2014 Second International Conference on Artificial Intelligence, Modelling and Simulation

Human Resource Assessment in Software Development Projects Using Fuzzy Linguistic 2-Tuples

Vassilis C. Gerogiannis
Dept. of Business Administration
Technological Educational Institute of Thessaly
Larissa, Greece
gerogian@teithessaly.gr

Elli Rapti
Institute of Research and Technology of Thessaly
Centre for Research and Technology
Volos, Greece
erapti@ireteth.certh.gr

Anthony Karageorgos
Dept. of Wood and Furniture Design and Technology
Technological Education Institute of Thessaly
Karditsa, Greece
karageorgos@teithessaly.gr

Panos Fitsilis
Dept. of Business Administration
Technological Education Institute of Thessaly
Larissa, Greece
fitsilis@teithessaly.gr

***Abstract*—Proper selection and allocation of human resources to software development tasks is one of the key challenges in software development projects. In this paper we present a fuzzy linguistic approach that supports the selection of suitable human resources based on their skills and the required skills for each project task. The proposed approach uses 2-tuple fuzzy linguistic terms and results in an objective aggregation of the ratings of required task related skills and provided skills from candidate human resources. The approach applies a group-based, multi-criteria, similarity degree-based aggregation algorithm. To reflect the contribution of one skill to the learning of other skills, the approach also considers possible relationships between skills. A numerical example is presented as a proof of concept to demonstrate the applicability of the approach.**

Keywords-software project management; human resource evaluation; 2-tuple fuzzy linguistic representation/computation model

I. INTRODUCTION

The problem of human resource allocation in software projects refers to the proper assignment of available human resources to various development tasks [1]. The process usually followed by software project management includes the division of the project effort into tasks, each one requiring specific skills, capabilities, and experience from the available human resources (e.g., analysts, programmers, testers etc.) [2]. Once the various development tasks to be performed have been defined, the most suitable candidates for each task should be selected according to task skill-related requirements [3]. Managing personnel in software projects still remains a complicated task due to the dynamic and complex context in which it takes place [4]. The problem of finding the "best human resource" is not always related to the optimal decision, since finding the "most suitable human resource" is required instead. Another key challenge is to achieve an, as much as possible, objective evaluation of skills of available human resources, according to various task-related skill requirements.

In dealing with the problem of knowledge/skills representation and evaluation in uncertain and imprecise settings, fuzzy logic [5] proves to be an efficient conceptual base, due to the fact that most human evaluation forms are approximate by their nature [3]. In this paper, we use the fuzzy linguistic 2-tuple representation/computation model [6] to build an assessment approach for human resources in software development tasks, according to provided/required skills/competencies. The presented approach is based on a group-based fuzzy multi-criteria method [7] that applies similarity degree-based aggregation to derive an objective assessment for provided/required skills/competencies. Since skills/competencies in software development are not always independent of each other (i.e., prior knowledge in various skills contributes to learning of other skills) [8], an advantageous characteristic of the proposed approach is the consideration of possible skill relationships and dependencies. The approach has been developed in the context of the SPRINT SMEs R&D project [9] that aims to suggest methods for software process improvement in the context of small and medium sized software development organizations.

The paper is organized as follows. In Section II we provide a brief overview of the relevant literature, while in Section III we describe the proposed approach for the evaluation and selection of human resources in software development tasks, providing a proof of concept example. In Section IV we briefly discuss upon the usefulness of the approach results and, finally, in Section V we conclude this work by presenting our future research plans.

II. OVERVIEW OF RELATED WORK

Various approaches have been proposed in the literature aiming to support the evaluation and allocation of personnel in software development projects. For example, in [8] the authors present the Best-Fitted Resource (BFR) methodology which considers how prior knowledge in various skills contributes to the learning of other skills. The BFR approach, although similar with the one presented in the current paper, does not take into account fuzziness and vagueness issues in characterizing capabilities and the levels of expertise required. There is also a lot of research focusing on use of methods from the area of computational intelligence, such as constraint satisfaction solving [2] and fuzzy logic [3, 10], to rank available developers according to how suitable they are

978-1-4799-7600-3/14 $31.00 © 2014 IEEE
DOI 10.1109/AIMS.2014.15

to certain tasks. However, in the relevant literature, we have found only few approaches for human resource evaluation in software projects which are based on fuzzy logic and also try to consider dependencies between skills. One such representative method is suggested in [3]. The main assumption of these approaches, such the one presented in [3], is that a software development organization maintains a knowledge base of fuzzy rules to describe, somehow arbitrarily, management knowledge about skill relationships and, consequently, follow a fuzzy inference mechanism to undertake human resource evaluation and decision. On the contrary, the presented approach is a group-based one that emphasizes on deriving subjective values for skill relationships and required/provided skill evaluations from corresponding objective expert judgments expressed by decision makers/project managers.

III. DESCRIPTION OF THE APPROACH

The proposed approach follows seven steps, which are described in the following sections accompanied with short illustrative examples, where appropriate.

Step A. Group-based linguistic evaluation of required skills

Assuming that a software development task T is planned to be executed as a set $T = \{x_1, x_2, \dots, x_n\}$ comprised by n individual development activities x, the approach applies group-based decision making by requiring from K project managers e_k $(k = 1, 2, \dots, K)$ to initially express levels of m skills c required for each individual activity to be completed successfully. Skill requirements are expressed in a qualitative form by utilizing the 2-tuple fuzzy linguistic terms approach as introduced in [6]. Specifically, the 2-tuple linguistic representation/computation model was chosen as the underlying basis of the suggested approach, as it can effectively avoid loss/distortion of information, an issue typical with other fuzzy linguistic methods when dealing with fuzzification/de-fuzzification of information [6, 11].

A 2-tuple linguistic variable is denoted as (s_i, a_i), where s_i corresponds to the central value of the i^{th} linguistic term in a term set and $a_i \in (-0.5, 0.5)$ is the distance from s_i. For example, let us assume that three project managers $(K = 3)$ evaluate four skills $(m = 4)$ (e.g., Object Oriented Design, C++, Visual Basic and Java) required to perform four activities $(n = 4)$ of a software development task. We assume that during this specific development task four software components have to be developed and, thus, there are, respectively, four development activities that have to be implemented. We further assume that in order to express their evaluations, project managers have used a linguistic label set $s = \{s_0, s_1, \dots, s_g\}$, where $g + 1$ is the granularity of the selected linguistic term set, which includes the following terms: $s_0 = VVL$ (Very Very Low), $s_1 = VL$ (Very Low), $s_2 = L$ (Low), $s_3 = M$ (Medium), $s_4 = H$ (High), $s_5 = VH$ (Very High), $s_6 = VVH$ (Very Very High). Project managers may also select different linguistic term sets (i.e., sets having different granularities or semantics) to express their evaluations on the required skills. In this case, all skill assessments have to be unified into a uniform linguistic term set by following the method proposed in [12, 13]. Since project managers evaluate each activity according to the required skills using linguistic terms from the term set s, the linguistic evaluation X_{ij} for an activity x_i $(i = 1, 2, \dots, n)$ with respect to each skill c_j $(j = 1, 2, \dots, m)$ is transformed into a 2-tuples of the form $(s_i, 0)$, according to the following transformation function [6]:

$$\theta: S \to S \times [0.5, 0.5),\ \theta(s_i) = (s_i, 0),\ s_i \in S \qquad (1)$$

Table I presents an example of project managers' evaluations in the form of 2-tuples for the levels of skills required for each one of four development activities $(n = 4)$ comprising a software development task. In each cell of Table I there are two evaluation values in the form of *tuple1*/*tuple2*, where *tuple1* and *tuple2* are both linguistic 2-tuples. The first tuple in each cell (*tuple 1*) corresponds to the judgment expressed by a project manager for the level of skill required from a human resource to perform an activity. The second tuple in each cell (*tuple 2*) corresponds to the judgment expressed by a project manager for the level of skill that characterizes a candidate human resource with respect to a required skill. These second tuples (i.e., *tuple 2* values) will be used in step 5 of the approach to derive an objective evaluation for the skills available from the candidate human resources. For example, according to project manager e_1 a 'Very Very High' level of competency in Java is required for activity x_1, expressed by the 2-tuple $(VVH, 0)$. In addition, project managers may have different expertise and background in managing software projects and assessing the needs of software development tasks; therefore, a different relative importance level-weight ε_k can be assigned to each project manager. In Table I we assume for simplicity reasons equal importance weights (i.e., each ε_k is equal to 1/3) for all three project managers involved in the case example.

Step B. Similarity degree-based aggregation of different skills' evaluations

By performing group-based linguistic evaluation, all skills required to perform a software development task are characterized by subjective project manager judgments. However, some of the provided judgments may be biased towards each required skill. To derive a more objective assessment, the proposed approach applies similarity degree-based aggregation as introduced in [7]. The final aggregated assessments consider not only the relative importance weights ε_k of project managers but also similarities in their respective evaluations. Therefore, the approach makes aggregation results to reflect the collective judgments of project managers more reasonably and more objectively. The similarity degree-based aggregation follows three sub-steps:

1) Similarity degree calculation: A similarity degree $sim(X_{ij}^k, X_{ij}^l) \in (0, 1]$ is calculated between any two skills' evaluations provided by two managers e_k and e_l $(k \neq l, k = 1, 2, \dots, K, l = 1, 2, \dots, K)$ for each activity x_i $(i = 1, 2, \dots, n)$ with respect to each skill c_j $(j = 1, 2, \dots, m)$. To calculate the similarity degree value, the distance between

X_{ij}^k and X_{ij}^l evaluations is computed, which is equal to $\left|\Delta^{-1}\left(X_{ij}^k\right) - \Delta^{-1}(X_{ij}^l)\right|$, where Δ^{-1} is the reverse function that transforms a 2-tuple linguistic variable into a numerical value [6].

In particular, given a linguistic term set s, $\beta \in [0, g]$ is a number representing the aggregation result of a symbolic aggregation operation. Let $i = round(\beta)$ and $\alpha = \beta - i$ be two values such that $i \in [0, g]$ and $a \in [-0.5, 0.5)$. The value a is the symbolic translation. The 2-tuple that expresses the equivalent information with the value β results from the translation function $\Delta(\beta)$ [6]:

$$\Delta: [0, g] \to S \times [-0.5, 0.5) \tag{2}$$

$$\Delta(\beta) = (s_i, a) = \begin{cases} s_i, i = round(\beta) \\ \alpha = \beta - i, a \in [-0.5, 0.5) \end{cases} \tag{3}$$

A 2-tuple linguistic variable can be transformed into an equivalent number $\beta \in [0, g]$ by the reverse function Δ^{-1} as follows [6]:

$$\Delta: S \times [-0.5, 0.5) \to [0, g] \tag{4}$$

$$\Delta^{-1}(s_i, a) = i + a = \beta \tag{5}$$

The similarity degree value $sim\left(x_{ij}^k, x_{ij}^l\right)$ is then computed according to the following formula [7]:

$$sim\left(X_{ij}^k, X_{ij}^l\right) = 1 - \left|\frac{\Delta^{-1}\left(X_{ij}^k\right) - \Delta^{-1}(X_{ij}^l)}{g}\right| \tag{6}$$

where $g + 1$ is the granularity of the used linguistic term set. The closer the similarity degree to 1, the more similar the evaluations of any two project managers are for the same activity with respect to a particular skill.

For example, considering the evaluations given by project managers e_1 and e_2 for activity x_2 with respect to skill c_1, that is expertise in object oriented design, $\Delta^{-1}(X_{21}^1) = 1$ and $\Delta^{-1}(X_{21}^2) = 0$ (Table I), the similarity degree between these two evaluations, according to (6) is $sim(X_{21}^1, X_{21}^2) = 0.83$. Accordingly, we calculate $sim(X_{21}^1, X_{21}^3)$ and $sim(X_{21}^2, X_{21}^3)$, which are equal to 1 and 0.83, respectively.

2) Average and relative similarity degree calculation: For each project manager, the average similarity degree $SM_{ij}(e_k)$ and the relative similarity degree $RSM_{ij}(e_k)$ are calculated, regarding the evaluation of each activity x_i $(i = 1, 2, \dots, n)$ with respect to each skill c_j $(j = 1, 2, \dots, m)$. These are respectively given by the following equations [7]:

$$SM_{ij}(e_k) = \frac{\sum_{l=1, l \neq k}^{K} sim\left(X_{ij}^k, X_{ij}^l\right)}{K - 1} \tag{7}$$

$$RSM_{ij}(e_k) = \frac{SM_{ij}(e_k)}{\sum_{l=1}^{K} SM_{ij}(e_l)} \tag{8}$$

As an example, having calculated in the previous step the similarity degrees for activity x_2 with respect to skill c_1 (object oriented design), the average similarity degree for each project manager according to (7) is $SM_{21}(e_1) = 0.92, SM_{21}(e_2) = 0.83$ and $SM_{21}(e_3) = 0.92$. Consequently, the relative similarity degree for each project manager according to (8) is $RSM_{21}(e_1) = 0.34$, $RSM_{21}(e_2) = 0.32$ and $RSM_{21}(e_3) = 0.34$.

3) Importance level calculation: The importance level w_{ij}^k for each project manager e_k is calculated by considering his/her relative importance weight ε_k and the relative similarity degree of his/her evaluations, as follows [7]:

$$w_{ij}^k = \frac{\varepsilon_k \times RSM_{ij}(e_k)}{\sum_{l=1}^{K}\left(\varepsilon_l \times RSM_{ij}(e_k)\right)} \tag{9}$$

Having assumed equal relative importance weights for all three project managers (i.e., each ε_k is equal to 1/3, Table I) and considering the calculated relative similarity degrees, we compute the importance level of the assessment of each project manager for activity x_2 with respect to skill c_1 using formula (9), that is $w_{21}^1 = w_{21}^2 = w_{21}^3 = 0.33$. The importance levels of the assessments of the three project managers are found to be equal, since, for simplicity reasons, they are assigned to equal relative importance weights. However, in the general case involving project managers with different relative importance weights, the levels of their assessments can be unequal.

Step C. Calculation of aggregated rating of importance for each required skill

The objective aggregation for all activities' ratings is computed by utilizing the weighted average operator, as defined for fuzzy linguistic 2-tuples in [6]. In particular, for a set of linguistic 2-tuples $\{(s_1, a_1), (s_2, a_2), \dots, (s_l, a_l)\}$ and their corresponding weights $\{w_1, w_2, \dots, w_l\}$, the 2-tuple weighted average operator $\bar{x}$ is computed as follows [6]:

$$\begin{aligned} \bar{x} &= \Delta\left(\frac{\sum_{i=1}^{l}(\Delta^{-1}(s_i, a_i) \times w_i)}{\sum_{i=1}^{l} w_i}\right) \\ &= \Delta\left(\frac{\sum_{i=1}^{l}(\beta_i \times w_i)}{\sum_{i=1}^{l} w_i}\right) \end{aligned} \tag{10}$$

In equation (10), β_i is calculated by the reverse function Δ^{-1} described in (5). The final aggregated rating FX_{ij} of each activity x_i $(i = 1, 2, \dots, n)$ with respect to each skill c_j $(j = 1, 2, \dots, m)$ can be computed by applying the weighted average operator on the linguistic evaluations of the activities and using as weights the previously calculated importance levels for these assessments. Thus, according to (10) the final aggregated ratings FX_{ij} are calculated as follows:

$$FX_{ij} = \Delta\left(\frac{\sum_{t=1}^{K}\left(\Delta^{-1}\left(X_{ij}^t, a_{ij}^t\right) \times w_{ij}^t\right)}{\sum_{t=1}^{K} w_{ij}^t}\right) \tag{11}$$

The final aggregated ratings for all activities with respect to the various required skills are presented in Table II.

Step D. Task profile evaluation with respect to skill requirements

Since a software development task is composed by a number of activities, an overall "profile" can be created for the composite development task as a vector of linguistic 2-tuples. This profile represents the level of resource skills required for the task successful implementation, according to the project managers' evaluations. The task profile tp_j with respect to each required skill c_j can be calculated by applying the weighted average operator (10) to the previously calculated final aggregated ratings of skills (11) and using as weights the importance degrees I_i of the development activities x_i which comprise the software development task. The importance degree I_i of a development activity x_i (Table II, column 2) reflects the value-priority of the software component that results from the activity implementation. Thus, the task profile tp_j is computed according to the following formula

$$tp_j = \Delta\left(\frac{\sum_{i=1}^{n}\left(\Delta^{-1}\left(FX_{ij}\right) \times I_i\right)}{\sum_{l=1}^{n} I_i}\right) \qquad (12)$$

where n is the total number of activities for each task. The resulted task profile for all individual required skills is calculated as a vector of linguistic 2-tuples and it is presented in the second column of Table III. From this specific task profile, we can conclude that, for this specific task, high level knowledge in C++ is required (i.e., the corresponding 2-tuple is equal to $(H, 0.14)$) and not so high-level knowledge in Visual Basic and Java (i.e., the corresponding 2-tuples are equal to $(L, 0.24)$ and $(VL, 0.34)$, respectively).

Step E. Linguistic evaluation of skills available from candidate human resources

After having calculated the task profile, we continue by evaluating candidate human resources according to their available skills with respect to the specific task required skills. To consider and evaluate objectively the capability/suitability of q candidate human resources r with respect to the task required skills c_j $(j = 1, 2, \dots, m)$, the previous steps are repeated. In particular, each project manager evaluates all candidate human resources according to their level of knowledge on different required skills using a linguistic label set $s = \{s_0, s_1, \dots, s_g\}$. The linguistic evaluations R_{ij} of resources according to their skills are then transformed into 2-tuples in the form $(s_i, 0)$ according to (1). In Table I the second tuple in each cell (*tuple 2*) corresponds to the judgment expressed by the corresponding project manager for the level of skill of each one human resource from the set of four candidates ($q = 4$) with respect to each required skill. For example, according to project manager e_1, human resource r_1 is characterized by a 'Very Low' level of knowledge regarding object oriented design.

To derive an objective assessment for these judgments, the similarity degree value between any two project managers' evaluations is calculated using formula (6). The importance level is calculated according to (9), using the average and relative similarity degree, calculated by (7) and (8), respectively. Then, a final aggregated rating FR_{ij} of each resource r_i $(i = 1, 2, \dots, q)$ with respect to each skill c_j $(j = 1, 2, \dots, m)$ is calculated according to (11). Finally, the capability/suitability of each resource cs_i $(i = 1, 2, \dots, q)$ is computed by applying the weighted average operator (10) on the final aggregated rating FR_{ij}, using as weights the previously calculated task profile assessments tp_j $(j = 1, 2, \dots, m)$ (12) for each individual required skill.

In the presented example, the final aggregated ratings FR_{ij} of each resource r_i with respect to each skill c_j are shown in columns 3-6 of Table III. The capability/suitability of each resource cs_i is shown in the seventh column of Table III.

Step F. Consideration of skills' relationships

Skills/competencies in software development are not always independent of each other. On the contrary, prior knowledge in various skills contributes to the learning of other skills [8]. For example, prior knowledge in object oriented design can be considered helpful to develop skills in C++ programming. In this step of the approach we consider skill relationships, which represent the level to which knowledge on one skill contributes to the improvement (via learning) of another skill. To this end, each manager evaluates subjectively skill relationships and a skill-relationships table is constructed, where relationships between skills are depicted in linguistic terms using a linguistic label set $s = \{s_0, s_1, \dots, s_g\}$, as shown in Table III. For example, according to project manager e_1, existing competency in C++ programming contributes at a 'Very Very High' level to improve skills in object oriented design.

To evaluate objectively the skill-relationships with respect to the task required skills c_j $(j = 1, 2, \dots, m)$, the previous steps are repeated again. Specifically, the linguistic evaluations C_{ij} of skill-relationships are transformed into 2-tuples in the form $(s_i, 0)$ according to (1). To derive a more objective assessment, the similarity degree values between the project managers' evaluations are calculated using formula (6). The importance levels are calculated according to (9), using the average and relative similarity degree calculated by (7) and (8) respectively. Finally, the objective skill-relationships are extracted through the final aggregated rating FSR_{ij} of each relationship between any two skills $(c_j, j = 1, 2, \dots, m)$. The final aggregated ratings FSR_{ij} of skill-relationships are calculated according to (11) and they are presented in Table IV.

Step G. Re-evaluation of the capabilities of human resources

As a last step of the approach, the capabilities of human resources on each required skill need to be re-evaluated according to the final aggregated rating of skill-relationships, which were calculated in the previous step. A new value is computed for the capability/suitability of each human resource, which results as the maximum value between the previously calculated capability of a skill and the weighted

average contribution on that skill from other skills as follows:

$$FR_{ij}^{NEW} = max\left(FR_{ij}; \frac{\sum_{h=1,h\neq j}^{m}\left(FR_{ih} \times FSR_{hj}\right)}{\sum_{h=1,h\neq j}^{m} FSR_{hj}}\right) \quad (13)$$

Consequently, the re-evaluated final aggregated rating FR_{ij} of each human resource r_i $(i = 1, 2, \dots, q)$ with respect to each skill c_j $(j = 1, 2, \dots, m)$ is computed by applying the weighted average operator (10) on the re-evaluated linguistic evaluations of the human resources, using as weights the previously calculated task profile assessments for each individual skill. The re-evaluated resource capabilities are presented in the last column of Table III. The comparison between the initial ratings (seventh column in Table III) and the final ratings (last column in Table III) of the candidate resources shows that definitely the most suitable candidate human resource to be involved in the activities of the development task is resource r_4.

IV. Discussion

Systematic utilization of all available human resources in a software development project is a very important issue. To address this issue, we introduced a method for the efficient evaluation and selection of human resources in a software development project. Using subjective evaluations on skills required for specific task activities, available human resource skills and relationships between the various skills, all expressed by project managers in a qualitative linguistic form, we are able to extract objective evaluation on resource capabilities and their suitability for the respective development task. Specifically, by considering the skill relationships which reflect the degree to which one skill contributes to the learning of other skills, the difference between the most suited human resource and the rest available resources for a specific task can be intensified, thus better indicating the most appropriate candidate for the specific task. According to the results of the proof of concept example, we can conclude that in the initial evaluation of the candidate human resources (Table III, seventh column) both resources r_3 and r_4 are assumed almost highly suitable for the task skill requirements, with ratings 3.64 and 3.94, respectively. However, due to the re-evaluation considering skill relationships, r_4 ends up being more suitable (rated as $VH(4.96)$) with greater difference from r_3 (Table III, last column).

V. Conclusion

In this paper we presented a fuzzy linguistic approach based on 2-tuple fuzzy linguistic terms for the assessment of human resources involved in the development tasks of a software development project. The method follows a group and similarity degree-based aggregation algorithm to obtain an objective aggregation of the ratings of multiple required task related skills and provided skills from the available human resources. In addition, skill relationships are evaluated as 2-tuple fuzzy linguistic terms to represent dependencies between the various task-related skills and reflect the contribution of one skill to the learning of other skills.

As a future work we plan to further improve the suggested approach by determining resource teams based on skill substitution and complementarity associations between candidate human resources. In addition, we aim to address the provision of appropriate support to the allocation of human resources to software development tasks by performing multi-objective optimization (according to budget and availability constraints) and by applying bio-inspired optimization approaches.

Acknowledgment

This research has been co-financed by the European Union (European Social Fund) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)-Research Funding Program: ARCHIMEDES III, Investing in knowledge society through the European Social Fund, under the "SPRINT SMEs" project (Research in software process improvement methodologies for Greek small medium sized software development enterprises).

References

[1] T. J. A. Santos, A. M. Lima, C. A. L. Reis, and R. Q. Reis, "Automated support for human resource allocation in software process by cluster analysis," in *Proceedings of the 4th International Workshop on Recommendation Systems for Software Engineering*, 2014, pp. 30-31.

[2] A. Barreto, M. d. O. Barros, and C. M. Werner, "Staffing a software project: A constraint satisfaction and optimization-based approach," *Computers & Operations Research,* vol. 35, pp. 3073-3089, 2008.

[3] D. A. Callegari and R. M. Bastos, "A Multi-criteria Resource Selection Method for Software Projects Using Fuzzy Logic," in *Enterprise Information Systems*, ed: Springer, 2009, pp. 376-388.

[4] L. C. e Silva and A. P. C. S. Costa, "Decision model for allocating human resources in information system projects," *International Journal of Project Management,* vol. 31, pp. 100-108, 1// 2013.

[5] L. A. Zadeh, "Fuzzy sets,"*Information and control,* vol. 8, pp. 338-353, 1965.

[6] F. Herrera and L. Martínez, "A 2-tuple fuzzy linguistic representation model for computing with words," *Fuzzy Systems, IEEE Transactions on,* vol. 8, pp. 746-752, 2000.

[7] X. Liao, Y. Li, and B. Lu, "A model for selecting an ERP system based on linguistic information processing," *Information Systems,* vol. 32, pp. 1005-1017, 2007.

[8] L. D. Otero, G. Centeno, A. J. Ruiz-Torres, and C. E. Otero, "A systematic approach for resource allocation in software projects," *Computers & Industrial Engineering,* vol. 56, pp. 1333-1339, 2009.

[9] *SPRINT SMEs Project*. Available at: http://sprint.teilar.gr/

[10] N. A. Ruskova, "Decision support system for human resources appraisal and selection," in *Intelligent Systems, 2002. Proceedings. 2002 First International IEEE Symposium*, 2002, pp. 354-357.

[11] F. Herrera and L. Martinez, "The 2-tuple linguistic computational model: advantages of its linguistic description, accuracy and consistency," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 9, pp. 33-48, 2001.

[12] F. Herrera, E. Herrera-Viedma, and J. Verdegay, "A rational consensus model in group decision making using linguistic assessments," *Fuzzy Sets and Systems,* vol. 88, pp. 31-49, 1997.

[13] F. Herrera, L. Martınez, and P. J. Sánchez, "Managing non-homogeneous information in group decision making," *European Journal of Operational Research,* vol. 166, pp. 115-132, 2005.

TABLE I. PROJECT MANAGER EVALUATIONS ON REQUIRED SKILLS (x) / RESOURCES (r)

Activity (x)/ Resource (r)	Project Manager (e)/Weight of Importance (ε)	Levels of Skills Required for a Task / Skills Available from Candidate Resources (c)			
		OO design (c_1) ***tuple1/tuple2***	***C++*** (c_2) ***tuple1/tuple2***	***VB*** (c_3) ***tuple1/tuple2***	***Java*** (c_4) ***tuple1/tuple2***
x_1/r_1	$e_1/(1/3)$	H(4, 0)/VL(1, 0)	L(2, 0)/L(2, 0)	VL(1, 0)/ H(4, 0)	VVH(6, 0)/VH(5, 0)
	$e_2/(1/3)$	VVH(6, 0)/L(2, 0)	VL(1, 0)/VL(1, 0)	M(3, 0)/VH(5, 0)	VH(5, 0)/M(3, 0)
	$e_3/(1/3)$	VVH(6, 0)/ VL(1, 0)	VVL(0, 0)/ VVL(0, 0)	M(3, 0)/ VH(5, 0)	M(3, 0)/ L(2, 0)
x_2/r_2	$e_1/(1/3)$	VL(1, 0)/VL(1, 0)	VL(1, 0)/ VL(1, 0)	VH(5, 0)/VH(5, 0)	VL(1, 0)/L(2, 0)
	$e_2/(1/3)$	VVL(0, 0)/VL(1, 0)	VL(1, 0)/VL(1, 0)	VVH(6, 0)/ H(4, 0)	L(2, 0)/L(2, 0)
	$e_3/(1/3)$	VL(1, 0)/ VVL(0, 0)	L(2, 0)/ VL(1, 0)	VH(5, 0)/VH(5, 0)	L(2, 0)/ L(2, 0)
x_3/r_3	$e_1/(1/3)$	VH(5, 0)/M(3, 0)	VH(5, 0)/VH(5, 0)	L(2, 0)/H(4, 0)	L(2, 0)/L(2, 0)
	$e_2/(1/3)$	VVH(6, 0)/ H(4, 0)	M(3, 0)/VH(5, 0)	L(2, 0)/ L(2, 0)	VL(1, 0)/VL(1, 0)
	$e_3/(1/3)$	VVH(6, 0)/ L(2, 0)	VVH(6, 0)/VVH(6, 0)	L(2, 0)/ M(3, 0)	VL(1, 0)/VL(1, 0)
x_4/r_4	$e_1/(1/3)$	L(2, 0)/VH(5, 0)	VVH(6, 0)/VVH(6, 0)	VL(1, 0)/VL(1, 0)	VL(1, 0)/VL(1, 0)
	$e_2/(1/3)$	H(4, 0)/VVH(6, 0)	VH(5, 0)/VH(5, 0)	VL(1, 0)/VL(1, 0)	VL(1, 0)/VL(1, 0)
	$e_3/(1/3)$	VH(5, 0)/VH(5, 0)	VH(5, 0)/VH(5, 0)	L(2, 0)/ VL(1, 0)	VVL(0, 0)/VVL(0, 0)

TABLE II. FINAL AGGREGATED RATINGS OF ACTIVITIES (FX) AND TASK PROFILE (TP)

Activity (x)	Activity Importance Degree (I)	Required Skills (c)				Task Profile (tp)
		OO design (c_1)	***C++*** (c_2)	***VB*** (c_3)	***Java*** (c_4)	
x_1	VL(1, 0)	5.43/(5, 0.43)	1/(1, 0)	2.43/(2, 0.43)	4.75/(5, −0.25)	$3.87/H(4, -0.13)$
x_2	L(2, 0)	0.66/(1, −0.34)	1.31/(1, 0.31)	5.31/(5, 0.31)	1.69/(2, −0.31)	$4.14/H(4, 0.14)$
x_3	M(3, 0)	5.69/(6, −0.31)	4.75/(5, −0.25)	2/ (2, 0)	1.31/(1, 0.31)	$2.24/L(2, 0.24)$
x_4	VVH(6, 0)	3.75/(4, −0.25)	5.31/(5, 0.31)	1.31/(1, 0.31)	0.69/(1, −0.31)	$1.34/VL(1, 0.34)$

TABLE III. FINAL AGGREGATED RATINGS OF RESOURCES (FR) / RE-EVALUATED RATINGS (FR^{NEW}) AND RESOURCE CAPABILITIES ASSESSMENT (CS)

Resource (r)	Task Profile (tp)	Required Skills (c)				Resource Capabilities (cs)	Resource Capabilities (cs) (re-evaluated)
		OO design (c_1) (FR/FR^{NEW})	***C++*** (c_2) (FR/FR^{NEW})	***VB*** (c_3) (FR/FR^{NEW})	***Java*** (c_4) (FR/FR^{NEW})		
r_1	H(4, −0.13)	1.31/2.41	1/2.76	4.69/4.69	3.25/3.25	2.07/L	3.08/M
r_2	H(4, 0.14)	0.69/1.92	1/2.06	4.69/4.69	2/2	1.73/L	2.51/M
r_3	L(2, 0.24)	3/3.41	5.31/5.31	3/3.27	1.31/3.95	3.64/H	4.13/H
r_4	VL(1, 0.34)	5.31/5.31	5.31/5.31	1/3.87	0.69/4.67	3.94/H	4.96/VH

TABLE IV. PROJECT MANAGER EVALUATIONS (C) AND FINAL AGGREGATED RATINGS OF SKILL RELATIONSHIPS (FSR)

Required Skills (c)	Required Skills (c)							
	OO design (c_1)		***C++*** (c_2)		***VB*** (c_3)		***Java*** (c_4)	
	C	FSR	C	FSR	C	FSR	C	FSR
OO design (c_1)	--	--	VH(5, 0)	5/VH(5, 0)	M(3, 0)	2.69/M(3, −0.31)	VH(5, 0)	5/VH(5, 0)
	--		H(4, 0)		L(2, 0)		VH(5, 0)	
	--		VVH(6, 0)		M(3, 0)		VH(5, 0)	
C++ (c_2)	VVH(6, 0)	5.43/VH(5, 0.43)	--	--	L(2, 0)	2.69/M(3, −0.31)	VH(5, 0)	4.69/VH(5, −0.31)
	H(4, 0)		--		M(3, 0)		H(4, 0)	
	VVH(6, 0)		--		M(3, 0)		VH(5, 0)	
VB (c_3)	L(2, 0)	1.69/L(2, −0.31)	M(3, 0)	2.69/M(3, −0.31)	--	--	L(2, 0)	1.69/L(2, −0.31)
	L(2, 0)		L(2, 0)		--		VL(1, 0)	
	VL(1, 0)		M(3, 0)		--		VL(1, 0)	
Java (c_4)	H(4, 0)	4.57/VH(5, −0.43)	VH(5, 0)	4.31/H(4, 0.31)	VL(1, 0)	2.43/L(2, 0.43)	--	--
	H(4, 0)		H(4, 0)		M(3, 0)		--	
	VVH(6, 0)		H(4, 0)		M(3, 0)		--	