

# WEIGHTED SALIENCE ALLOCATION TO GOALS IN THE UNC-METHOD

## ASIGNACIÓN DE PORCENTAJES DE IMPORTANCIA A LOS OBJETIVOS EN UNC-METHOD

CARLOS MARIO ZAPATA

*Profesor Asociado, Facultad de Minas, Universidad Nacional de Colombia, Sede Medellín, cmzapata@unal.edu.co*

JUAN FERNANDO ACEVEDO

*Investigador, Facultad de Minas, Universidad Nacional de Colombia, Sede Medellín, jacevedoo@unal.edu.co*

DAVID MORENO NIÑO

*Investigador, Facultad de Minas, Universidad Nacional de Colombia, Sede Medellín, damorenon@unal.edu.co*

Received for review March 1<sup>st</sup>, 2012, Accepted May 26<sup>th</sup>, 2013, final version June, 15<sup>th</sup>, 2013

**ABSTRACT:** The KAOS (Knowledge Acquisition autOated Specification) goal diagram can be used to hierarchically represent and analyze organizational goals. In the context of the UNC-Method, such goals are often written in the aforementioned diagram by using a set of verbs denoting achievement, in order to represent the organizational goals and requirements. Nowadays, the importance of the goals is only established by the diagram hierarchy. However, requirements engineering considers other elements as well as the hierarchy in such a matter. On the other hand, there are proposals that assign importance to the goals on other diagrams, but they neglect the hierarchy and the type of verb that is used in wording. These reasons lead us to propose in this paper, a method for allocating weighted salience to KAOS diagram goals, in the context of the UNC-Method. We consider the verb type and the hierarchy. We explain this proposal by means of a lab study.

**KEYWORDS:** Goals, KAOS diagram, achievement verbs, requirements engineering

**RESUMEN:** El diagrama de objetivos de KAOS (especificación automatizada para la adquisición del conocimiento, por sus siglas en inglés) se puede utilizar para representar y analizar jerárquicamente las metas de una organización. En el contexto de UNC-Method, dichas metas se suelen redactar en el mencionado diagrama empleando un conjunto de verbos que denotan logro, para representar los objetivos y requisitos de la organización. Actualmente, la importancia de los objetivos sólo la establece la jerarquía del diagrama, lo cual no es suficiente, puesto que la ingeniería de requisitos reconoce otros elementos que influyen en la importancia de los objetivos. Por otro lado, existen propuestas que asignan la importancia a los objetivos sobre otros diagramas, pero dejan de lado la jerarquía y el tipo de verbo que se emplea en su redacción. Por las razones anotadas, en este artículo se propone un método para asignar porcentajes de importancia a los objetivos del diagrama de KAOS, en el contexto de UNC-Method, tomando en consideración la clase de verbo y la distribución jerárquica que poseen. Esta propuesta se ejemplifica con un estudio de laboratorio.

**PALABRAS CLAVE:** Objetivos, diagrama de KAOS, verbos de logro, ingeniería de requisitos

### 1. INTRODUCTION

Organizational goals can be hierarchically represented by using the KAOS (Knowledge Acquisition autOated Specification) goal diagram. In this diagram, high level goals—the most important ones—are decomposed into others until the lowest level of importance (where expectations and requirements are located) is achieved. Goal decomposition is a way to explain the operationalization of goals by using requirements and expectations [1].

In the context of the software development method called the UNC-Method [2], goal and requirement

writing commonly starts with a special kind of verb denoting achievement. Such verbs can belong to one of three categories: maintenance, improvement, and achievement [3]. Also, the hierarchical position of the elements (goals, requirements, and expectations) reveals their importance. So, elements located at the same hierarchical level have the same importance. However, the hierarchical level of a goal is not convenient for calculating the importance of the goal. In fact, according to Liaskos et al. [4], “the traditional notion of requirements prioritization originates exactly from the observation that not all requirements have the same importance for all stakeholders.”

Also, some of the state-of-the-art methods include both quantitative and qualitative procedures for determining the salience of a goal, among others [4-7]. Notwithstanding, such proposals work on *i\** (a non-hierarchical diagram) and they avoid the usage of achievement verbs, leading to goal-activity misunderstanding [8]. In some cases [4-7], the analyst and the stakeholder subjectively assign the priorities to the goals instead of using the hierarchical features of the diagram and the linguistic information related to the kind of verb. Such information is important to establish whether a goal is located either in a high level of importance (more “abstract”) or a low level of importance (more “operational”).

The aforementioned reasons lead us to propose, in this paper, a method for allocating salience weights to requirements and goals belonging to the KAOS goal diagram, in the context of the UNC-Method. We use the category of verb, and some structural criteria, for determining the weight salience of the elements. Such criteria provide an objective way to assign priorities to the goals.

The remainder of this paper is organized as follows: in Section 2 we present the theoretical framework for the conceptual basis of this work; in Section 3 we summarize the state-of-the-art review about determining salience among goals; in Section 4 we define the needed criteria for assigning salience to goals in the context of the UNC-Method and we exemplify the proposal by using a lab study. Conclusions and future work are presented in Section 5.

## 2. THEORETICAL FRAMEWORK

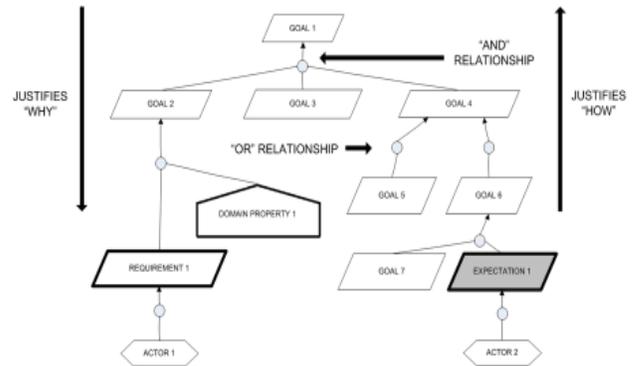
### 2.1. KAOS goal diagram

The goal diagram is one of the artifacts belonging to the KAOS methodology. In this diagram, high level goals are defined and then hierarchically decomposed into more specific goals, until we reach the level of requirements, expectations, or domain properties [1]. Goals in certain levels justify “why” the lower level goals are defined and “how” the upper level goals are achieved. The elements of the KAOS goal diagram (see Fig. 1) are described as follows [1]:

- Goal: what is intended to be achieved.
- Requirement: non-negotiable goal of a software application.
- Actor: person in charge of supporting either a

requirement or expectation.

- Domain property: feature needed to achieve goals at a certain level.
- Connectors: Arrows for linking goal diagram elements. If the elements linked by a connector are simultaneously needed, the connector



**Figure 1.** KAOS goal diagram in the context of the UNC-Method (translated from [8])

### 2.2. Achievement verbs

Zapata and Lezcano [8] and Zapata *et al.* [9] propose an achievement verb taxonomy to be used in writing organizational goals. So, Zapata and Lezcano [8] increase the verb list proposed by Anton [3], initially grouped into maintenance, improvement, and achievement verbs (see Table 1). According to Anton [3], this taxonomy provides an indirect measure of the importance of the verbs, since achievement verbs can be located closer to the operational level, but maintenance and improvement verbs are considered more strategic in nature. The UNC-Method [10] uses the three categories of verbs for clarifying the information about the goals expressed by the stakeholder. In fact, the stakeholders tend to use operational verbs for expressing goals (*e.g.*, “selling more cars”), but the analyst—working with the UNC-Method—can suggest a situation closer to the real expression of the goal (*e.g.*, “increasing the car sales”). In this case, the operationalization of the goals is postponed until we can define the adequate requirement related to the “goal” sentence provided by the stakeholder.

### 2.3. Goal analysis in the context of the UNC-Method

The KAOS goal diagram is used for representing organizational goals, in order to determine the impact of a software application on the organization itself. In the context of the UNC-Method [10], such a diagram

is divided into levels, by numbering the lowest level, one, and increasing by one for every higher level. The level is used for calculating the impact of one node (goal, requirement, or expectation) on the software application. By doing this, we are recognizing equal salience to nodes located at the same level, and linear salience progression among levels. Even though the hierarchy is implicit, in this way of assigning salience, the distinction among goals and requirements seems to be unimportant. Also, the calculation provides a higher value in the sum of goals which are children of a goal. For example, in the Figure 1 the goal 1 is from level 4 while goals 2, 3, and 4 are from level 3. The sum of such goals is 9, higher than 4, the “importance” of goal 1. For this reason, we require a new salience analysis for the goals in the context of the UNC-Method.

**Table 1.** Maintenance, improvement, and achievement verbs (taken from [8])

<i>Maintenance</i>		
Administrare	Manage	Concede
Endorse	Save	Achieve
Keep	Maintain	Get
Give	Preserve	Offer
Guarantee	Grant	
<i>Improvement</i>		
Develop	Diminish	Increase
Foster	Reduce	Enlarge
Improve	Assess	Decrease
Augment	Broaden	Expand
<i>Achievement</i>		
Cause	Dispose	Form
Make	Produce	Formulate
Prepare	Promote	Advance
Execute	Do	Create
Accomplish	Compose	Act

### 3. BACKGROUND

According to Liaskos *et al.* [4], establishing priorities (salience percentages) to requirements is based on the disparity of stakeholder perception among goals. For this reason, they propose a priority system on  $i^*$  requirements models to determine the degree of stakeholder satisfaction for each requirement. This task is achieved by using weighted formulas to be compared, and finally, to compute the requirement preference ranking.

Giorgini *et al.* [5] and Sebastiani *et al.* [6] propose the analysis of positive-negative impact of a leaf goal (inside an  $i^*$  diagram) over its parent and vice-versa. Also, by using propagation techniques according to an

axiom set, they calculate the contribution of every goal to the entire impact on the diagram.

Karlson and Ryan [7] set goal salience based on goal value and relative cost, by applying the Analytic Hierarchy Process (AHP) [11]. The results of this process are discussed with the stakeholder, and displayed in a cost-value diagram. Finally, the project leader assigns the salience percentage to each requirement and decides on the implementation of the most important ones.

Even though the aforementioned authors recognize that each goal has a relative salience for the stakeholder, they exhibit some drawbacks:

- Most of the work is related to the  $i^*$  diagram, which is non-hierarchical. As we previously established, hierarchy is important to understand the operationalization of goals. When we work with  $i^*$  diagrams, such relationships are missing and goal salience is consequently affected.
- None of the authors considered the kind of verb used in writing goals, because they subjectively assign the importance percentage, based on the perception of the analyst and the stakeholder. For this reason, they avoid the discussion about the abstraction degree of the goals (if they are closer to the operative level [8]). So, a more abstract goal (*e.g.*, “increase”) seems to be equally important than a more concrete goal (*e.g.*, “formulate”) even though the second one is closer to the operative level than the first one.

### 4. A METHOD FOR ASSIGNING THE SALIENCE PERCENTAGE TO GOALS

Our aim with this method is to determine the goal or requirement salience percentage, according to the hierarchical distribution of the KAOS goal diagram, the verbs used in writing the goals, and the kind of node (goal or requirement) to be analyzed. We are trying to improve the objectivity in allocating the salience percentage to goals (and consequently to problems, since problems are unsatisfied goals) associated with an organization (see Zapata *et al.* [10]). At the end of this Section, we present a lab study for exemplifying the usage of the criteria.

### 4.1. Criteria for assigning the salience percentage

These criteria are based on the assertion, from the KAOS theory, that one goal/requirement is considered “satisfied” when the set of goals/requirements that it depends on it are satisfied [2]. Note that the KAOS goal diagram is the result of hierarchically decomposing a goal with father-children, AND/OR relationships. For this reason, we propose salience percentage of a goal as the distribution of the percentage of a father goal into each of its children by using certain criteria. The root of the tree (the most important goal) has a 100% salience. Such a distribution poses a major change in the way we interpret the importance of the goals in the context of the UNC-Method, because in this case the sum of the percentages allocated to the set of child goals, is equal to the percentage allocated to the correspondent father goal. As we mentioned before, currently this sum does not have the same value as the father goal value.

*Criterion 1: Allocating salience percentage to the children in AND-type connections*

A goal/requirement gives equal percentages to any child, with no distinction of either goal-type or requirement-type of the nodes. Graphically, we can see an example of this criterion in Table 2.

**Table 2.** Example of the criterion 1

KAOS goal diagram	Distribution
	-Father: 100% -Connectors: 2 -Connector percentage: (Father/Connectors) = 100/2 = 50% -Elements in connector: 2 -(Connector percentage / elements in connector): 50/2 = 25%

*Criterion 2: Weighting the percentage to be assigned by considering the category of verb.*

As suggested by Hughes [13], when we estimate values in the context of software engineering methods, we can use expert judgment. In the absence of information about the salience related to the verb category, we can use some expert judgment to assign a percentage to the goals/requirements. As we mentioned before, improvement and maintenance are closer to the higher organizational level while achievement is closer to the

operational level. In such a case, when the child uses either a maintenance-type or improvement-type verb, we decide to assign double the weight to its percentage than with achievement-type verbs. In Table 3 we can see an example of this rule.

**Table 3.** Example of the criterion 2

KAOS goal diagram	Distribution
	-Father: 100% -Weight sum: maintenance (2) + maintenance (2) + achievement (1) + improvement (2) = 7 -Element: (Father/Weight sum)*Element weight = (100/7) * Element weight

*Criterion 3: Weighting the percentage to be assigned by considering the type of elements.*

Expert judgment is again the device for assigning percentages to the goals/requirements. The rationale behind this assumption is based on the fact that the goals are always in the root of the goal diagram, while the requirements are usually leaves. So, when the child is a goal-type element, we decide to assign double the weight to the percentage than for requirement-type elements. Again, in Table 4 we can see an example of this rule.

Finally, we calculate the average of the percentages obtained by applying the three criteria, and the result is considered the salience percentage to be allocated to each goal/requirement belonging to the KAOS goal diagram. Be advised that the values obtained by using only the level are completely different to those obtained by using this method.

**Table 4.** Example of the criterion 3

KAOS goal diagram	Distribution
	-Father: 100% -Weight sum: goal (2) + requirement (1) + requirement (1) + goal (2) = 6 -Element: (Father/weight sum)*Element weight = (100/6) * Element weight

### 4.2. Lab study

In this Section we propose an example in order to apply the criteria defined in the previous Section. The research question about this problem is the following: using the previously defined criteria, can we assign the percentages to the goals/requirements of the goal diagram?

The starting point is the goal diagram of the Fig. 3, taken from Zapata [12]. In this example, some goals belonging to a software development company are included. The core activity of this company is the requirements elicitation process and their transformation into the so called UN-Lencep discourses (UN-Lencep is a controlled language for defining requirements).

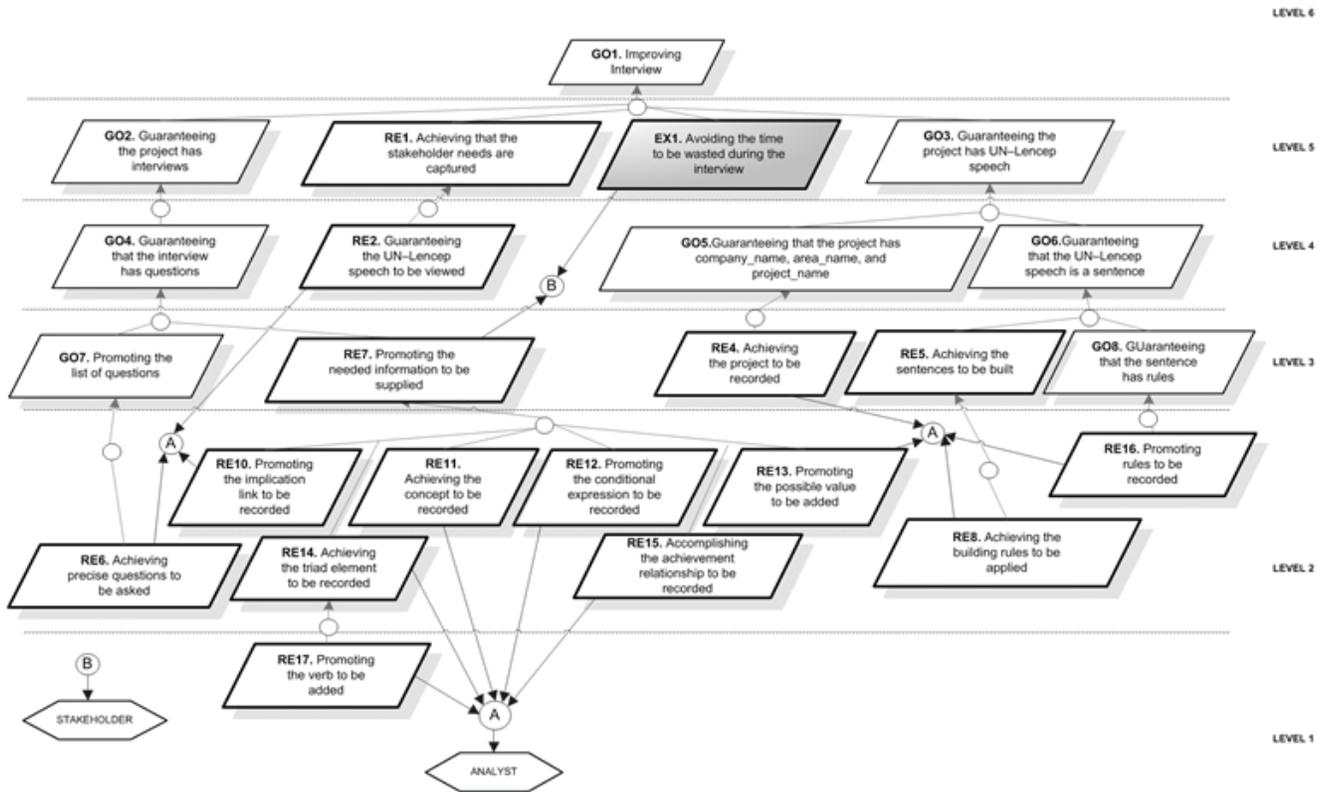


Figure 2. Goal diagram for explaining the lab study (taken from [12])

Note that the diagram in Fig. 2 has the level-based structure defined by the UNC-Method.

The remarkable features of the diagram are the following:

- Every node is encoded to ease the traceability process (all of the goals are encoded with “GO”+number, all of the requirements with “RE”+number, and the only expectation is encoded “EX1”).
- All of the nodes belonging to the KAOS goal diagram start with a verb which can be classified into one of the three categories defined by Antón [3].

- Each node can be graphically classified either as a goal, a requirement, or an expectation. The last one is not considered for the analysis, since we assume it will not be present in the software application.
- References A and B are only used for linking some nodes to the respective actor.

Table 5 summarizes the results of applying the criteria to the Fig. 2. In such a Figure, the columns have the following values: a) *father* is the immediate higher level node of the current node, b) *KAOS element* is the node itself, c) *level* and % are the values linked to the criterion 1, d) *verb weight* and *verb %* are the values linked to the criterion 2, e) *element weight* and

*element %* are the values linked to the criterion 3, f) *weighted %* is obtained from the average of *level %*, *verb %*, and *element %*, g) *leaf %* is the *weighted %* belonging to the leaves of the KAOS goal diagram.

As a way to explain the calculations made to obtain Table 5, next we detail the process for applying the

criteria to the elements belonging to level 5.

- Criterion 1: (Allocating salience percentage to the children in AND-type connections) goal GO1 has three children (GO2, RE1, and GO3) and 100% to allocate. Consequently, each child inherits 33.33%.

**Table 5.** Results of the weighted salience allocation by applying the criteria

Father	KAOS element	Level	%	Verb weight	Verb %	Element Weight	Element %	Weighted %	Leaf %
ROOT	GO1 Improving interview	6	100	2	100	2	100	100	0
			<b>100</b>	<b>2</b>	<b>100</b>	<b>2</b>	<b>100</b>	<b>100</b>	
GO1	GO2. Guaranteeing the project has interviews	5	33.33	2	33.33	2	40	35.56	0
GO1	RE1. Achieving that the stakeholder needs are captured	5	33.33	2	33.33	1	20	28.89	0
GO1	GO3. Guaranteeing the project has UN-Lencep speech	5	33.33	2	33.33	2	40	35.56	0
			<b>100</b>	<b>6</b>	<b>100</b>	<b>5</b>	<b>100</b>	<b>100</b>	
RE1	RE2. Guaranteeing the UN-Lencep speech to be viewed	4	33.33	2	33.33	1	20	28.89	28.89
			33.33	2	33.33	1	20	28.89	
GO2	GO4. Guaranteeing that the interview has questions	4	33.33	2	33.33	2	40	35.56	0
			<b>33.33</b>	<b>2</b>	<b>33.33</b>	<b>2</b>	<b>40</b>	<b>35.56</b>	
GO3	GO5. Guaranteeing that the project has company_name, area_name, and project_name	4	16.67	2	16.67	2	20	17.78	0
GO3	GO6. Guaranteeing that the UN-Lencep speech is a sentence	4	16.67	2	16.67	2	20	17.78	0
			<b>33.33</b>	<b>4</b>	<b>33.33</b>	<b>4</b>	<b>40</b>	<b>35.556</b>	
GO4	GO7. Promoting the list of questions	3	16.67	1	11.11	2	26.667	18.15	0
GO4	RE7. Promoting the needed information to be supplied	3	16.67	2	22.22	1	13.333	17.41	0
			<b>33.33</b>	<b>3</b>	<b>33.33</b>	<b>3</b>	<b>40</b>	<b>35.556</b>	
GO5	RE4. Achieving the project to be recorded	3	16.67	2	16.67	1	20	17.78	17.78
			<b>16.67</b>	<b>2</b>	<b>16.67</b>	<b>1</b>	<b>20</b>	<b>17.778</b>	
GO6	RE5. Achieving the sentences to be built	3	8.33	2	8.33	1	6.67	7.778	0
GO6	GO8. Guaranteeing that the sentence has rules	3	8.33	2	8.33	2	13.33	10	0
			<b>16.67</b>	<b>4</b>	<b>16.67</b>	<b>3</b>	<b>20</b>	<b>17.778</b>	
RE7	RE10. Promoting the implication link to be recorded	2	2.78	2	4.04	1	2.22	3.01	3.01
RE7	RE11. Achieving the concept to be recorded	2	2.78	1	2.02	1	2.22	2.34	2.34
RE7	RE12. Promoting the conditional expression to be recorded	2	2.78	2	4.04	1	2.22	3.01	3.01
RE7	RE13. Promoting the possible value to be added	2	2.78	2	4.04	1	2.22	3.01	3.01
RE7	RE14. Achieving the triad element to be recorded	2	2.78	2	4.04	1	2.22	3.01	0.0
RE7	RE15. Accomplishing the achievement relationship to be recorded	2	2.78	2	4.04	1	2.22	3.01	3.01
			<b>16.67</b>	<b>11</b>	<b>22.22</b>	<b>6</b>	<b>13.33</b>	<b>17.41</b>	
GO7	RE6. Achieving precise questions to be asked	2	16.67	2	11.11	1	26.67	18.15	18.15
			<b>16.67</b>	<b>2</b>	<b>11.11</b>	<b>1</b>	<b>26.67</b>	18.15	
GO8	RE16. Promoting rules to be recorded	2	8.33	2	8.33	1	13.33	10	10.00
			<b>8.33</b>	<b>2</b>	<b>8.33</b>	<b>1</b>	<b>13.33</b>	<b>10</b>	
RE5	RE8. Achieving the sentences to be built	2	8.33	2	8.33	1	6.67	7.78	7.78
			<b>8.33</b>	<b>2</b>	<b>8.33</b>	<b>1</b>	<b>6.67</b>	<b>7.78</b>	
RE14	RE17. Promoting the verb to be added	1	2.78	2	4.04	1	2.22	3.01	3.01
			<b>2.78</b>	<b>2</b>	<b>4.04</b>	<b>1</b>	<b>2.22</b>	<b>3.01</b>	<b>100</b>

- Criterion 2: (Weighting the percentage to be assigned by considering the category of verb) each child is headed by a maintenance verb (guaranteeing and achieving) with a relative weight of 2. For this reason, again, each child inherits 33.33%.
- Criterion 3: (Weighting the percentage to be assigned by considering the type of elements) goal GO1 has two goal-type children and one requirement-type child. Goals have a relative weight of 2 while the only one requirement has a relative weight of 1. Consequently 100% of goal GO1 should be divided by 5 ( $2*2 + 1*1$ ) resulting in 20%. Finally, each goal obtains  $20%*2=40%$  and the requirement obtains 20%.

The detailed weighted salience allocation is the average of the results of applying the criteria, as shown in Table 6.

**Table 6.** Results of the weighted salience allocation for the level-5 elements

KAOS Element	Level %	Verb %	Element %	Average %
GO2. Guaranteeing the project has interviews	33.33	33.33	40	35.56
RE1. Achieving the stakeholder needs are captured	33.33	33.33	20	28.89
GO3. Guaranteeing the project has UN-Lencep speech	33.33	33.33	40	35.56
	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>

Also, the final results are consistently allocated, since the sum of the percentages allocated to leaf elements is 100%, the value of the root GO1. (See Table 7).

**Table 7.** Weighted salience allocation for the leaf-type elements

Father	KAOS element	Weighted %
RE1	RE2. Guaranteeing the UN-Lencep speech to be viewed	28.89
GO5	RE4. Achieving the project to be recorded	17.778
RE7	RE10. Promoting the implication link to be recorded	3.01
RE7	RE11. Achieving the concept to be recorded	2.34
RE7	RE12. Promoting the conditional expression to be recorded	3.01
RE7	RE13. Promoting the possible value to be added	3.01
RE7	RE15. Accomplishing the achievement relationship to be recorded	3.01
GO7	RE6. Achieving precise questions to be asked	18.15
GO8	R16. Promoting rules to be recorded	10.00
RE5	RE5. Achieving the sentences to be built	<b>7.78</b>
RE14	RE17. Promoting the verb to be added	3.01
TOTAL		<b>100</b>

The resulting percentage allocation of the lab study was reviewed by two experts in the requirements elicitation field. They compared the current results with the previous allocation proposed by the UNC-Method and they believe the current results are more realistic, since every node (goal/requirement) gives its percentage to its children—in Section 2.3 we discussed the previous approach.

Also, the criteria 2 and 3 were discussed with them in order to validate it, but only one of the experts was in agreement with the assignation value, while the other declared he was neutral about this situation.

## 5. CONCLUSIONS

In this paper we proposed a method for weighted salience allocation to the KAOS goal diagram elements in the context of the UNC-Method. The proposed method considers the level of the element (by recognizing the hierarchical nature of the goal diagram), the category of verb employed by the element (maintenance, improvement, or achievement), and the type of element (goal or requirement).

The proposed method replaced the previous way of weighting goals and requirements, based only on the hierarchy level (as discussed in Section 2.3). Even though the previous method made an effort to assign a weight to the KAOS diagram elements, the allocated weight was linear in distribution and uneven in the distributed values. For example, with the previous method the value 6 would be allocated to GO1 while 5 would be allocated to GO2. By using the proposed method as explained in the lab study, 100% was allocated to GO1, while 35.56% was allocated to GO2. Also, in the previous method the sum of the weighted percentages allocated to leaf elements would be completely different to the weight of the root element. Consequently, the proposed method can be considered closer to the stakeholder way of thinking than the previous one, since stakeholders perceive more realistic and proportional values of the goals/requirements.

Some work has still to be done related to this line of research:

- Proposing new ways to extend the method in order to consider expectation-type elements of the KAOS

goal diagram. When omitting the expectations in the analysis, we are introducing some mistakes in the final calculations.

- Integrating the main ideas of this proposal with other artifacts belonging to organizational analysis, as a way to improve the quantitative results of such artifacts.
- Exploring the possibility of using a Delphi survey to reach agreement on the values to be assigned by the criteria 2 and 3. The experts to be consulted should have experience in both requirements engineering and computational linguistics.
- Developing a software application with the required criteria for performing the salience allocation.

## REFERENCES

- [1] Dardenne, A., Lamsweerde, A. and Fickas, S., Goal Directed Requirements Acquisition. *Science of Computer Programming*, Vol. 20, No. 1–2, pp. 3–50, 1993.
- [2] Zapata, C. and Arango, F., The UNC-Method: a Problem-Based Software Development Method. *Ingeniería e Investigación*, Vol.29, No.1, pp. 69–75, 2009.
- [3] Anton, A., Goal-Based Requirements Analysis. *Proceedings of the 2nd IEEE International Conference on Requirements Engineering*, Colorado Springs (USA). pp. 136–144, 1996.
- [4] Liakos, S., Mcilraith, S. and Mylopoulos, J., Towards Augmenting Requirements Models with Preferences. *International Conference on Automated Software Engineering 24th IEEE/ACM*, Auckland (New Zealand). pp. 565–569, 2009.
- [5] Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R., Reasoning with Goal Models. *Proceedings of the 21st International Conference on Conceptual Modeling*, London (UK). pp. 167–181, 2002.
- [6] Sebastiani, R., Giorgini, P., Mylopoulos, John. Simple and Minimum-Cost Satisfiability for Goal Models. *International Conference on Advanced Information Systems Engineering*, Berlin (Germany), vol. 3084. pp. 20–35, 2004.
- [7] Karlsson, J. and Ryan, K., A Cost-Value Approach for Prioritizing Requirements. *IEEE Software*, Vol. 14, No. 5, pp. 67–74, 1997.
- [8] Zapata, C. and Lezcano, L., Caracterización de los verbos usados en el diagrama de objetivos, *Dyna*, Vol. 76, No. 158, pp. 219–228, 2009.
- [9] Zapata, C., Lezcano, L. and Tamayo, P. Validación del método para la obtención automática del diagrama de objetivos desde esquemas preconceptuales. *Revista Escuela de Ingeniería de Antioquia*, Vol. 8, pp. 21–35, 2007.
- [10] Zapata, C., Giraldo, G. and Villamizar, K., A Method for Assigning Percentages to Cause-and-Effect Diagram in the UNC-Method Context. En: Leonardo López (Ed.), *Software Engineering: Methods, Modeling, and Teaching*, pp. 149–163, 2011.
- [11] Saaty, T., *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill, New York, 1980.
- [12] Zapata, C., *Un modelo de diálogo para la generación automática de especificaciones UN-Lencep*. EAE publishing, Saarbrücken, 2011.
- [13] Hughes, R., Expert Judgment as an Estimating Method. *Information and Software Technology*, Vol. 38, pp. 67–75, 1996.