International Journal of Software Engineering and Its Applications
Vol.8, No.3 (2014), pp.89-96
http://dx.doi.org/10.14257/ijseia.2014.8.3.09

Making a Decision about Importance Analysis and Prioritization of Use Cases through Comparison the Analytic Hierarchy Process (AHP) with Use Case Point (UCP) Technique

Bokyung Park[1] and Robert Youngchul Kim[2]

SE Lab., Dept. Of CIC (Computer and Information Communication)
Hongik University, Sejong Campus, 339-701, Korea
[1]Bkpark@mail.hongik.ac.kr, [2]bob@hongik.ac.kr

Abstract

The existing studies suggested the Importance and Prioritization Methods which for exactly reflecting customers' requests from the proper requirements [1,2]. However, this method decides ERRC (Eliminate, Reduce, Raise, Create) of Use Case by the customers' subjective judgment. Moreover, such a previous approach can't present a verification method for logical consistency about customers' judgment [3]. In order to more objectively solve such a problem, this paper suggests another method for the importance extraction and prioritization of Use Case with AHP mechanism. With this suggested method, it compares and verifies the data extracted from each method of Use Case Point or AHP, and also judges whether the customers' idea is consistently determined through Consistency Assessment of AHP or not. We show a Case Study to adapt this to Automobile Goods Management System, for the exact customers' requests.

Keywords: *Hybrid Approach, Use Case Point (UCP), Goal oriented Requirements Engineering Process (GoRE), Analytic Hierarchy Process (AHP)*

1. Introduction

The existing studies suggested the *Use Case based Goal oriented Requirements Engineering (GoRE)*[1,2]. This method has the goal to reflect the exact customers' requests in developing software. However, in this method, customers subjectively judged and decided ERRC of Use Case according to the judgment standard of Hybrid Approach [1,2]. Moreover, it is very difficult for customers to judge the extracted Use Case whether it is correct or not. In order to solve such a problem, this paper extracts the importance of Use Case with AHP, and prioritizes them. To do this work, it extracts the importance of *AHP-based Use Case* which is compared with the importance of the existing *Use Case Point-based Use Case*. Also, it judges whether the major idea is consistently determined or not through the estimation of AHP consistency. This paper is composed as follows: Chapter 2 looks at the related works. Chapter 3 discusses the method of ERRC decision adapting AHP method and Chapter 4 mentions Conclusion and future studies.

2. Related Works

AHP, which was developed by T. Saaty, performs the pairwise comparison between factors composing the decision hierarchy. It accomplishes the assessor's consistency estimation with such a pairwise comparison. Accordingly, it can decide human being's subjective judgment reasonably [4]. The UCP technique, which was developed by Gustav

ISSN: 1738-9984 IJSEIA

Karner in 1993, estimates the software size of the object oriented systems. The UCP method is based on similar principles as Function Point (FP) software estimation method, but was created to solve for the specific needs of object oriented systems and system requirements based on use cases.

3. Decision Method of Use Case based on ERRC Using AHP Technique

We suggest *use case oriented requirement process* [2,3] which extracts and prioritizes requirements based on value-innovative requirement engineering (ViRE)[1] in ① part of Fig 1. We verify this extracted data with UCP in here. Next, it will be finally decided the customer requirements with hybrid approach. In Fig. 1, ② is the process to adapt the result of ① to AHP technique, to decide ERRC, and to perform the consistency assessment. Figure 1 shows the decision process of *Use Case-based using AHP technique* [1,2].

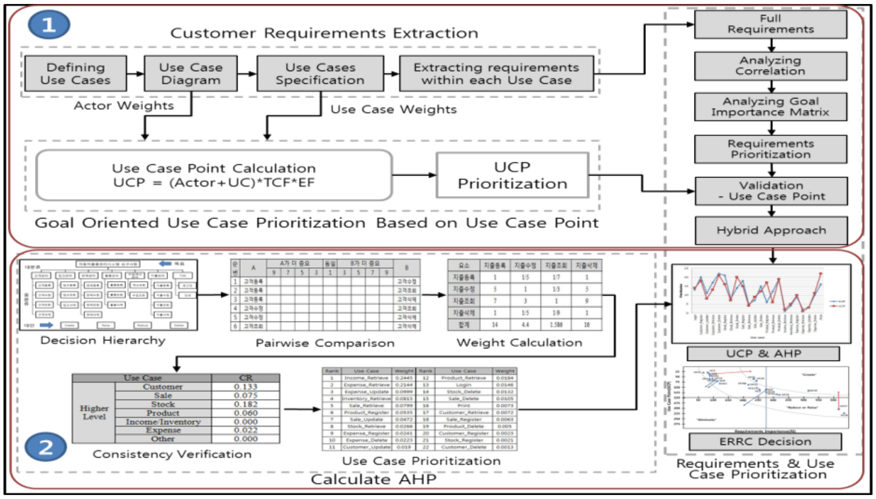


Figure 1. Decision Process of Use Case-based ERRC using AHP Technique

Step 1. Decision Hierarchy

In the first step of decision hierarchy, it arranges Use Case extracted in ① in the hierarchical order like Goal, Criteria, Sub-criteria, and Alternatives. In the top level of hierarchy, it lies the aim of the most comprehensive decision making. It comprises of various elements affected to make a decision in other level of the hierarchy [4]. This elements are looked upon the extracted use cases in the ① part of Figure 1. In this decision-making system, it is composed of the standard form like Figure 2 [4]. In the case study, we extracts total 22 use cases

In level 1(Goal), it indicates and chooses the importance of automobile goods management system. In level 2(Criteria), it classifies use cases with the common properties of the extracted use cases into the common area. The customer register/update/retrieve/delete is classified into customer management, and the stock register/retrieve/delete classified into the stock management. In level 3(Sub-Criteria), it is arranged use cases associated with level 2.

Step 2. Pairwise Comparison

In the second step, it collects data through participants' survey, and performs the pairwise comparison. To execute pairwise comparison, we assign *n* elements (use cases) on the column of n*n matrix. According to any criteria, it executes pairwise comparison with all use cases. The weight values used by the pairwise comparison are collected through mainly questionnaire investigation, which assigns 1~9 score for measurement range of the measure criteria.

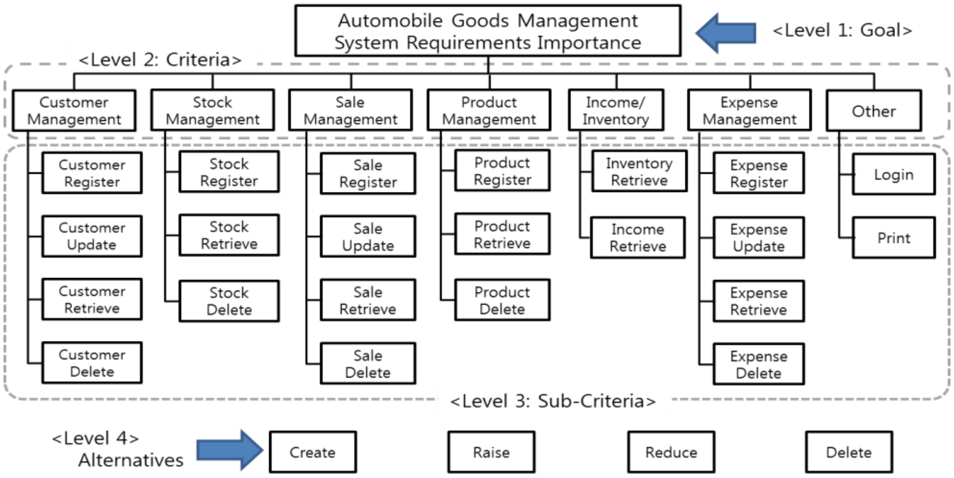


Figure 2. The AHP Hierarchy of Automobile Goods Management System

Step 3. Weight Calculation

It executes the pairwise comparison with the collected data based on the questionnaire investigation. Calculation method of the weight values is composed of three steps.

1) Assigns weight value for the pairwise comparison into matrix, then sums each column. It is the criteria of the first left row of the pairwise comparison matrix. In Table 1, if ‘expense register’ is not more important than ‘expense update’, then we input 1/7 score. This means relatively to get ‘expense register’ the importance of 1/7 in contrast to ‘expense update’. Like this, we make the pairwise comparison matrix to reversely be symmetric.

Table 1. Normal Procedure of Pairwise Comparison Matrix

Factor	Expense_ Register	Expense_ Retrieve	Expense_ Update	Expense_ Delete	Normalization			
Expense_ Register	1	1/5	1/7	1	0.071	0.045	0.090	0.063
Expense_ Retrieve	5	1	1/3	5	0.357	0.227	0.210	0.312
Expense_ Update	7	3	1	9	0.500	0.682	0.630	0.562
Expense_ Delete	1	1/5	1/9	1	0.071	0.045	0.070	0.063
Total	14	4.4	1.586	16	0.999	0.999	1	1

2) Normalizes the pairwise comparison matrix. We divide the sum of each column into each column in Table 1. This means relatively to get ’expense register’ the importance of 1/7 in contrast to ‘expense update’.

Each row (Expense_Register, Expense_Retrieve, Expense_Update, Expense_Delete) on 'Expense Retrieve' of the column of matrix is calculated as follows: (1/7)/1.586=0.090, (1/3)/1.586= 0.210, 1/1.586=0.630, (1/9)/1.586= 0.070

The Sum of each row (0.090+0.210+0.0630+0.070) is 1.

3) Calculates the weight value of the pairwise comparison matrix. The weighted values are calculated with the average of each column on the normalized matrix. The extracted weight value is relatively level of importance. In the extracted weight values, the weighted values are 0.067 (Expense_Register), 0.0277 (Expense_Update), 0.564 (Expense_Retrieve), and 0.062 (Expense_Delete). The 'Expense_Management' Use case has assigned the importance in the weighted order as follows: 0.564 (Expense_Retrieve), 0.0277 (Expense_Update), 0.067 (Expense_Register), and 0.062 (Expense_Delete).

Step 4. Consistency Verification

It executes the consistency verification at this stage. The reason why to verify the consistency should judge whether it consistently calculates values of the pairwise comparison matrix or not. The Consistency Verification Method uses the formula in Table 3, and calculates the Consistency Index (CI) and Consistency Ratio (CR) [4].

Table 3. Consistency Verification

Consistency Index(CI)	Consistency Ratio(CR)
$(\mathrm{CI}) = (\lambda_{max} - n)/(n-1)$ λ_{max} : *Eigen Value Method*	CR = CI/RI(RI: Random Index) CR<0.1: Keep A Consistency CR≤0.2 : Permissible level of Consistency

RI of the Consistency Ratio(CR) means Random Index and describes permissible limit of consistency. Table 4 shows calculation method of the consistency.

Table 4. Calculation Method of the Consistency Verification

element	1 Step of consistency verification	2 Step of consistency verification
Expense_ Register	$(1 \times 0.067) + \left(\frac{1}{5} \times 0.277\right) + \left(\frac{1}{7} \times 0.564\right) + (1 \times 0.062) = 0.265$	0.265/0.067=3.955
Expense_ Retrieve	$(5 \times 0.067) + (1 \times 0.277) + \left(\frac{1}{3} \times 0.564\right) + (5 \times 0.062) = 1.11$	1.11/0.277=4.007
Expense_ Update	$(7 \times 0.067) + (3 \times 0.277) + (1 \times 0.564) + (9 \times 0.062) = 2.422$	2.422/0.564=4.294
Expense_ Delete	$(1 \times 0.067) + \left(\frac{1}{5} \times 0.277\right) + \left(\frac{1}{9} \times 0.564\right) + (1 \times 0.062) = 0.247$	0.247/0.062=3.983

If the Consistency Ratio is less than 0.1, we understand to keep the consistency. If CR less equal 0.2, it is permissible consistency. If CR is greater than 0.2 and not keeps the consistency, it should re-investigate them [4]. The consistency ratio values are less than 0.1 in 'expense management'. Therefore, we recognize to execute the pairwise comparison in that the response maintains completely consistency.

Table 5 shows the evaluation result with the weight value of the pairwise comparison and the consistency in this way.

Table 5. Consistency Ratio

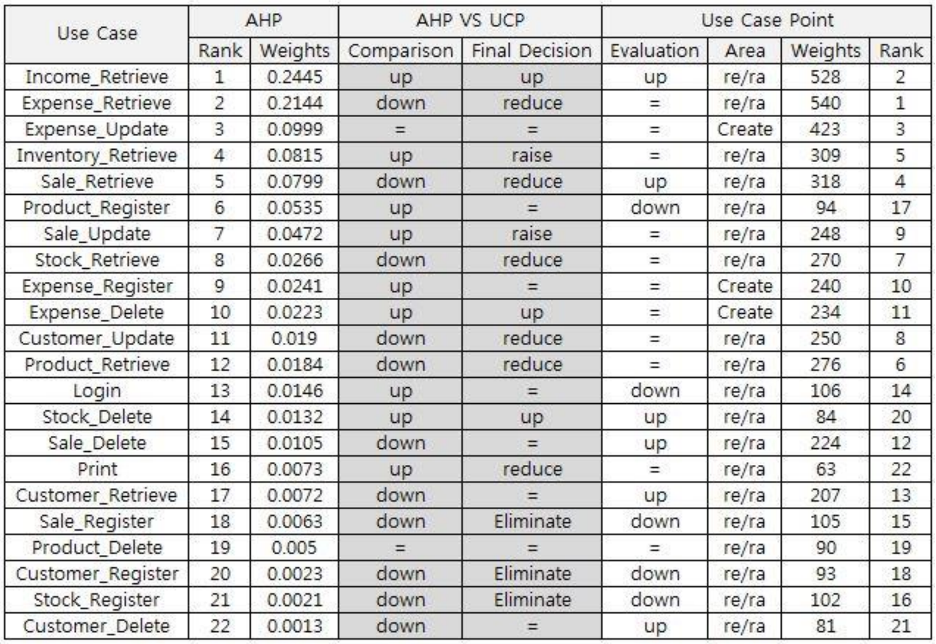
Classification	CR	Classification	CR
Customer_Management	0.133	Income/Inventory	0.000
Sale_Management	0.075	Expense_Management	0.022
Stock_Management	0.182	Other	0.000
Product_Management	0.060		

The Consistency Ratio of Case Study is less than 0.1, which is suggested by Satty, in the rest of Use Case except customer and storage management. That is, it is shown to have high consistency. However, it can be understood that they have a consistency ratio less than 0.2 so that it has consistency in an acceptable degree. Accordingly, it can be judged that the extracted values keep the consistency in the Use Case.

Step 5. Use-Case Prioritization

In the fifth step, it compares the weights of the third step, and prioritizes Use Case.

Table 6. AHP&UCP Comparison



Use Case	AHP		AHP VS UCP		Use Case Point			
	Rank	Weights	Comparison	Final Decision	Evaluation	Area	Weights	Rank
Income_Retrieve	1	0.2445	up	up	up	re/ra	528	2
Expense_Retrieve	2	0.2144	down	reduce	=	re/ra	540	1
Expense_Update	3	0.0999	=	=	=	Create	423	3
Inventory_Retrieve	4	0.0815	up	raise	=	re/ra	309	5
Sale_Retrieve	5	0.0799	down	reduce	up	re/ra	318	4
Product_Register	6	0.0535	up	=	down	re/ra	94	17
Sale_Update	7	0.0472	up	raise	=	re/ra	248	9
Stock_Retrieve	8	0.0266	down	reduce	=	re/ra	270	7
Expense_Register	9	0.0241	up	=	=	Create	240	10
Expense_Delete	10	0.0223	up	up	=	Create	234	11
Customer_Update	11	0.019	down	reduce	=	re/ra	250	8
Product_Retrieve	12	0.0184	down	reduce	=	re/ra	276	6
Login	13	0.0146	up	=	down	re/ra	106	14
Stock_Delete	14	0.0132	up	up	up	re/ra	84	20
Sale_Delete	15	0.0105	down	=	up	re/ra	224	12
Print	16	0.0073	up	reduce	=	re/ra	63	22
Customer_Retrieve	17	0.0072	down	=	up	re/ra	207	13
Sale_Register	18	0.0063	down	Eliminate	down	re/ra	105	15
Product_Delete	19	0.005	=	=	=	re/ra	90	19
Customer_Register	20	0.0023	down	Eliminate	down	re/ra	93	18
Stock_Register	21	0.0021	down	Eliminate	down	re/ra	102	16
Customer_Delete	22	0.0013	down	=	up	re/ra	81	21

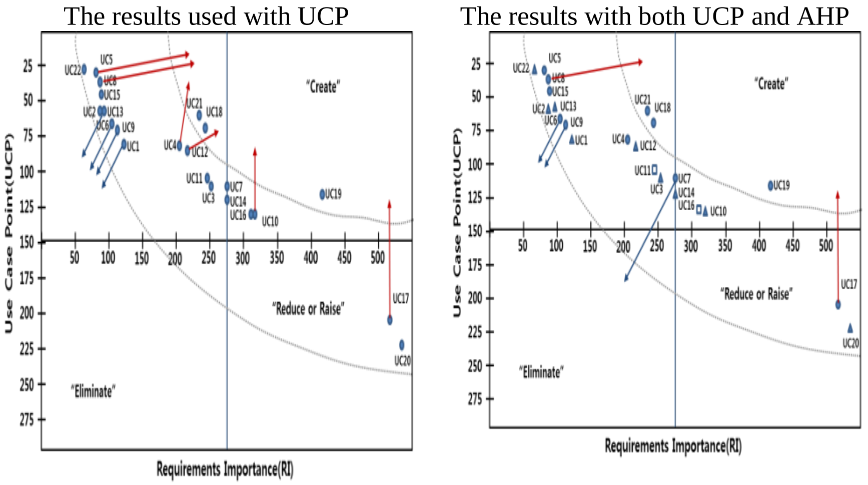
Step 6. UCP&AHP

In the sixth step, it compares and verifies the importance of Use Case derived from Use Case Point(UCP) and AHP techniques. The previous research extracted use case prioritization based on Use Case Point. But we cannot verify whether it did exactly reflect the customer requirements or not. Therefore we want to exactly reflect the customer requirement through verifying and comparing with the values of two different ways. Table 6 shows the results of step 5,6.

We can make the result extracted with AHP in left side of Table 6, and also other result with UCP in right side of it. On AHP, we extract the priority of use case with the product of the weight value of level 2 by the detained item's weight value of level 3. We finally decide to compare the extracted results with the previous UCP.

Step 7. The ERRC Decision

In the last step, it decides ERRC of Use Case based on AHP&UCP Results in Figure 3. The left picture is the result using UCP technique, and the right one shows the result of comparing AHP with UCP. Figure 3 shows AHP&UCP Results. In total, two Use Cases moved to the area of Create (UC 8, 17). On the other hand, the total number of Use Cases moved toward the area of Eliminator is 3(UC 6, 7, 9). The Use Cases of Raise Function are UC 11 and UC 16, but the Use Cases of Reduce Function are UC 2, 3, 10, 12, 13, 14, 20, 22.


The results used with UCP

The results with both UCP and AHP

Figure 3. The Final Decision Results from UCP&AHP Comparison

4. Conclusion

In order to extract and prioritize the exact requests, this show to compare and analyze the methods of AHP and UCP. The existing method decides the customers' requests with their subjective judgments. Therefore, it is not an objective one, and also cannot present a logically consistent verification method about customer's judgment. It extracts and prioritizes the importance of Use Case with AHP technique to supplements the problems caused by the existing qualitative judgment. As a result, we show to compare the result of the existing UCP with the result of AHP and analyzes them. We can verify them whether it is able to refine requests to exactly reflect the customer's requests or not.

Acknowledgements

This work was supported by the IT R&D Program of MKE/KEIT [10035708, "The Development of CPS(Cyber-Physical Systems) Core Technologies for High Confidential Autonomic Control Software"] and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2013R1A1A2011601)

References

[1] S. Kim and H. Peter, "VIRE: Sailing a Blue Ocean with Value-Innovative Requirements", Software IEEE, vol. 25, **(2008)**, pp. 80-87.

[2] B. Park and R. Y. Kim, "A Comparative Study on the ViRE & Refined GoRE's Process", Journal of KIISE: Computing Practices and Letters, vol. 19, no. 4, **(2013)**.

[3] B. Park, S. Moon, K. Kim, B. Kim and R. Y. Kim, "Use Case Oriented Requirements Engineering for improving the previous ViRE's Process", The 38th KIPS Fall Conference, Jeju, Korea, **(2012)**, November 22-23.

[4] G. Cho, Y. Cho and H. Kang, "The Analytic Hierarchy Process", DongHyun Books, **(2003)**.

Authors

Bokyung Park received the B.S. and M.S degree in Software Engineering from Hongik University, Korea in 2012. He is also currently Ph.D student and works Research Assistant(RA) in Dept. of Computer Information & Communication, Hongik University, Korea.. His research interests are in the areas of Goal Oriented Requirement Engineering(GoRE), User Behavior Analysis, and Testing.

R. YoungChul Kim received the B.S. degree in Computer Science from Hongik University, Korea in 1985, and the Ph.D. degree in Illinois Institute of Technology (IIT), USA in 2000. He also worked as a primary manager in LG industrial Research Center. He is currently a professor in Hongik University. His research interests are test case generation, test maturity model(TMM), and modeling based testing(MBT).