning utility values for each scenario, ASs for addressing the scenarios are developed. The purpose of the second phase is to compute benefits and costs for ASs. Information for decision making such as utility values of ASs and benefits are calculated based on this utility values and weights of scenarios obtained at the first phase. Also costs for implementing ASs are collected from stakeholders and an initial ranking of the ASs is established based on the calculated return (benefit/cost).

The two phases can be repeated up to two times. The second iteration can be optionally used for more sophisticated and realistic decision makings by considering information about risk, uncertainty, side effects and etc.

2.2 AHP

Analytic Hierarchy Process (AHP) is a decision making approach designed to select the best alternative from a set of alternatives by evaluating them with respect to a set of given criteria. The following are the main steps of AHP:

1. Structuring the problem into a hierarchy
2. Setting priorities through pair-wise comparisons
3. Synthesizing the judgments and checking logical consistency

We illustrate the AHP method with an example used by T. L. Saaty (Saaty 1999). Figure 2 is the result of applying Step 1 to the Environmental Quality goal.

The highest level of the hierarchy is the overall objective (i.e. Environmental Quality) and the intermediate levels consist of criteria and sub-criteria for evaluating it. The lowest level consists of alternatives that will contribute positively or negatively to the main goal

[1] CBAM does not provide a detailed method for reducing scenarios. Choosing the top one-third or top 50% is not mandatory and in some cases all collected scenarios may be used.

through their impact on the intermediate criteria. Here we call a node in the hierarchy graph *an element* and a homogeneous group of elements in the graph *a cluster* (or *a component*).

In Step 2 of AHP, the decision maker judges the relative importance of all the elements in the same clusters. For example, pair-wise comparisons are conducted to decide relative importance of 'Vividness', 'Intactness', and 'No Noise, No Disturbances' with respect to 'Aesthetics' criteria. In this way, the decision maker carries out simple pair-wise comparisons, which are then used to determine the overall priorities of the alternatives.

Figure 3 is a pair-wise comparison matrix. This matrix is generated for each level, i.e. relative importance of the criteria based on the goal. For example, we can get a pair-wise comparison matrix for relative importance of 'Aesthetics' and 'Hydrology' with respect to 'Environmental Quality', a matrix for relative importance of 'Vividness', 'Intactness', and 'No Noise, No Disturbances' with respect to 'Aesthetics' criteria, and so on (from the hierarchy of Fig. 2). a_{ij} represents the importance of alternative i over alternative j. The eigenvector consolidates the relative importance ratios of the matrix into the measures of intensity where intensity indicates the weight or degree of preference. Step 3 of AHP synthesizes all of them for calculating the total priorities. For example, in Fig. 2, if 'Vividness' is three times as important as 'Intactness' we enter 3 in the a_{12} position and enter 1/3 in the a_{21} position in Fig. 3.

Also AHP allows for lack of consistency in judgments and provides a means to improve consistency (Saaty 1980; Saaty and Vargas 1982, 1994). When evaluating the architectural alternatives for real world systems, there may be dependencies from lower level elements to higher level elements besides the dependencies from higher level elements to lower level elements. That is, even though an element, whose weight is very small within a higher level, can become important in the process of considering loops and feedbacks if its weight is high when considered from lower level to higher level Also there may be dependencies among the elements within the same level. But since AHP assumes that criteria and alternatives are orthogonal and have no feedback between them, we cannot guarantee that the result is an optimal solution in case there exists interdependency among elements.

2.3 ANP

Analytic Network Process (ANP) is a general framework for dealing with decisions. It does not assume that higher-level elements are independent from lower level elements or that the elements within the same level of the hierarchy are independent from each other (Saaty 2005). In AHP, *dominance*, or the relative importance of influence, was the central concept but now ANP uses a network where a network is a structure with no associated levels. Figure 4 illustrates the difference between a hierarchy and a network. A hierarchy has a goal, criteria, and alternatives that are organized as a top down linear structure (illustrated in Fig. 4a) while the network spreads out in all directions and its clusters are not arranged in a particular order (illustrated in Fig. 4b). But it is possible that a network has a hierarchy as

Fig. 3 Pair-wise comparison matrix

$$W_{ij} = \begin{bmatrix} w_1/w_1(a_{11}) & w_1/w_2(a_{12}) & \dots & w_1/w_n(a_{1n}) \\ w_2/w_1(a_{21}) & w_2/w_2(a_{22}) & \dots & w_2/w_n(a_{2n}) \\ \vdots & \vdots & & \\ w_n/w_1(a_{n1}) & w_n/w_2(a_{n2}) & \dots & w_n/w_n(a_{nn}) \end{bmatrix}$$

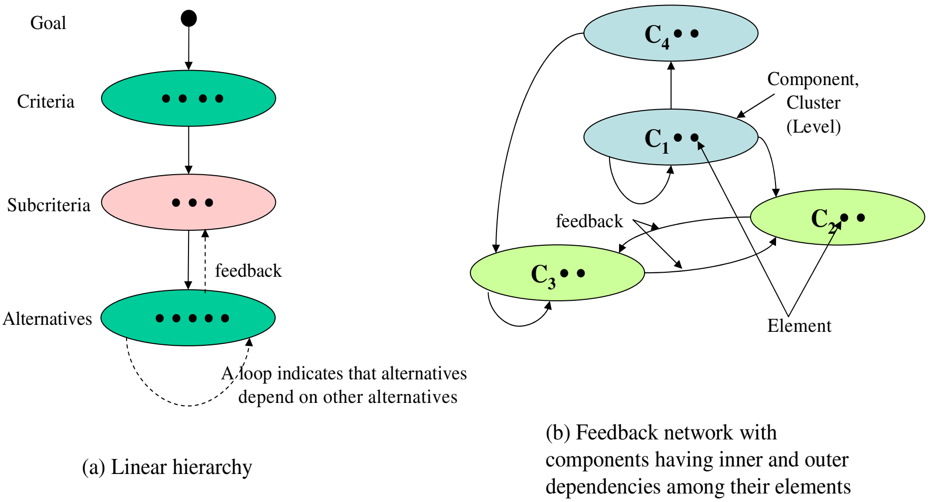
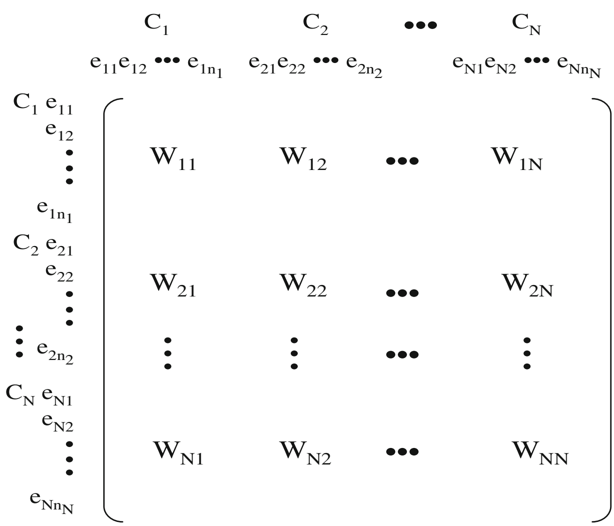


Fig. 4 Difference between a hierarchy and a network structure (Saaty 2005)

its main structure but has additional dependencies (illustrated with the upward arrow or loop like dotted line in Fig. 4a).

Assume that we have a system of N clusters in which the elements in a cluster interact with or have an impact on or are themselves influenced by other elements of the same component or other components. And assume that there is a property governing the interactions of the entire system. Cluster h, denoted by C_h, $h=1, \ldots, N$, has n_h elements, that we denote by $e_{h1}, e_{h2}, \ldots, e_{hnk}$ (Saaty 2005). The priority vector Wij, which is the same as in Fig. 3 for AHP, is derived by pair-wise comparisons. The values of the uninfluenced elements are set to zero. The following supermatrix synthesizes relative intensities of clusters (Fig. 5).

Fig. 5 The Supermatrix of the ANP model (Saaty 2005)



3 The CBAM-Based Software Architecture Evaluation Methods

In this section, we first define a generic process for cost benefit analysis for ASs selection that generalizes CBAM. The generic process shows what are the essential steps and activities that are involved in cost benefit analysis for ASs selection and shows what parts of the process allow alternative methods, thereby opening the possibilities of improving the existing methods. Then based on the generic process, we develop in Sections 3.1 and 3.2, respectively, two CBAM-based methods.

The generic process consists of two phases and is described in Fig. 6. The first phase of the generic process is the application of the Scenario-based ASs Development Technique. In the first phase, the scenarios are determined and AS is developed for each scenario. The second phase of the generic process is the Cost Benefit Analysis Phase, in which the applied technique produces certain benefit and cost scores of the ASs.

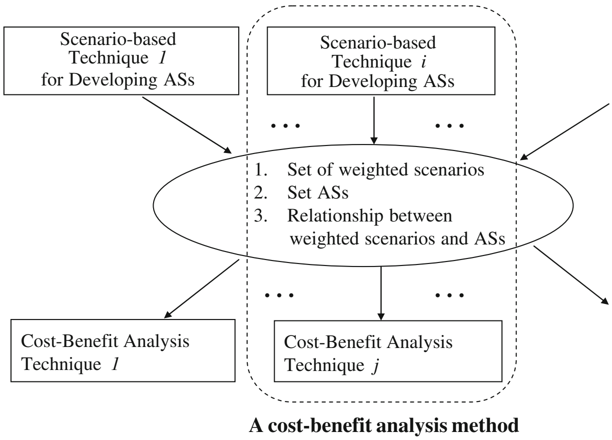
Including CBAM, any specific evaluation method based on the generic process should have the outputs 1, 2 and 3 in Fig. 6, which connect the first phase and the second phase regardless of what specific techniques are used for them.

The details of the first and second phases of the generic process are as follows:

Phase I Scenario-based ASs Development

- Step 1) Choose and refine scenarios: In this step we can collate the scenarios elicited during the ATAM with the new ones proposed by stakeholders for removing duplication. After prioritizing these scenarios based on the satisfaction of system's business goals we choose the top one-thirds for further progress. But in case that the number of scenarios is small or new scenarios are not proposed we do not need to prioritize or collate them. Also in this step, the chosen scenarios are quantified and made verifiable, if necessary.
- Step 2) Prioritize scenarios: Weight of each scenario is determined by using the method specific technique.
- Step 3) Develop ASs to address scenarios: One or more ASs are developed for each scenario.

Fig. 6 Generic process of cost benefit analysis of ASs



Phase II Cost benefit Analysis

- Step 4) Calculate benefit for each AS: Benefit for each AS is calculated by using the method specific technique.
- Step 5) Calculate cost of each AS: Cost for each AS is calculated by using the method specific technique.
- Step 6) Evaluate reliability of decisions: Reliability of decisions is evaluated by using the method specific evaluation technique.
- Step 7) Calculate benefit/cost and prioritize ASs.

Notice that the first and second phases of the generic process correspond to the first and second phases of the CBAM process in Fig. 1. The artifacts produced from the first phase of the generic process are the same artifacts produced in the CBAM process in Fig. 1. All steps of the genetic process are mandatory in CBAM. Most techniques used in CBAM are consensus and voting. So QA response levels used to calculating benefits, determining utility response curves, and dealing with uncertainty are important for correct decision. Benefits in CBAM are presented as a form of utility values and these are interpolated using utility response curve while technique for determining costs is not specified. But because the generic process is defined for classifying common and variable aspects it is different from original CBAM process.

In the following two subsections, we develop two specific methods for architecture evaluation based on the generic process by adopting AHP and ANP. Adopting AHP and ANP mostly affects Step 2 of Phase I and Steps 4 and 5 of Phase II. Steps 4 and 5 of Phase II are the steps for calculating benefits and costs of ASs and Step 2 determines relative weights of scenarios that are to be used in Steps 4 and 5. But appropriately choosing the particular techniques for these steps in the generic process, we can come up with a method equivalent to CBAM or other methods that can evaluate economical aspects of ASs better than CBAM. For example, as we can see in detail later in Table 7 of Section 4, we can choose the best AS through structuring stakeholders' experimental knowledge and intuition better with CBAM+AHP and CBAM+ANP than with CBAM. We can not only reflect the knowledge of experts more accurately but also reduce misjudgments. The effectiveness of AHP and ANP comes from that by concentrating on comparison of just two things, which is the simplest case one can make judgment about, each time we can make as sure judgment as one can make and the final outcome is determined by combining surest judgments thus obtained.

3.1 CBAM+AHP

CBAM+AHP determines weights of scenarios and their relative relations with ASs by conducting pair-wise comparisons rather than, as with CBAM, simply collecting decision information through consensus or vote and the stakeholders' experiences.

AHP constructs a hierarchy by clustering the elements according to their interrelation among them. In the paper (Lee et al. 2006), we constructed an AHP model for determining benefits of ASs. It is reproduced in Fig. 7. At the first level is Benefit of AS, which is the overall goal. At the second level are the criteria that contribute to the goal, which are scenarios in CBAM+AHP. At the third level are the alternatives that will be evaluated in terms of the criteria at the second level, which are ASs in CBAM+AHP.

The hierarchy for determining cost of implementing AS is shown in Fig. 8 (Lee et al. 2006). The alternatives for determining cost and benefit are the same. Also it is possible to construct the hierarchies for risk or uncertainty, if necessary.

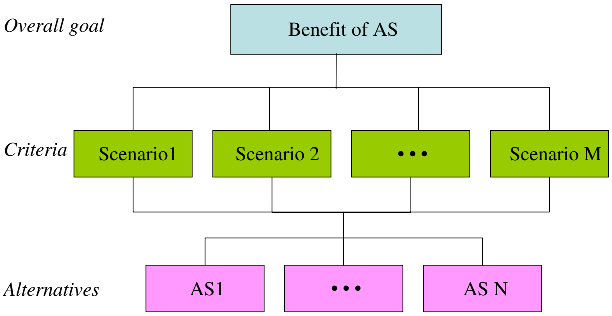


Fig. 7 AHP benefit model for an AS

We determine the ranks of ASs according to the following procedure based on the models of Figs. 7 and 8.

Phase I Scenario-based Architectural Strategies Development

Step 1) Choose and refine scenarios.
Step 2) Prioritize scenarios—Conducting pair-wise comparisons for making judgments about the relative importance of scenarios with respect to the overall goal, i.e. Benefit of AS. The calculated results are used as weights of scenarios.
Step 3) Develop ASs to address scenarios.

Phase II Cost benefit Analysis

Step 4) Calculate benefit for each AS by

a) Conducting pair-wise comparisons of ASs at the alternatives level, comparing them pair-wise with respect to how much one is better than the other in satisfying each scenario at the criteria level.
b) Synthesizing the results of Steps 2 and 4a and calculating benefits of ASs.

Step 5) Calculate cost of each AS by

a) Conducting pair-wise comparisons for making judgments about the relative importance of ASs on the alternatives level with respect to the overall goal, Cost of Implementing AS.
b) Synthesizing the result of Step 5a and calculating costs of ASs.

Step 6) Evaluate reliability of decisions

If the result of consistency checking is not within the range limit, go to Step 2.

Step 7) Calculate benefit/cost and prioritize ASs.

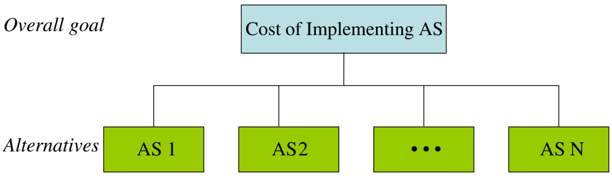


Fig. 8 AHP cost model for an AS

3.2 CBAM+ANP

Benefit and cost affect decision for AS and it is necessary to review which one affects decision more. Also, because an AS can address several scenarios we have to consider relative influence on the criterion level with regard to the alternatives level. Figure 9 is the decision model for ANP. In ANP decision model, benefit and cost are treated within the same network while they are dealt independently in the AHP decision model. In Fig. 9, there are dependencies between goals and scenarios clusters; between scenarios and alternative clusters; between goals and alternative clusters. Also, there are feedbacks from the alternative cluster to the criteria cluster and loop in the goal cluster.

The process of CBAM+ANP is as follows:

Phase I Scenario-based Architectural Strategies Development

Step 1) ~ Step 3) same as CBAM+AHP

Phase II Cost benefit Analysis

Step 4) Calculate benefit for each AS by

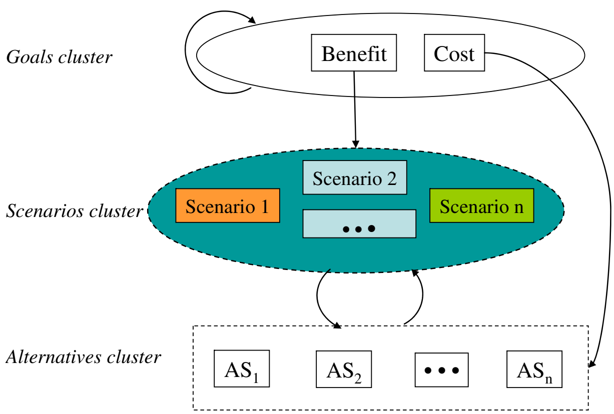
a) Conducting pair-wise comparisons of ASs on the alternatives level, comparing them pair-wise with respect to how much one is better than the other in satisfying each scenario on the criteria level.
b) Conducting pair-wise comparisons of scenarios with respect to how much one is better than the other in satisfying each AS on the alternatives level.
c) Conducting pair-wise comparisons of scenarios on the criteria level with respect to the relative influences on each ASs on the alternatives level.
d) Synthesizing the matrix (using SuperDecisions software (SuperDecision 2007)), supermatrix, two-dimensional matrix of elements by elements.
e) Making limiting forms of supermatrix
f) Synthesizing the results of Steps 2 and 3 and calculating benefits of ASs.

Step 5) Calculate cost of each AS:

This step is the same as that of CBAM+AHP except that the cost is modeled as a goal as with benefit as shown in Fig. 9.

Step 6) ~ Step 7) same as CBAM+AHP

Fig. 9 ANP model for AS decision



In CBAM+ANP, risk and uncertainty can also be modeled as a goal cluster and synthesized.

3.3 Comparison of CBAM, CBAM+AHP and CBAM+ANP

In CBAM+AHP and CBAM+ANP it is not necessary to decide QA response levels, reviewing scenarios addressed by an AS, and cost estimation by using COCOMO. In CBAM stakeholders should decide worst, current, desired, and best QA response goal achieved by a given scenario and AS for deciding utility value. Decision maker draws a utility curve based on these four utility values and interpolate utility values by using it. However, stakeholders do not need to decide the four cases of QA response goals in CBAM+AHP and CBAM+ANP. Because stakeholders decide relative importance between two scenarios it is enough to refine scenarios in a quantitative level for more correct evaluation.

Also stakeholders decide only relative required cost between two ASs while they should estimate cost using COCOMO or other methods in CBAM. The relative cost and benefit obtained from pair-wise comparisons is enough to decide which AS should be implemented with priority. The comparison results of the three methods are summarized in Table 1.

In Step 4 of Table 1, CBAM method determines scenarios affected by each AS, but CBAM+AHP method determines ASs related with each scenario to calculate benefit for each AS. On the other hand, CBAM+ANP method considers both. Therefore CBAM+ANP method is similar to CBAM in that they consider each AS's impact on scenarios but CBAM+AHP method is not.

Table 2 is the result of comparing CBAM, CBAM+AHP and CBAM+ANP from the usability and efficiency point of view.

Table 1 Comparison of CBAM, CBAM+AHP and CBAM+ANP

Generic Process	CBAM	CBAM+AHP	CBAM+ANP
1. Choose and refine scenarios	Consensus		
2. Prioritize scenarios	Vote	Relative weighting (pair-wise comparison)	
3. Develop ASs to address scenarios	Consensus		
4. Calculate benefit for each AS	Assign utility for each scenario Determine scenarios affected by each AS using consensus Interpolate or extrapolating utility of scenarios	Determine relative preferences in unidirectional based on AHP hierarchy model (relative preference of ASs with respect to scenarios)	Determine relative preferences in bidirectional based on ANP network model (relative weights of ASs with respect to scenarios and relative weights of scenarios with respect to ASs)
5. Calculate cost of each AS	COCOMO and etc.	Relative weighting	
6. Calculate benefit/cost and prioritize ASs	Common		
7. Evaluate reliability of decision making	Interpolation using Utility Curve/Uncertainty-range	Consistency rate	

Table 2 Usability and efficiency comparison of CBAM, CBAM+AHP, and CBAM+ANP

	CBAM	CBAM+AHP	CBAM+ANP
Process weight	Heavy	Light	Light
Number of comparisons	N/A	Medium	Large
Calculation complexity	Medium	Medium	High
Each AS's impact on scenarios	Reflected	Not reflected	Reflected
Difficulty of interpolation	Difficult	Easy	
Inconsistency check of decision	Difficult	Easy	Easy
Tool support	N/A	ExpertDecision (ExpertChoice 2007)	SuperDecision (SuperDecision 2007)

In Table 2, process weight indicates the number of steps and the number of participating stakeholders; Number of comparisons means the number of pair-wise comparisons; Calculation complexity means the difficulty of calculating the benefit and cost of ASs; Difficulty of interpolation means how costly it is to deal with inconsistency and uncertainty of decision making; Inconsistency check of decision means the possibility of misjudgment.

3.4 Consistency Checking Decisions

In decision-making problems it is important to have consistent results in the sense that if A is preferred to B and B is preferred to C, then A must be preferred to C. For AHP and ANP, the Consistency Ratio (CR) can be defined as the ratio of Consistency Index (CI) and the Random consistency Index (RI), i.e.

$$\mathrm{CR} = \mathrm{CI}/\mathrm{RI}$$

If n is the number of elements of pair-wise comparison matrix and $\lambda_{\max}$ is eigenvector of pair-wise comparison matrix, CI is calculated as

$$\mathrm{CI} = (\lambda_{\max} - n)/(n - 1) \quad \text{and}$$

and RI means average random CI. In order to get some feel for what the consistency index tells us about a positive n-by-n matrix A, choose the entries of A above the main diagonal at random from the 17 values {1/9, 1/8, ..., 1/2, 1, 2, ..., 8, 9}. Then fill in the entries of matrix A below the diagonal by taking reciprocals and compute the consistency index. Do these 50,000 times and take the average, this is the RI (Saaty 2005). Saaty proposed in his book (Saaty 1990) that if CR is not less than 0.10, then the decision is not consistent, and we must study the given decision making problem and revise the decisions. CBAM does not have any consistency checking method for design decisions but our CBAM-based methods have consistency checking mechanisms. For detailed example of how inconsistency rate is computed, see "Appendix".

4 Case Study Comparison of CBAM, CBAM+AHP and CBAM+ANP

In this section we apply CBAM, CBAM+AHP and CBAM+ANP to an industry system development in order to compare their efficacy. The system for the case study is the Hyundai POI Inspection & Management System (HPIMS) (Kim et al. 2007).

4.1 Overview of the HPIMS System

HPIMS is a tool that provides the following functions:

- An automatic mechanism for inspecting Point Of Interest (POI) raw data that have defects such as logical inconsistency or duplication error.
- A basic analysis mechanism of the POI raw data according to the predefined criteria such as regional distribution, class, and rank of the POI data. It also provides inspection history.
- A convenient graphic user interface for the user of inspection, analysis, and other functionalities.
- Enabling the user to configure the rules that will be applied to inspect and cleanse the target POI raw data.
- A mechanism for cleansing the errors from the POI raw data that are inspected by the inspection function with the predefined cleansing rules. This includes deletion of falsely duplicated records and correction of logical errors.

And HPIMS has to meet the following Quality Attribute Requirements:

- Performance: The execution of inspection with two million POI records should be completed within 10 h.
- Usability: The architecture should enable the user to know how much time tasks have consumed, so that the user can predict the completion time of the tasks. And HPIMS should provide easy ways to cancel the currently running tasks without any corruption of the raw POI data file.
- Integrity: The system state after cancelation should be the same as the state before the execution.
- Modifiability: The architecture can be modified with minimal impact on the other components during the development time. It should also let the users add or change the rules easily after development.

4.2 The Experiment Process

We conducted ATAM before starting CBAM and used the artifacts (i.e. scenarios and ASs) from the ATAM session as inputs. If the customer proposes new quality requirements scenarios, architects develop ASs for analyzing cost and benefit for them. The procedure we used for the experiment is as follows:

(1) Stakeholders (domain expert, analyzer, architect, developer) decide cost and benefit of ASs according to the CBAM process
(2) The same stakeholders of (1) decide cost and benefit of ASs according to the process of CBAM+AHP after being introduced to the method
(3) The same stakeholders of (1) recalculate cost and benefit of ASs considering the additional decision information added as required by CBAM+ANP.
(4) The results from (1), (2), and (3) are compared.

The purpose of the experiment is to show that we can get decision information in a more intuitive and simple way when applying CBAM+AHP or CBAM+ANP. The results of the experiment with the three methods might be similar or different. In case the results of applying CBAM+AHP and CBAM+ANP are different from that of applying CBAM, we decide what actions to take after checking the correctness of decision by comparing the consistency check results for them and the uncertainty-range obtained from CBAM.

If the results of applying the three methods are the same in deciding high priority AS, then we can interpret it as meaning that CBAM+AHP and CBAM+ANP can give the same result as CBAM with less effort. If the results of applying the three methods are different, more reliable one should be determined comparing the uncertainty-range of CBAM and inconsistency rate of CBAM+AHP and CBAM+ANP. Since with CBAM the uncertainty-range is calculated by probability value that stakeholders provide, another kind of uncertainty is injected. Therefore we can more easily evaluate reliability of decision making with CBAM+AHP or CBAM+ANP than with CBAM. Because CBAM+AHP and CBAM+ANP check inconsistency through only the existing decision information there are no additional injection of inconsistency.

4.3 Experiment Results

Step 1) While conducting an ATAM session, several outputs are produced, of which the ASs and the quality attribute scenarios become inputs to CBAM. Collated scenarios are as follows:

- Execution Time of Inspection function (ETI): When a user selects the predefined 200 inspection rules and triggers the inspection with around 700 MB of MS access database file, the execution should finish within 10 h.
- Progress Tracking (PT): A user can know how much time remains for executing the function.
- Canceling Operation (CO): A user can cancel inspection, cleansing or analysis operation being executed. If the operation belongs to cleansing function, the state of POI data should be the same as before the operation was executed.
- Configurable Rule (CR): A user can configure and generate a variety of rule sets from the full list of the inspection or cleansing rules.
- Rule Changeability (RC): When the maintenance staff adds or changes rules, it affects code as little as possible.
- Abnormal Case Handling (ACH): Whenever any error occurs during inspection or cleansing operation in the rule executor, the errors should be notified to the coordinator component. The ongoing operation continues after processing the incoming error.

Step 2) CBAM. In CBAM, stakeholders allocate votes to each scenario which is refined with QA response levels of the worst, current, desired, and best. The votes allocated to the entire set of scenarios were 10. Table 3 shows the results.

Table 3 Refined quality requirement scenarios with response levels

Scenario	Votes	Response levels			
		Worst	Current	Desired	Best
ETI	14	12 h	10 h	8 h	4 h
PT	5	4% fail	3% fail	1% fail	0% fail
CO	5	>5% lost	2% lost	0% lost	0% lost
CR	6	<70% possible	80% possible	90% possible	100% possible
RC	6	10% affection	5% affection	2% affection	2% affection
ACH	4	7% fail	4% fail	1% fail	0% fail

Table 4 Result of pair-wise comparison conducted by domain expert

	ETI	PT	CO	CR	RC	ACH	Weights
ETI	1	4	3	1	2	2	0.281
PT	1/4	1	1	1/4	1/2	1/3	0.069
CO	1/3	1	1	1/2	1/2	1/2	0.087
CR	1	4	2	1	2	1	0.237
RC	1/2	2	2	1/2	1	1	0.148
ACH	1/2	3	2	1	1	1	0.178

Step 2) CBAM+AHP and CBAM+ANP. In CBAM+AHP and CBAM+ANP stakeholders conduct pair-wise comparison for determining relative importance of each scenario. Table 4 is the result of pair-wise comparison conducted by domain expert and the computed weights do the same role as votes in CBAM. Table 5 shows the result of combining relative importance assigned by all the stakeholders who take part in the decision.

Step 3) As mentioned earlier, four ASs were developed during an ATAM. The ASs below describe the four variations on how to store inspection result and how to deploy the POI data.

- AS_IS_DD: Indirect storage and distributed deployment, where the inspection is executed by the distributed hardware system with each POI data replica and the result are stored to the one management database through DB management component indirectly.
- AS_IS_CD: Indirect storage and centralized deployment of POI data, where POI data for inspection is centralized and the result is stored through DB manager component indirectly.
- AS_DS_DD: Direct storage and distributed deployment of POI data, where POI data for inspection is distributed and the result is stored directly, not through DB manager.
- AS_DS_CD: Direct storage and centralized deployment of POI data, where POI data for inspection is centralized and the result is stored directly.

Step 4 ~ Step 7) CBAM. In CBAM, stakeholders evaluate scenarios and their ASs to obtain the evaluation results in Table 6.

Step 4 ~ Step 7) CBAM+AHP. To apply CBAM+AHP, stakeholders select their decisions by conducting pair-wise comparisons for the collected scenarios and developed ASs. The synthesized results are in Table 7. The inconsistency rate of CBAM + AHP is between 0.01 and 0.03.

Table 5 Combined weights for quality requirement scenarios

	Domain expert	Analyzer	Architect	Developer	Combined weight
ETI	0.281	0.290	0.454	0.167	0.308
PT	0.069	0.290	0.095	0.171	0.145
CO	0.087	0.160	0.056	0.345	0.148
CR	0.237	0.088	0.114	0.097	0.138
RC	0.148	0.100	0.210	0.110	0.147
ACH	0.178	0.072	0.071	0.110	0.113

Table 6 The results of cost benefit analysis using CBAM

AS	Benefit (Step 4)	Cost (Step 5)	ROI (Step 6)	Rank
AS_IS_DD	390	4375	0.09	4
AS_IS_CD	400	2893	0.14	3
AS_DS_DD	490	3348	0.15	2
AS_DS_CD	850	2565	0.33	1

According to Saaty (Saaty 1990) this means that the decisions are consistent. (See "Appendix" for the details of calculation for the consistency rate.)

Step 4 ~ Step 7) CBAM+ANP. Now we group all the derived vector weights as columns in the appropriate positions of the supermatrix. The factors are listed systematically so that the right vectors are listed to indicate the impact of the relevant factors on the left on the factors at the top. Synthesis using the SuperDecisions software (SuperDecision 2007) produced the results in Table 8. The inconsistency rate of CBAM+ANP is between 0.01 and 0.03 (See "Appendix").

4.4 Findings

We expected that the difference among scores for ASs of each method would be big. The AS selected for rank 1 was the same for all three methods. However, the ranks of the remaining ASs for the methods are different. Moreover, the differences between their scores are too small to have any significance. This is because the pair-wise comparison scores were in between 1 and 3, which means equal importance or moderate importance. So, though it was a consistent decision, it did not give any significant differences to the final combined results.

CBAM uses uncertainty estimation while AHP and ANP use inconsistency rate for reviewing the inaccuracies of decision making. In the case of the uncertainty estimation, if two promising alternatives are too close to differentiate one from the other then additional efforts are needed. (CBAM recommends that the probability of one being greater than the other be between 0.4 and 0.6.) However, the uncertainty estimation is made by stakeholders and so another kind of inaccuracy is thereby introduced.

But the inconsistency rate of AHP and ANP is determined by only using the existing pair-wise comparison results. So another source of inaccuracy is not injected. AHP and ANP consider that the decision is not consistent in case the inconsistency rate is more than 0.1. Besides, AHP and ANP support analyzing the priority changes of alternatives depending on changes of decision information (i.e. weights of criteria) through the

Table 7 Synthesized results matrix of CBAM+AHP

ASs	Benefit (Step 4)	Cost (Step 5)	ROI (Step 6)	Rank
AS_IS_DD	0.232	0.467	0.50	4
AS_IS_CD	0.254	0.210	1.21	2
AS_DS_DD	0.238	0.199	1.20	3
AS_DS_CD	0.277	0.124	2.23	1

Table 8 Limit SuperDecision matrix of CBAM+ANP

ASs	Ideals	Normals	Raw (Steps 4 and 5)	Rank
AS_IS_DD	0.286766	0.118192	0.059096	4
AS_IS_CD	0.513527	0.211653	0.105827	3
AS_DS_DD	0.625973	0.257999	0.128999	2
AS_DS_CD	1.000000	0.412156	0.206078	1

sensitivity analysis. When a decision is critical and likely to be sensitive to weight changes, sensitivity analysis should be conducted. A decision is sensitive if the ranking of ASs can change due to a slight change of weights. Stakeholders who take part in the decision, including domain experts, can make such changes and perform sensitivity analysis. As we noticed in Tables 7 and 8, the final weights of ASs are too close. So we performed sensitivity analysis to analyze the changes of ranking in accordance with the slight changes of weight.

Figure 10 depicts the results of dynamic sensitivity in the decision making using CBAM+AHP. It shows that the highest priority alternative is AS_DS_CD. But when we increase the weight of scenario ETI to 50% (Fig. 11), the result of the sensitivity analysis says that the best alternative addressing ETI is AS_IS_DD. That is to say, the changes of the weight for ETI influence the weight of PT and ABH and that gives an impact to the scores of AS_DS_CD. Through these sensitivity analyses, decision makers can make their decisions much considerately.

A cause of high calculation complexity in ANP is that ANP has to compute supermatrix for considering inter-dependencies among clusters. The interpolation method in AHP and ANP using the consistency check gives more confidence than CBAM, which uses utility curve. AHP and ANP can check inconsistency more easily than the CBAM's uncertainty estimation because uncertainty estimation can be calculated based on the probability distribution of a ROI score.

AHP and ANP are techniques for obtaining information for decision making. In general, factors that have influence on decision might have complex inter-relationships between them. However, since AHP models the relationships among these factors using hierarchy,

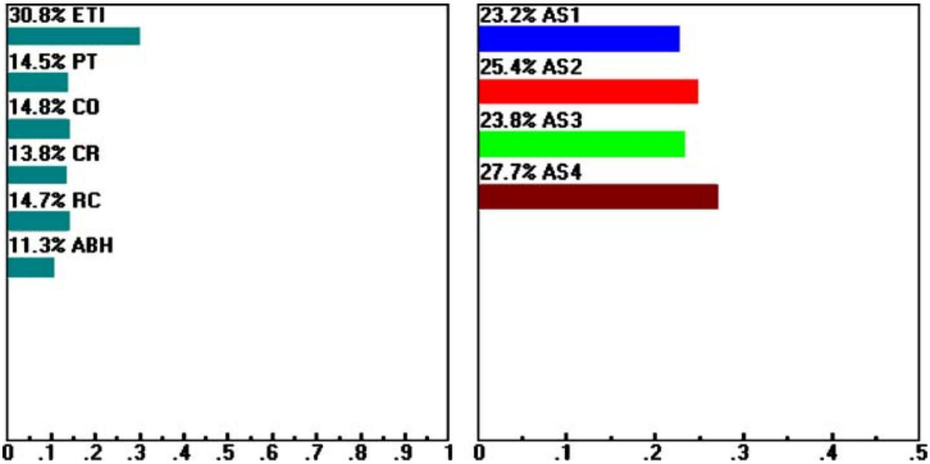


Fig. 10 Dynamic sensitivity for goal

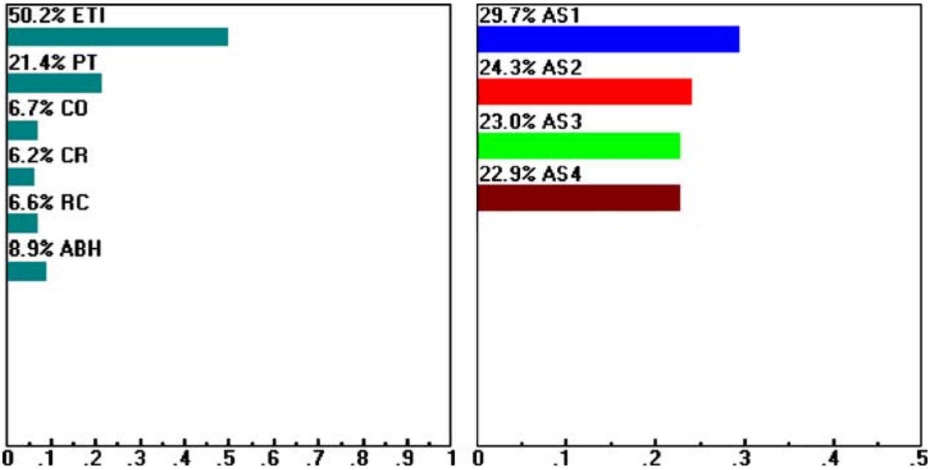


Fig. 11 Dynamic sensitivity after increasing the weight of ETI

we can reflect only the relationships between factors adjacent to each other on the decision hierarchy (cf. Figs. 2 and 7). On the contrary, because ANP has a network structure (cf. Figs. 4 and 9), ANP allows us to model any complex inter-relationships. That is, modeling capability of ANP is superior to AHP. As we explain in Sections 3.2 and 3.3, AHP cannot consider inter-dependencies among benefit, cost, scenarios, and architectural strategies though these relationships have to be considered to decide an architecture strategy. This is the reason why we developed another variation from the generic process by applying ANP. But in our experiment the results of applying CBAM+ANP was as good as those for CBAM+AHP and the difference in evaluation capability between AHP and ANP was not exposed. However, we can get information for deciding AS more easily by using AHP than by using ANP and when we need more sophisticated decision making for complex decision making problem we have to use ANP.

5 Related Works

AHP was used to evaluate architecture (Al-Naeem et al. 2005; Lee et al. 2006; Kim et al. 2007; Svahnberg et al. 2003; Svahnberg 2004), design model (Bretthall and Wohlin 2002), and system (Wallin et al. 2007; Liu and Fan 2007) by transforming qualitative attributes into quantitative forms. ANP was used for the same purpose but usually was used to evaluate more complex problems such as a complex e-business system and network security (Yi et al. 2005; Raisighani et al. 2007).

Al-Naeem et al. (Al-Naeem et al. 2005), Zhu et al. (Zhu et al. 2005), J. H. Lee et al. (Lee et al. 2006), C. K. Kim et al.(Kim et al. 2007), and Svahnverg et al. (Svahnberg et al. 2003; Svahnberg 2004) are researches closely related to the work in this paper. The goals of Al-Naeem et al. (Al-Naeem et al. 2005) and Svahnverg et al. (Svahnberg et al. 2003; Svahnberg 2004) are similar to that of Zhu, et al. (Zhu et al. 2005) in that both tried to evaluate architectural alternatives based on quality attributes. But Zhu et al. utilized the evaluation results for analyzing tradeoff and sensitivity between quality attributes and design alternatives. Al-Naeem et al. (Al-Naeem et al. 2005) is quite similar to Svahnverg et al. (Svahnberg et al. 2003; Svahnberg 2004) but they are different in that Svahnberg et al. assumes that a small set of architecture candidates have been created and Al-Naeem et al. does not. In this aspect, the work of Svahnberg et al. (Svahnberg et al. 2003; Svahnberg 2004) is similar to ours, but they do not differentiate benefit and cost, i.e. cost was regarded a one of the quality attributes and benefit was not dealt with explicitly, while we separately treat both of them. Also Svahnberg et al. (Svahnberg et al. 2003) considers relative preference of architecture candidates with respect to quality attributes but that is not a consideration of interdependency between two levels of AHP hierarchy but the approach by Svahnberg et al. uses the synthesis method after conducting pair-wise comparison separately. And these works differ from ours in that our work makes up the deficiency of CBAM using AHP and ANP.

Some researches (Nord et al. 2003; Kazman et al. 2004) have utilized CBAM but only as a method for achieving their goals. However, because CBAM requires the stakeholders to make a consensus or vote for collecting data for decision making, it is difficult to accurately reflect the stakeholders' knowledge in the process. C. K. Kim et al. (Kim et al. 2007) utilized AHP to reduce these subjective errors and the number of steps. But they did not clarify what processes of CBAM are substituted by AHP and how and why. Especially they did not consider that because AHP assumes that criteria and alternatives are orthogonal and have no feedback between them, we cannot reflect all processes of CBAM well.

For complimenting and extending the results of C. K. Kim et al., this paper presented a method utilizing ANP. This was done by first developing a generic software architecture evaluation process that generalizes CBAM so that various specific methods can be obtained by instantiating generic steps with concrete techniques.

6 Conclusion

AHP and ANP are widely used in industry and government by decision makers for organizing their thoughts and experiences to reach decisions. AHP and ANP are systematic ways to develop overall priorities by simple pair-wise comparisons. In this paper we presented the generic process to cost benefit analysis of architectural strategies that is based on CBAM and developed specific methods of architecture cost benefit analysis utilizing AHP and ANP techniques. We named these new methods respectively as CBAM+AHP, and CBAM+ANP and conducted an experiment to compare CBAM, CBAM+AHP, and CBAM+ANP.

Most techniques used in CBAM rely on stakeholders' consensus and voting. So CBAM is a heavy process that is costly to put into practical use. On the other hand, because AHP and ANP are based on pair-wise comparison we can not only reflect the knowledge of experts more accurately but also reduce misjudgments. AHP and ANP helped us check inconsistency in decisions, interpolate the determined priorities and estimate costs more easily than CBAM. Moreover, the consistency check used in CBAM+AHP and CBAM +ANP gives more confidence than the interpolation method of CBAM. Their inconsistency check also can be used for uncertainty and risk assessment for the second iteration of CBAM for more sophisticated and realistic decision-makings. In addition, we learned from this experiment that for more critical decisions CBAM+ANP is a better method to choose than CBAM+AHP because we can consider inter-dependencies among benefit, cost, scenarios, and architectural strategies.

We developed variations of CBAM by considering only the factors that CBAM originally used. However there might be unrevealed factors that should be considered in architectural strategy decision. Because the cost benefit analysis process that we presented in this paper is generic, new cost benefit analysis techniques other than the three methods we examined here can be derived by adopting various different constituent techniques. The new methods will have capabilities and characteristics different from those of CBAM+AHP and CBAM+ANP. An interesting future research direction is to explore further variations beyond CBAM+AHP and CBAM+ANP and validate their effects through larger case studies. In the future we are going to conduct quantitative comparison of the methods introduced in this paper to analyze the required efforts in order to help select appropriate methods according to the complexities of decision making.

Acknowledgements This work was partially supported by Defense Acquisition Program Administration and Agency for Defense Development under contract (2008-SW-14-IJ-02) and by the Industrial Technology Development Program funded by the Ministry of Commerce, Industry and Energy (MOCIE, Korea).

The authors are thankful to Jae-ha Song of NHN Corp., Gu-yul Noh of Samsung SDS, and Jong-gul Park of Tmaxsoft, for participating in the empirical study conducted for this research.

Appendix

In decision making it is important to know how consistent the decisions are. In this section we present how to compute the inconsistency rate using pair-wise comparison results (Table 9).

We obtain the normalized matrix through each row divided by column total, its row sums, and the percentages of relative overall priorities (i.e. weights) from the pair-wise comparison result of domain expert (See Table 10).

And then we calculate average of each scenario by multiplying weight of scenario to pair-wise comparison results with remaining scenarios i.e. the row of Table 9.

$$0.281 \times \begin{bmatrix} 1 \\ 0.25 \\ 0.33 \\ 1 \\ 0.5 \\ 0.5 \end{bmatrix} + 0.069 \times \begin{bmatrix} 4 \\ 1 \\ 1 \\ 4 \\ 2 \\ 3 \end{bmatrix} + 0.087 \times \begin{bmatrix} 3 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \end{bmatrix} + 0.237 \times \begin{bmatrix} 1 \\ 0.25 \\ 0.5 \\ 1 \\ 0.5 \\ 1 \end{bmatrix} + 0.148 \times \begin{bmatrix} 2 \\ 0.5 \\ 0.5 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

$$+ 0.178 \times \begin{bmatrix} 2 \\ 0.33 \\ 0.5 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Table 9 Results of pair-wise comparison conducted by domain experts

	ETI	PT	CO	CR	RC	ACH
ETI	1	4	3	1	2	2
PT	1/4	1	1	1/4	1/2	1/3
CO	1/3	1	1	1/2	1/2	1/2
CR	1	4	2	1	2	1
RC	1/2	2	2	1/2	1	1
ACH	1/2	3	2	1	1	1
Total	3.58	15	11	4.25	7	5.83

Table 10 Normalized matrix, row sums, and weights

	ETI	PT	CO	CR	RC	ACH	Sums	Weights
ETI	0.28	0.27	0.27	0.24	0.29	0.34	1.68	0.281
PT	0.07	0.07	0.09	0.06	0.07	0.06	0.41	0.069
CO	0.09	0.07	0.09	0.12	0.07	0.09	0.53	0.087
CR	0.28	0.27	0.18	0.24	0.29	0.17	1.42	0.237
RC	0.14	0.13	0.18	0.12	0.14	0.17	0.89	0.148
ACH	0.14	0.20	0.18	0.24	0.14	0.17	1.07	0.178

Now we obtain eigenvector λ by dividing this result by weights vector.

$$\lambda = \begin{bmatrix} 6.074733 \\ 6.070048 \\ 6.105364 \\ 6.084388 \\ 6.060811 \\ 6.092697 \end{bmatrix}$$

And we calculate maximum eigenvalue λ_{max} by averaging the eigenvector λ:

$$\lambda_{max} = 6.08134$$

Consistency index can be obtained by calculating $(\lambda_{max}-n)/(n-1)$ as follows

$$CI = \frac{(6.08134 - 6)}{(6 - 1)} = 0.016$$

In case the size of matrix is 6, RI[2] is 1.24 and the consistency ratio is

$$CR = 0.016/1.24 = 0.013.$$

As the result, inconsistency rate of domain expert is 0.013. So according to the paper (Saaty 1990) we can decide that the decision was a consistent one.

References

Al-Naeem T, Gorton I, Rabhi MA, Benatallah B (2005) A quality-driven systematic approach for architecting distributed software applications. Proc. Int. Conf. Software Engineering (8th ICSE), St. Louis, USA, pp 244–253

Bratthall L, Wohlin C (2002) Is it possible to decorate graphical software design and architecture models with qualitative information?—an experiment. IEEE Trans Softw Eng 28(12):1181–1193

[2] According to T. L. Saaty (Saaty 1999), if numerical judgments were taken at random from the scale 1/9, 1/8, 1/7, ..., 1/2, ..., 1, 2, ..., 9, then we would have the following average consistencies for different-order random matrices:

Matrix Size	1	2	3	4	5	6	7	8	9	10
RI	0.00	0.00	0.52	0.89	1.11	1.25	1.35	1.40	1.45	1.49

Clements P, Kazman R, Klein M (2002) Evaluating software architectures methods and case studies. Addison Wesley, Boston, MA

ExpertChoice (2007). Expert Choice11.5™. Available at http://www.expertchoice.com

Kazman R, Asundi J, Klein M (2002) Making architecture design decisions: an economic approach (CMU/SEI-2002-TR-035, ESC-TR-2002-035). Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA

Kazman R, In H, Chen H (2004) From the requirements to negotiation to software architecture decisions. Proc. Int. Conf. Software Engineering Research, Management and Applications (SERA), Los Angeles, CA, USA

Kim CK, Lee DH, Ko IY, Baik JM (2007) A lightweight value-based software architecture evaluation. Proc ACIS Int Conf Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD) 2:646–649

Lee JH, Kang SW, Cho JH, Kim JS (2006) Architecture evaluation utilizing CBAM and AHP. Korea Inf Process Soc Trans 13-D(5):683–690

Liu B, Fan Y (2007) A new performance evaluation model and AHP-based analysis method in service-oriented workflow. Proc. Int. Conf. Grid and Cooperative Computing (GCC), 685–692

Nord RL, Barbacci MR, Clements P, Kazman R, Klein M, O'Brien L et al (2003) Integrating the architecture tradeoff analysis method (ATAM) with the cost benefit analysis method (CBAM) (CMU/SEI-2003-TN-038). Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA

Raisighani M, Meade SL, Schkade LL (2007) Strategic e-business decision analysis using the analytic network process. IEEE Trans Eng Manage 54(4):673–686

Saaty TL (1980) The analytic hierarchical process. McGraw-Hill, New York

Saaty TL (1990) Multicriteria decision making: the analytic hierarchical process, I, AHP series. RWS, Pittsburgh, PA

Saaty TL (1999) Decision making for leaders. RWS, Pittsburgh, PA

Saaty TL (2005) Theory and applications of the analytic network process. RWS, Pittsburgh, PA

Saaty TL, Vargas LG (1982) The logic of priorities. Kluwer-Nijhoff, London

Saaty TL, Vargas LG (1994) Decision making in economic, political, social and technological environments with the analytic hierarchy process. RWS, Pittsburgh, PA

SuperDecision (2007) Creative decisions foundation. SuperDecision, available at http://www.superdecisions.com

Svahnberg M (2004) An industrial study on building consensus around software architectures and quality attributes. Inf Softw Technol 46(12):805–818

Svahnberg M, Wholin C, Lundberg L (2003) A quality-driven decision-support method for identifying software architecture candidates. Int J Softw Eng Knowl Eng 13(5):547–573 doi:10.1142/S0218194003001421

Wallin P, Froberg J, Axelsson J (2007) Making decisions in integration of automotive software and electronics. Proc. Int. Workshop on Software Engineering for Automotive Systems (ICSE Workshops SEAS '07)

Yi JN, Meng WD, Ma WM, Du JJ (2005) Assess model of network security based on analytic network process. Proc Int Conf Mach Learn Cybern 1:27–32 doi:10.1109/ICMLC.2005.1526914

Zhu L, Aurum A, Gorton I, Jeffery R (2005) Tradeoff and sensitivity analysis in software architecture evaluation using analytic hierarchy process. Softw Qual J 13:357–375 doi:10.1007/s11219-005-4251-0

Jihyun Lee received the BS degree in Information and Communications Engineering from the Chonbuk National University and the MS and the PhD degrees in Computer Sciences from the Chonbuk National

University. She is currently a research assistant professor in Software Technology Institute at ICU. Her research interests include SPL, software architecture, business process maturity, and SOA.

Sungwon Kang received his BA from Seoul National University, Korea in 1982, and received his MS and PhD in computer science from the University of Iowa, USA in 1989 and 1992, respectively. From 1993, he was a principal researcher of Korea Telecom R & D Group until October 2001 when he joined Information and Communications University. He is currently an associate professor at the university. Since 2003, he has been an adjunct faculty member of Carnegie-Mellon University, USA, for the Master of Software Engineering Program. He served as a co-chair of 1997 International Workshop on Testing of Communication Systems and 2001 International Conference on Formal Techniques for Networked and Distributed Systems. His current research areas include software architecture, software modeling and analysis, software testing, and formal methods.

Chang-Ki Kim received BS and MS degrees in Electronics Engineering from the Pusan National University, Pusan, Korea in 1995 and 1997, respectively. He also received MS degree in software engineering from Carnegie-Mellon University in 2005. He worked as an Engineer in the Samsung Electronics from 1997 to 2000. Since 2000, he has been a senior member of engineering staff at Electronics and Telecommunications Research Institute (ETRI). His main research interests include mobile communication, radio network protocols design and performance analysis of mobile communication system.