BAREMO: How to Choose the Appropriate Software Component Using the Analytic Hierarchy Process

Article · January 2002

DOI: 10.1145/568760.568893 · Source: OAI

2 authors, including:

A. Lozano-Tello
Universidad de Extremadura
82 PUBLICATIONS **756** CITATIONS

SEE PROFILE

BAREMO: How to Choose the Appropriate Software Component Using the Analytic Hierarchy Process [1]

Adolfo Lozano-Tello
Departamento de Informática
Universidad de Extremadura
Avda Universidad sn 10071 Cáceres. SPAIN
+34 927257195
alozano@unex.es

Asunción Gómez-Pérez
Facultad de Informática (LIA)
Universidad Politécnica de Madrid
Campus Montegancedo 28660 Madrid. SPAIN
+34 913367439
asun@fi.upm.es

ABSTRACT

To select a software component from several similar candidates is a complex task, since each project pursues different objectives. We intend to use the Analytic Hierarchy Process in the taking of multicriteria decisions for software component reuse. This method is called BAREMO. It will help the software engineer to make estimations which will enable him/her to choose the appropriate component. The article presents a case study of the application of the method, where a project manager assesses a certain software component in order to consider its reuse in the domain of image processing.

Categories and Subject Descriptors

D.2.13 [**Software Engineering**]: Reusable Software – *Reuse models.*

D.2.8 [**Software Engineering**]: Metrics – *Products metrics.*

General Terms

Measurement, Design.

Keywords

Reuse of software components, decision support, AHP.

1. INTRODUCTION

The reuse of previously developed knowledge, processes and software products increases productivity, reduces cost, avoid risk and enhance the quality of software applications [12]. For these reasons, the current tendency is to reuse software components (SCs) stored in repositories. It allows the software developers to examine such repositories with the purpose of determining which SC is better suited to the requirements of their project. The repositories contain -with its proper classification taxonomy- information about the stored components in order to help the user to find and understand the best alternatives.

Pressman [8] warns: "... often the use of reusable components during the planning stage is overlooked, despite it being the main concern in the software development process. It is much better to specify the requirements of software resources at the beginning. Thus, you can control the technical evaluation of alternatives and you can obtain the right components". However, you can encounter many difficulties in selecting a SC at the planning stage, due to the fact that it is not easy to carry out precise estimations of costs and time, in this initial period. Nevertheless the company management often requests these estimations before the development process, so they may be necessary to do them.

Boehm [2] states that effective software engineering requires a continuous process of goal identification, making decisions about opposing objectives, and aiming the project towards several objectives simultaneously. In software engineering, it is not usual to make decisions with a unique decision criterion. For this reason, the engineer must bear in mind several considerations in order to choose a SC from several valid similar [9] alternatives, whenever they satisfy certain domain specifications [1]. The selection depends on various factors with varying for each domain, product and production company; therefore, it is a multicriteria choice problem.

An inadequate evaluation of these factors can cause the software project to fail. Selection of the SC is based on the software engineer's knowledge and experience, but it may be "delicate" to justify their selection to their company managers (above all if the selection is wrong). This work describes BAlanced REuse MOdel (BAREMO), a method of making quantitative estimations about several objectives in order to select a SC, and thus, make rational decisions in this multicriteria problem.

This paper is organised as follows. In section 2, we present a set of general SC characteristics which have a direct impact on these decision-making issues. Section 3 describes briefly the Analytic

SEKE '02, July 15-19, 2002, Ischia, Italy.

[1] This work has been partially supported by *Junta de Extremadura* under the project 2PR01A023.

Hierarchy Process (AHP) in the taking of multicriteria decisions. Section 4 shows how we have adapted AHP in the choice of software components. Finally, section 5 presents a case study of the application of BAREMO.

2. IMPACT OF SOFTWARE COMPONENT CHARACTERISTICS ABOVE DECISION-MAKING FACTORS

The software reuse process extends to different reusable "artefacts" [7]. Apart from source code or data, project plans, cost estimates, architectures, specifications and requirements models, designs, user documentation, and techniques, human interfaces and test cases can also be reused. However, although the requirements specification for two projects differ, the same factors should be taken into account, to which different importance will be attached in each case. The first task for the people who are to decide which components to reuse will be to establish the minimum requirements for each issue to be considered for their particular project, and to be met by the reusable components.

When software developed by others or even personally is reused, not all of the functional needs of the new project are generally met. This will involve having to adapt and develop software to meet all the demands of the project, with the resulting expenditure in terms of effort and time. Apart from the production effort involved in adapting and adding to the software that is to be reused, extra work will have to be done on studying its specifications and the conceptual and implementation compatibility of the new modules with those that have been selected for reuse.

Project managers should carry out the following tasks before adapting SCs to the needs of the new system:

- Gather the candidate software components from libraries or repositories.
- Study the characteristics of each SC against the same frame of reference.
- Establish thresholds for each factor involved in the decision on whether or not to reuse an adaptable SC.
- Select the candidate that is best suited to the needs of the final product.
- Decide who is to adapt (in-house development team, supplier or another company).

In our experience, we have found how difficult it is to select and decide which SC to reuse both with regard to the reuse of functional procedures of traditional software development at companies and the development of applications based on the reuse of knowledge-based software [5]. The proposed factors provide

DIMENSIONS	FACTORS / CHARACTERISTICS	**QD**:Quality of documentation	**M**:Methodology	**TH**:Training help	**AH**:Adaptation help	**DH**:Development help	**Mo**:Modularity	**Co**:Complexity	**S**:Size	**RC**:Requirement covered	I:Interoperativity	**TQ**:Tools quality	**EH**:Ease of handling	**AL**:Acquisition license	**KM**:Kind of mantenance	**UC**:Update cost	**Res**:Human, Hw & Sw resources	**ES**:Execution speed	**OA**:Output accuracy	**A**:Applications	**Ma**:Maturity	**Cr**:Credibility
Production Time	**TT**:Training time	X	X	X			X	X	X													
	AT:Adaptation time	X	X		X		X	X	X	X	X	X										
	DT:Development time					X							X									
Cost Rating	**LP**:Licenses price													X	X	X						
	AE:Adaptation expenses				X												X					
	DE:Developm. expenses					X											X					
Product Quality	**E**:Effectiveness																X	X	X			
	R:Reliability		X												X					X	X	X
Developm. Risk	**F**:Feasibility										X									X		
	C:Capability		X		X	X																

Table 1. Impact of software components characteristics above decision-making factors.

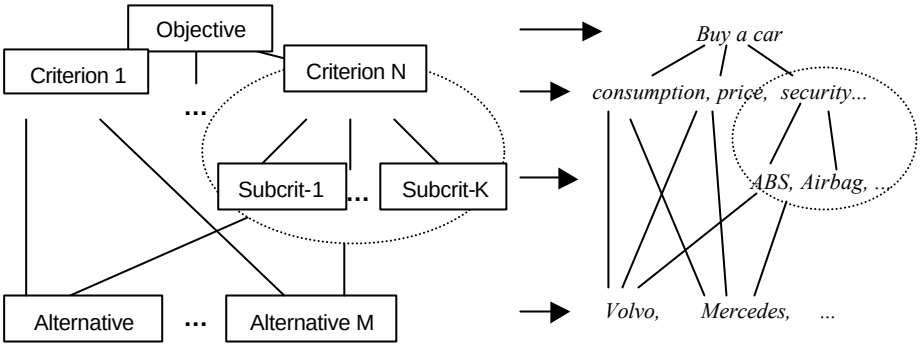

Figure 1. Hierarchical structure of the AHP method, and example of the objective "Buy a car".

an overview for undertaking the study of any SC and is designed to offer a conceptual framework with which to analyse the fitness process of the above components.

We have identified four main dimensions in the decision-making stage when using any type of reusable SC:

- **Production time**: project development time, after having selected a particular SC.
- **Cost rating**: capital investment to be made by the company in resources to be able to carry out the project with the aforesaid SC.
- **Final product quality**: assessment of the final product outputted. The characteristics of the component chosen will have an impact on final product quality.
- **Development risk**: probability of successfully outputting the final product by selecting a reusable component.

As shown in table 1, attached to each dimension, we have identified a set of factors that state the decision fundamental elements for SC selection, and for each factor we have identified a set of significant SC characteristics to consider before the SC selection process. Note that some characteristics of the reusable components can have an impact on several of the above factors and dimensions; a summary of the interrelation between characteristics, factors and dimensions is given in table 1. It is also important to mention that the SC characteristics proposed will have different impact on each selection factor, and also each factor will have different influence on the dimensions.

All project managers should weigh up each dimension, factor and characteristic to decide how important they are, relating them to the peculiarities and demands of their particular project. In [6], Gómez-Pérez and Lozano-Tello describe the relevant characteristics (showed in table 1) to select a SC. They draw up a framework of guidelines that can be an aid to project managers for considering the characteristics of SCs that will have an impact on the factors that will determine their selection. The impact of each characteristic and each factor will be linked to each project, and each development company; this assessment will depend on the subjective judgement of the person who examines the SC and the particular project. However, the proposed schema (showed in table 1) has to be kept in mind to achieve a good selection.

3. THE ANALYTIC HIERARCHY PROCESS

The Analytic Hierarchy Process (AHP) was devised by Thomas L. Saaty in the early seventies [10]. It is a powerful and flexible tool for decision-making in complex multicriteria problems. This method allows people to gather knowledge about a particular problem, to quantify subjective opinions and to force the comparison of alternatives in relation to established criteria. The method consists of the following steps:

STEP 1: define the problem and the main objective to make the decision.

STEP 2: build a hierarchy tree (as shown in figure 1) in this way: The root node is the objective of the problem, the intermediate levels are the criteria, and the lowest level contains the alternatives. This hierarchical organisation is used to obtain a general overview of the criteria and their relations.

STEP 3: for each level, build a pairwise comparison matrix with the brothers (sons of the same node). The matrix contains the weights of pairwise comparisons between brother nodes. This provide us with a pairwise comparison matrix (like example of table 2) for each father nodes.

	Consumption	Price	Security	...
Consumption	1	1/6	2	
Price	6	1	3	
Security	1/2	1/3	1	
...				

Table 2. Example of comparison matrix for the first level to "Buy a car." The appraiser considers "Consumption" six times less important than "Price", twice as important as "Security", etc.

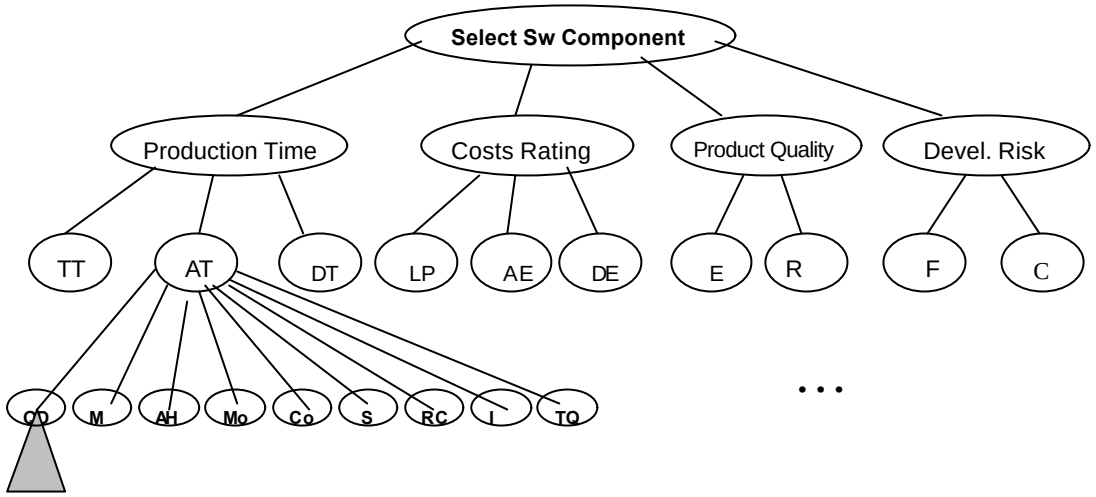

Figure 2. Representation of the decision criteria for the selection of a SC by means of the hierarchical structure of the AHP method. In this example, the criterion *Adaptation Time* (AT) has been expanded.

For each comparison matrix, an eigenvector must be calculated, using the equation: $|A - \lambda I| = 0$, where A is the comparison matrix, I is the identity matrix and λ is the *eigenvector*. This calculus must be performed for each level of the tree. The entire process can be studied in [11].

STEP 4: value each alternative (leaf nodes) with a fixed scale. The scales for rating characteristics should be established and described in a precise way.

STEP 5: determine the value of each criterion using a weighted addition formula, with the weights from step 3 and the values from the step 4. These results ascend up the tree to calculate the final value of the objective (root). This final value is used to make a decision about the objective.

4. BAREMO: APPLYING THE MODEL OF ANALYTIC HIERARCHIES IN THE CHOICE OF SOFTWARE COMPONENTS

The AHP model can be applied to decide whether or not to reuse SCs. This method is called BAREMO (BAlance REuse MOdel). In order to decide the reuse in a new software project, BAREMO can be used to: 1) select the most appropriate SC among various alternatives or, 2) decide the suitability of a particular SC for the project.

Taking into account the general steps of AHP, we have adapted the method to be used in the reuse of SCs:

STEP 1: specify the project objectives. The engineer should know the exact guidelines of their company and available resources in relation to the new software project.

STEP 2: build the decision tree (as shown in figure 2) from the concepts of table 1, so that the objective, "select the most appropriate SC for a new software project", is placed at the root node; the dimensions (production time, cost rating, final product quality and development risk) are placed at the first level; the factors of each dimension at the second level; and underneath these factors, the sub-trees of specific characteristics of the particular SC. The general characteristics of all types of SCs (shown in table 1) should be specialised according to: the particular SC, the specific target project, and the company that will develop the project.

STEP 3: for each set of brother nodes, make the pairwise

Terrible reputation	0	0	1.3	2.2
Low reputation	1.3	2.2	3.3	4.2
Not significant	3.3	4.2	5.4	6.5
Prestigious	5.4	6.5	7.7	8.6
Very Prestigious	7.7	8.6	10	10

Degrees of scale: “Prestige of the supplier”

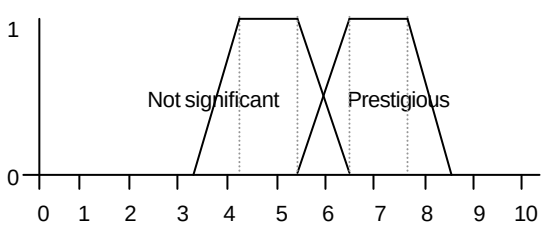

Figure 3. Representation of the degree: *Prestigious* and *Not_significant*

comparison matrices with the criteria of the decision tree. The *eigenvectors* are calculated from these matrices.

STEP 4: for each alternative SC, assess its characteristics. These values will (always multiplied by the weights calculated in the step 3) ascend up to the superior nodes of the tree, until the node root is calculated. For each one of these characteristics, the engineer should establish a scale of appropriate ratings.

STEP 4.1: this model assigns linguistic values (non-numbers) to the alternatives because the human beings, in their daily activities, usually make this type of judgment. For example, if an analyst evaluates the "prestige of the supplier", he/she can assess this quality using the linguistic scale: *very_prestigious*, *prestigious*, *not_significant* or *unknown*, *low_reputation* and *terrible_reputation*. It is better than a numeric scale between zero and ten. In this process, it is important that the groups of the linguistic values are precisely defined. Following the example, you could determine if the supplier of the component is "prestigious" in relation to: time in the market, obtained awards, other users' opinions, and others sources of information that the engineer considers important for the judgment.

However it is not possible to perform calculations with linguistic values. One possible representation of these linguistic values is diffuse intervals [4]. The diffuse intervals are determined by their angular points in a scale from 0 to 10, as shown in figure 3.

By assigning linguistic values with diffuse intervals let us perform basic mathematical operations for intervals. This way, it can be defined the operations of: sum of intervals (1) and product by a constant (2):

$$(a1,a2,a3,a4)+(b1,b2,b3,b4) = (a1+b1,a2+b2,a3+b3,a4+b4) \quad (1)$$

$$n * (a1,a2,a3,a4) = (n*a1, n*a2, n*a3, n*a4) \quad (2)$$

STEP 4.2: with these established linguistic scales for each one of the criterion, the engineer will proceed to study each one of the SCs that have been considered as alternatives, and to value them using these scales.

STEP 5: lastly, combine the vectors of weights *W* obtained in the step 3 with the values of the alternatives *V*, using the formula (e.g.): $\Sigma n\ wi\ vi$.

In large projects, which require a team of analysts, each person can provide their own values, and it will be necessary to reach an agreement. In this case, all the steps up to step 4.1 should reach a common consensus among the members of the team. Later, each analyst can value each one of the component candidates in an individual way. Finally, the suitable software component is chosen based on the results obtained.

5. A CASE STUDY OF THE APPLICATION OF BAREMO

To show an application example, we expose the steps given by a software project manager to reuse a particular SC. The chosen company has branches for software development in several cities of Spain and, logically, they usually develop and share SCs within their branches to increase their productivity. The company has personnel dedicated to test these products and they make descriptive documentation on the characteristics and operation of the component. On the other hand, they do not reuse software systematically, they do not maintain a repository with the developed components, nor do they pass rigorous controls of quality on the developed SC. Therefore, the company carries out an internal and opportunist reuse [3].

The real case that we examine is the reuse of a module with *Visual Basic 3.0* functions (its called *Scimg_6.BAS*). These functions can perform some simple operations on bitmaps images (zoom, rotations, contrasts, etc.) for *Windows*. The company considered to reuse this SC in a new project (by another development team in another branch) needed similar functions about images processing; although some other functions should be incorporated into the module. The project's client was a public organisation, and its aim is the digital storage of scanned certified documents. The project manager examined similar company projects and found a project of processing images that used *Scimg_6.BAS*. He agreed to use BAREMO to decide the suitability of this component for their new project.

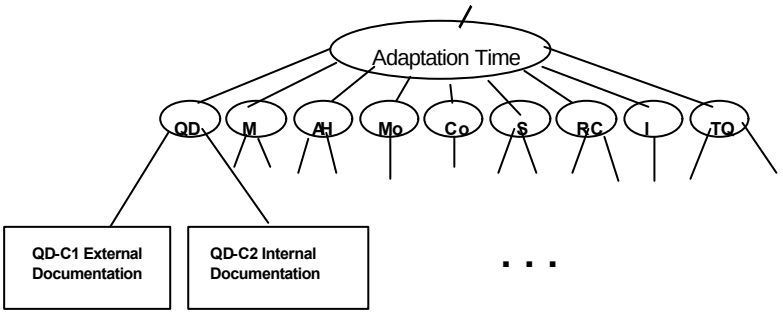

Figure 4. Specialisation of the sub-tree *Adaptation Time* for the example about reuse of the component *Scimg_6.BAS*. The specialised characteristics as example, belong together with the identified ones in table 4.

Scimg_6.BAS– V2 - *Adaptation Time*					
General Characteristic	**Code**	**Specific Characteristic**	**Weight (W)**	**Value (V)**	**Weighted Value (W x V)**
Quality of Documentation	AT-C1	External Documentation Quality	0.09	*Normal*	(0.306, 0.396, 0.504, 0.594)
	AT-C2	Internal Documentation Quality	0.06	*Bad*	(0.072, 0.132, 0.204, 0.264)
Methodology	AT-C3	Standarization of variables	0.05	*Normal*	(0.17, 0.22, 0.28, 0.33)
	AT-C4	Clarity of Reuse Contracts	0.01	*Very Bad*	(0, 0, 0.012, 0.022)
Adaptation Help	AT-C5	Are the original programmers available?	0.26	*Good*	(2.028, 2.288, 2.6, 2.6)
	AT-C6	Attitude of original programmers	0.19	*Excellent*	(1.482, 1.672, 1.9, 1.9)
Modularity	AT-C7	Does it need other external functions?	0.01	*Very Bad*	(0, 0, 0.012, 0.022)
Complexity	AT-C8	Arise average	0.02	*Bad*	(0.024, 0.044, 0.068, 0.088)
Size	AT-C9	Number of functions	0.01	*Bad*	(0.012, 0.022, 0.034, 0.044)
	AT-C10	Implementation with useful lines of code	0.02	*Good*	(0.112, 0.132, 0.156, 0.176)
Requirement Covered	AT-C11	Number of valid functions	0.12	*Good*	(0.672, 0.792, 0.936, 1.056)
	AT-C12	Appropriate image resolution	0.05	*Excellent*	(0.39, 0.44, 0.5, 0.5)
Tools Quality	AT-C13	Test data	0.02	*Very Bad*	(0, 0, 0.024, 0.044)
Interoperativity	AT-C14	Is the programming language known?	0.08	*Excellent*	(0.624, 0.704, 0.8, 0.8)
	AT-C15	Changes with the new programming language version	0.01	*Good*	(0.056, 0.066, 0.078, 0.088)
RESULT			**1**		**(5.948, 6.908, 8.108, 8.528)**

Table 3. Table of valuation of the impact of specialised characteristics in the *Adaptation Time.* The value (5. 948, 6. 908, 8. 108, 8. 528) will influence the *Production Time* dimension.

STEP 1: *Specify the project objectives*. The objectives of the project were, in general, to develop the computer program with the smallest possible cost, at one time reasonably large, and its guarantee against failure.

STEP 2: *Build the decision tree*. Firstly, the engineer must specialise the characteristics for the studied component type. The general characteristics shown in figure 2 must be defined ad hoc for this *Visual Basic* module. Figure 4 is an example of the specialisation of the *Adaptation Time* sub-tree. We can appreciate that the manager director has specialised the general characteristic *Quality of Documentation* (QD) in two specific characteristics of this component type: *Quality of the External Documentation* (legible and detailed information appeared in the printed manuals), and *Quality of the Internal Documentation* (quality of the internal explanations of functions and data). The engineer should specialise each general characteristics of the figure 2. The entire specialisation of the *Adaptation Time* can be seen in table 3.

STEP 3 *Build the pairwise comparison matrices for each set of brother nodes*. The comparison used to calculate the weights of the dimensions (first level) is shown in table 4; row 6 shows the weights calculated for this issue.

STEP 4.1: *Establish the linguistic scales of values for each one of the characteristics*. In the example followed by the *Adaptation Time*, the engineer determined a scale (exposed in table 6), with five degrees of valuation that could be employed to qualify all the influential characteristics in this factor.

Scimg_6.BAS M1	Production Time	Costs Rating	Product Quality	Risk
Production Time	**1**	**1/6**	**1/4**	**1/8**
Costs Rating	**6**	**1**	**2**	**1/2**
Product Quality	**4**	**1/2**	**1**	**3**
Risk	**8**	**2**	**1/3**	**1**
RELATIVE WEIGHTS	**0.31**	**0.17**	**0.28**	**0.24**

Table 4. Comparison matrix of the criterion relation to the dimensions. Last line shows the weights calculated from this matrix.

STEP 4.2: *Valuation of the characteristics*. It was studied the particular characteristics of the component (*Scimg_6.BAS*) to obtain the nodes father's values. Table 3, in the column *Value*, contains the appreciation of the characteristics of this module. The project manager has used the scales proposed in the step 4.1. (Notice that the specified scales of degrees and the specialised

TT – Scimg_6.BAS	Value	Weight	Weighted Value
Production Time	(6.1222, 7.1223, 7.7559, 8.1912)	0.31	(1.897882, 2.207913, 2.404329, 2.539272)
Costs Rating	(3.7232, 4.6332, 7.3344, 7.8312)	0.17	(0.632944, 0.787644, 1.246848, 1.331304)
Product Quality	(7.2328, 8.8631, 9.0245, 9.7153)	0.28	(2.025184, 2.481668, 2.52686, 2.720284)
Development Risk	(5.4542, 6.8327, 8.1224, 8.9332)	0.24	(1.309008, 1.639848, 1.949376, 2.143968)
RESULT		1	**(5.865018, 7.117073, 8.127413, 8.734828)**

Table 5. Calculation of the suitability value of the component *Scimg_6. BAS*.

characteristics are such that they are established by the manager of this project; another engineer might identify another one).

Linguistic scale	Diffuse Interval			
Very Bad	0	0	1.2	2.2
Bad	1.2	2.2	3.4	4.4
Normal	3.4	4.4	5.6	6.6
Good	5.6	6.6	7.8	8.8
Excellent	7.8	8.8	10	10

Table 6. Linguistic scale used in the valuation of characteristics in *Adaptation Time* factor.

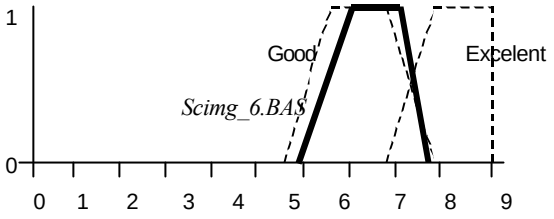

Figure 5. Representation of the suitability value of Scimg_6. BAS in front of the Good and Excellent degrees of valuation.

STEP 5 *Calculation of the nodes father's values*. As appears in table 3, the weights of each criterion are multiplied by the

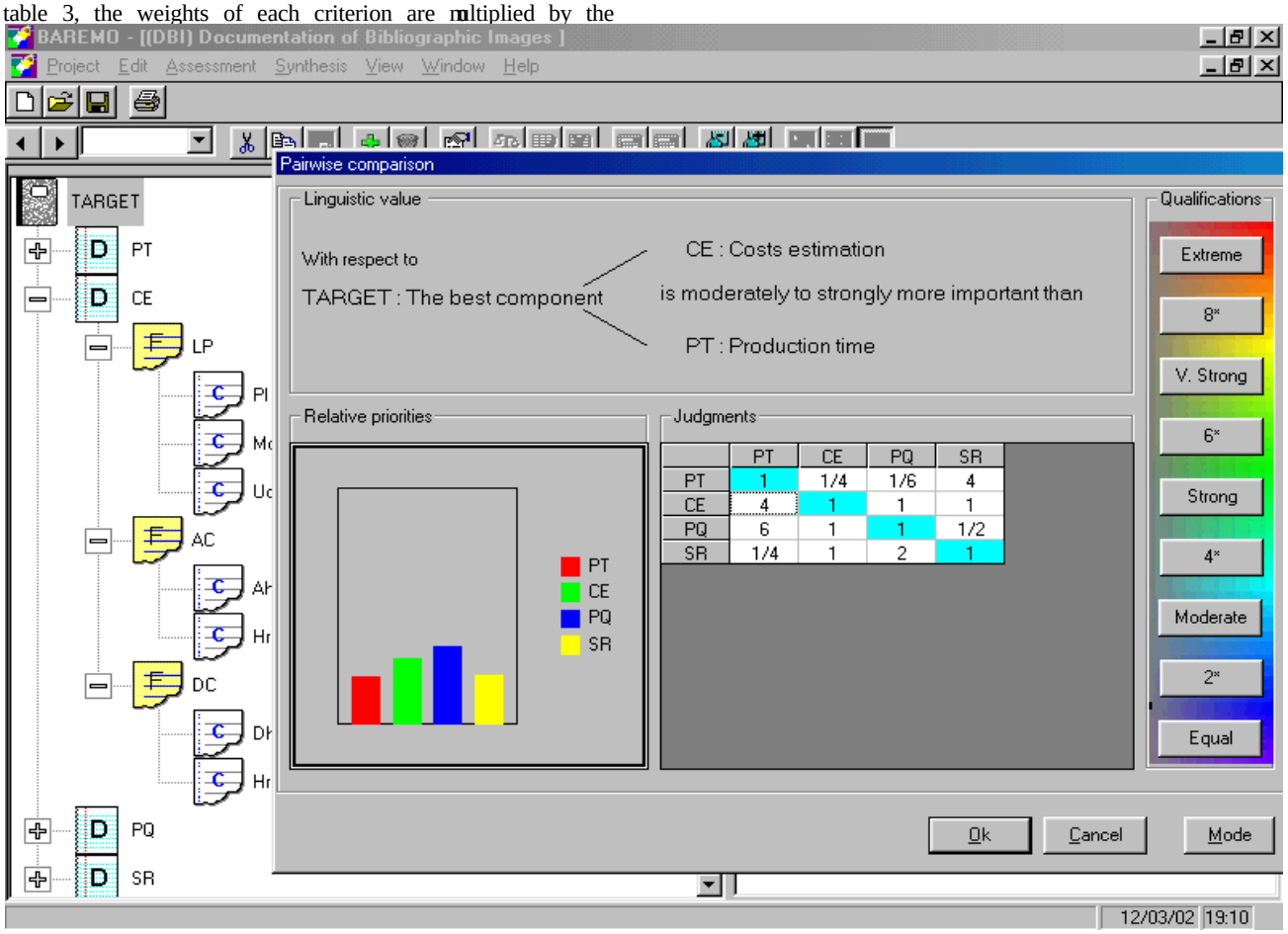

Figure 6. Example of *BAREMO Tool* with the followed study case.

linguistic valuation (with the corresponding numeric relationship indicated in table 6), being obtained the diffuse intervals of the column *Weighted Value*.

The result is calculated by means of the combination of weighted addition ($\Sigma n\ wi\ vi$). This value multiplied by their weight (that was calculated in the corresponding comparison matrix), enable to find the father value *Time of production* (i.e., it will be calculated the weighted addition with the brothers nodes: *Training Time* and *Development Time*). The value obtained for the node *Time of production* (together with their brother nodes) helps us to obtain the final valuation of the objective of the suitability of the component *Scimg_6.BAS*. The results are shown in table 5.

The obtained suitability value was: (5.865018, 7.117073, 8.127413, 8.734828) that, relating it with the linguistic scale chosen by the engineer (table 6), indicates that it is close to *Good* and to *Excellent* (as shown in figure 5). With this result, the engineer decided to reuse the component *Scimg_6.BAS* in the project that it is completing the established development plans. Also, it allowed the project manager to obtain a detailed report for the directive of the company about the taken decision.

Although in this case only the suitability of a component was valued, for other candidates, steps 4.2 and 5 should be performed repeatedly, since the comparison frame (designed in the previous steps) has been established. The complete process to obtain the value of suitability of the component *Scimg_6.BAS* took the engineer about twenty minutes, using the *BAREMO Tool*. The figure 6 shows this tool with the followed case study.

6. CONCLUSIONS AND FUTURE WORKS

BAREMO is an application of the AHP model to help software engineers choose the appropriate component for a new project; in order to do this, the engineer must compare the importance of the objectives, and study carefully the SC characteristics. Although the specialisation of the characteristics and the assessment of the criteria of a particular SC require a considerable effort, the above framework provides a useful schema to carry out complex multicriteria decision-making.

Feedback from project managers who have used the method, reveals that specifying the characteristics of a certain SC is complicated and expensive, and its assessment is quite subjective; however, they state that, once the framework has been defined and if it is applied to one particular type of SC, BAREMO helps to justify decisions taken, to "clarify ideas", and to weigh up the advantages and the risks involved in choosing a component from another. There is a software tool, *BAREMO Tool* [1], which is used to apply the method.

Future works consist of adapting the method to different kinds of SCs, and establishing formal metrics to assess their suitability on software projects in different domains.

7. ACKNOWLEDGEMENTS

We want to thank Mr. Francisco Javier Prieto, production manager of the company MECASER S.A. and Mr. Enrique Pons, systems manager of the ASOCIACIÓN PROMI CÓRDOBA, for their opportune ideas in the conception of BAREMO, and for their invaluable effort in the tests they have slaved over.

8. REFERENCES

[1] Basili, V.; Briand L.; and Thomas W.: 'Domain Analysis for the Reuse of Software Development Experiences', Proceedings 19th Annual Soft. Eng. Workshop, NASA/GSFC, Greenbelt, MD, Dec. 1994

[2] Boehm, B.: 'Software Engineering Economics' Prentice-Hall, Englewood Cliffs. NJ., 1981.

[3] Frakes, W.; and Terry C.: 'Software Reuse: Metrics and Models' ACM Computing Surveys, vol.28, no.2, pp.415-435, Jun. 1996.

[4] Gomez-Perez, A.; Juristo, N.; Montes, C.; and Pazos, J.: 'Knowledge Engineering', Ed. Centro de Estudios Ramón Areces, 1997.

[5] Gomez-Perez, A.: 'Knowledge Sharing and Reuse', The Handbook of Applied Expert Systems', Ed. J. Liebowitz, CRC Press, 1998.

[6] Gomez-Perez, A, and Lozano, A.: 'Impact of Software Components Characteristics above Decision-making Factors', Workshop on Component-based Software Engineering, pp, 15-23, 22nd International Conference on Software Engineering (ICSE'00) Limerick, Ireland, Jun 2000.

[7] Jones, C.: 'Software Return on Investment Preliminary Analysis'. Soft. Productivity Research, Inc, 1993.

[8] Pressman, R.S.: 'Software Engineering: A Practitioner's Approach'. Ed. McGraw-Hill, Inc. 1997.

[9] Prieto-Diaz, R.; and Freeman, P.: 'Classifying Software for Reusability', IEEE Soft, vol.4, n.1, pp.6-16. Jan. 1987.

[10] Saaty, T.: 'A Scaling Method for Priorities in Hierarchical Structures'. Journal of Mathematical Psychology, vol.15, pp 234-281, 1977.

[11] Saaty, T.: 'How to Make a Decision: The Analytic Hierarchy Process'. European Journal of Operational Research, vol.48, pp 9-26, 1990.

[12] Taylor, D.: 'The Use and Abuse of Reuse', Object Magazine, vol. 6, n.2, pp.16-18, Apr. 1996.

[1] http://webepcc.unex.es/~alozano/baremo/