Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management
Dubai, United Arab Emirates (UAE), March 3 – 5, 2015

A fuzzy linguistic approach for human resource evaluation and selection in software projects

Vassilis C. Gerogiannis
Dept. of Business Administration
Technological Education 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*—One of the key challenges in software projects is the efficient allocation of human resources to software development tasks. To achieve this challenge, the proper human resource evaluation and selection is an important step. In this paper we present a fuzzy linguistic approach that utilizes 2-tuple fuzzy linguistic terms and supports the selection of suitable human resources based on their skills and the required skills for each project task. The proposed approach follows a group, multi-criteria, similarity degree-based aggregation algorithm and results in an objective aggregation of the ratings of required task related skills and provided skills from candidate human resources. In addition, we consider skill relationships as 2-tuple fuzzy linguistic terms to reflect the contribution of one skill to the learning of other skills. The proposed approach has been implemented in the context of SPRINT SMEs project. SPRINT SMEs aims to suggest a practical framework of methods for the improvement of software processes which take place in small & medium sized software development enterprises.**

Keywords—software project management; human resource evaluation; 2-tuple fuzzy linguistic model; similarity-degree based aggregation

I. INTRODUCTION

The human resource allocation problem in a software project refers to the proper assignment of available human resources to the development tasks [1]. Software development organizations usually follow a process that divides the project development effort into tasks, each one requiring specific skills, capabilities, and experience from the available human resources [2]. Once the various tasks to be performed have been defined, the appropriate resources for the project must be selected and suitable candidates for each task should be found according to specific skill-related requirements [3]. The assignment of roles and the consequent formation of project teams rely on project managers' subjective judgments, project constraints (e.g., budget and timing constraints), people constraints (e.g., availability) and skill-related requirements, and all these information often is not systematically recorded and handled [4]. Proper resource evaluation and selection is an important decision in project management and software project management, in particular [3]. This decision is not just crucial for the generation of efficient software teams, but it is of strategic importance in software development organizations since it can support them to develop competitive advantages by exploiting long term competences of available personnel and implement successful projects [5]. Many general approaches discussed in the literature, regarding resource selection and allocation, handle resources in a quantitative manner, where all resources are treated as equally skilled [3]. However, in human centered activities, such as software development projects, personnel assignment decisions considering individual level of skills of each available resource can clearly contribute to project success [6]. One expected benefit derived from improvements in a resource allocation process is the decrease in project tasks' durations, an issue which further increases a company's productivity [7].

Despite all research and advances in the field, managing personnel in software projects still remains a very complicated task due to the dynamic and complex context in which it takes place [5]. The projects manager/decision-maker has to face a number of different possible staff combinations and many correlated variables (usually conflicting), such as time, cost, quality, and so forth, and respectively make the best decision under pressure and high demands [1]. In addition, a major contributor to this complexity is the increased demand for specialized individual skills in the workforce, which results from high turnover rates and the fast pace at which new technologies and techniques are being developed. As a result of higher demands, candidates with exact required skills to work tasks are usually not available [6]. However, finding the "best resource" is not always related with the optimal decision. For example, if a software development task demands "high skill level for Java" and "low skill level for C++", then both requirements should be addressed by considering the respective levels of required knowledge. Thus, a resource who meets both criteria is considered to be more suitable to the task as opposed to another resource with very high level skills in both Java and C++, since such a resource could be assigned to a more demanding development task [3]. Moreover, the lack of systematic methods to assess personnel capabilities often forces decision makers to assign

978-1-4799-6065-1/15/$31.00 ©2015 IEEE

resources to tasks based only on subjective measures. Subjective decisions may result in selecting inadequate resources which require more time for training, a fact that consequently affects the schedule of projects. Particularly, in skill-based working environments, such as software development projects, where the objective of the development tasks is to provide customized and non-repetitive solutions to end-users/customers, the risk of utilizing inadequate resources is high, which further results in excessive project costs and high probability for developing unsuccessful software products/services [8].

Thus, a key problem in software development is to achieve an, as much as possible, objective evaluation of knowledge and skills of available human resources, according to various task-related skill requirements needed by a software development organization to achieve its goals. Many approaches do not address efficiently this problem when it comes to resource knowledge/skills representation and evaluation, by employing a conventional two-valued logic for characterizing required skill requirements, which proves to be very inefficient when dealing with the uncertainty and vagueness of this characterization. In dealing with the problem of knowledge/skills representation and evaluation in uncertain and imprecise settings, fuzzy logic [9] proves to be an efficient conceptual base, due to the fact that most human reasoning and evaluation forms are approximate by their nature [3]. In this paper, we use the fuzzy linguistic 2-tuple representation/computation model [10] to build an evaluation and selection approach for human resources in software development tasks, according to provided/required skills/competencies. The proposed approach is based on a group-based fuzzy multi-criteria method [11] that applies similarity degree-based aggregation to derive an objective assessment for provided/required skills/competencies. Since skills/competencies in software development are not independent of each other (i.e., prior knowledge in various skills contributes to learning of other skills) [6], an important contribution of the suggested approach is that it emphasizes on deriving objective assessments of human resources matching skills required by software tasks by considering possible skill relationships and dependencies. The approach has been developed in the context of the SPRINT SMEs project [12], an R&D project that aims to suggest techniques for software process improvement tailored to the needs of small and medium sized software development enterprises.

The remainder of the paper is structured as follows. In Section 2 we provide an overview of the SPRINT SMEs project, while in Section 3 we describe the proposed approach for the evaluation and selection of human resources in software development tasks. Along with the approach description, in Section 2 we also present a proof of concept example. In Section 4 we discuss upon the usefulness of the approach results. An overview of the relevant literature work is given in Section 5, while in Section 6 we conclude the paper and present our future research plans.

II. SPRINT SMEs Project

Software Process Improvement (SPI) of Small Medium Enterprises (SMEs) is gaining momentum in software research and industry. SPI is a challenging process for most SMEs aiming at preventing project failures and delivering high quality software products/services consistent with end-customers' needs. The objective of SPRINT SMEs project (Research in Software PRocess ImprovemeNT Methodologies for Small & Medium sized Software Development Enterprises) is to propose and develop a practical framework of methods for supporting the improvement of processes which take place in small and medium sized software development enterprises (SW SMEs). The framework emphasizes on suggesting rigorous decision making methods for supporting process assessment and improvement problems met by SW SMEs. The framework of methods concentrates on problems in selected process domains such as requirements engineering, project planning and project staffing [13].

For project staffing problems, in particular, a systematic approach for evaluating the suitability of candidate human resources for each software development task is an imperative issue for a successful implementation of SPI [4]. While mid-size to large companies are characterized by a large pool with a considerable number of available resources with different skills to choose from, small to mid-size companies need to follow a tailored decision making process to effectively utilize the limited human resources available [3]. The aim of SPRINT SMEs project is to take into consideration the particular needs of a SW SMEs and propose easily applied and systematic approaches to help improving the domain of software project staffing and human resource allocation processes.

III. Evaluating Human Resources in Software Development Tasks

The proposed approach for human resource evaluation and selection in software development tasks implemented by a SW SME includes seven distinct steps, which are described in detail in the following sections accompanied with short illustrative examples, where appropriate.

Step 1: Group-based linguistic evaluation of required skills/competencies

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_i, $i = 1, 2, \dots, n$, the proposed approach applies a group-based decision method by requiring from K project managers/evaluators e_k ($k = 1, 2, \dots, K$) to initially express levels of m skills/competencies c_j, $j = 1, 2, \dots, m$, required for each individual activity to be completed successfully. Skill/competency requirements are expressed in a qualitative form by utilizing the 2-tuple fuzzy linguistic terms approach as introduced in [10]. 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 available information [10, 14].

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 different skills/competencies ($m = 4$) (e.g., Object Oriented Design, C++, Visual Basic and Java) which are 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 (i.e., by following a component-based software development paradigm) 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 assessment of the required skills. In this case, all linguistic assessments have to be transformed and expressed into a uniform linguistic term set by following the method proposed in [15, 16].

Since project managers evaluate each activity according to the required skills/competencies 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/competency 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 [10]:

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

Table I presents an example of project managers' evaluations in the form of 2-tuples for the level of skills/competencies required for each one of four development activities ($n = 4$) which comprise the 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 a development 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 perform objective evaluation of 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 have assumed for simplicity reasons equal importance levels-weights for all three project managers (i.e., each ε_k is equal to 1/3) which are involved in the case-study example.

Step 2: 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 judgments of project managers. However, some of the provided judgments may be biased towards each required skill/competency. To derive a more objective assessment, the proposed approach adopts the similarity degree-based aggregation technique as introduced in [11]. The final aggregated assessment considers not only the relative importance levels-weights ε_k of project managers but also similarities in their respective evaluations. Therefore, it can make aggregation results to reflect the collective judgments of project managers more reasonably and more objectively. The aggregation technique follows three distinct sub-steps that are respectively presented in the following subsections.

A. Similarity degree calculation

Initially, a similarity degree value $sim(x_{ij}^k, x_{ij}^l) \in (0, 1]$ is calculated between any two skills' evaluations provided by any two managers e_k and e_l $(k \neq l, k = 1, 2, \dots, K, l = 1, 2, \dots, K)$ for each development activity x_i $(i = 1, 2, \dots, n)$ with respect to each skill/competency 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 $\left|\Delta^{-1}(x_{ij}^k) - \Delta^{-1}(x_{ij}^l)\right|$, where Δ^{-1} is the reverse function that transforms a 2-tuple linguistic variable into its equivalent numerical value [10].

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 $\alpha \in [-0.5, 0.5)$, respectively. The number α is called a symbolic translation. The 2-tuple that expresses the equivalent information with β results from the translation function $\Delta(\beta)$ as follows [10]:

$$\Delta: [0, g] \to S \times [-0.5, 0.5) \qquad (2)$$

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

A 2-tuple linguistic variable can be transformed into an equivalent number $\beta \in [0, g]$ by the reverse function Δ^{-1} as follows [10]:

$$\Delta: S \times [-0.5, 0.5) \to [0, g] \qquad (4)$$

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

The similarity degree value $sim(x_{ij}^k, x_{ij}^l)$ is then computed according to the following formula [11]:

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

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 (Table I), the similarity degree between these two evaluations, according to (6) is:

$$sim(x_{21}^1, x_{21}^2) = 1 - \left|\frac{\Delta^{-1}(x_{21}^1) - \Delta^{-1}(x_{21}^2)}{g}\right| = 1 - \left|\frac{1-0}{6}\right| = 0.83$$

The values of $sim(x_{21}^1, x_{21}^3)$ and $sim(x_{21}^2, x_{21}^3)$ are computed in the same manner and they are equal to 1 and 0.83, respectively.

B. 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/competency c_j $(j = 1, 2, \dots, m)$. These are respectively given by the following equations [11]:

$$SM_{ij}(e_k) = \frac{\sum_{l=1, l\neq k}^{K} sim(x_{ij}^k, x_{ij}^l)}{K-1} \quad (7)$$

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

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), using formula (7) we can initially calculate the average similarity degree as follows:

$$SM_{21}(e_1) = \frac{sim(x_{21}^1, x_{21}^2) + sim(x_{21}^1, x_{21}^3)}{K-1} = \frac{0.83+1}{3-1} = 0.92$$

$SM_{21}(e_2)$ and $SM_{21}(e_3)$ are equal to 0.83 and 0.92, respectively. The relative similarity degree for project manager e_1 according to (8) is:

$$RSM_{21}(e_1) = \frac{SM_{21}(e_1)}{SM_{21}(e_1) + SM_{21}(e_2) + SM_{21}(e_3)} = \frac{0.92}{0.92+0.83+0.92} = 0.34$$

Following the same procedure we can calculate $RSM_{21}(e_2)$ and $RSM_{21}(e_3)$ which are equal to 0.32 and 0.34, respectively.

C. Importance weight calculation

The importance weight w_{ij}^k of the evaluation/assessment of project manager e_k is calculated by considering his/her relative importance level-weight ε_k and the relative similarity degree of his/her evaluations, as follows [11]:

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

Having assumed equal importance levels for all three project managers (i.e., each ε_k is equal to 1/3, Table I), we apply formula (9) to compute the weight of the assessment of project manager e_1 for activity x_2 with respect to skill c_1 as follows:

$$w_{21}^1 = \frac{\varepsilon_1 \times RSM_{21}(e_1)}{\varepsilon_1 \times RSM_{21}(e_1) + \varepsilon_2 \times RSM_{21}(e_1) + \varepsilon_3 \times RSM_{21}(e_1)} = \frac{1/3 \times 0.34}{1/3 \times 0.34 + 1/3 \times 0.34 + 1/3 \times 0.34} = 0.33$$

The weight of the assessment of project managers e_2 and e_3 are also equal to 0.33, since, for simplicity reasons, they are assigned to equal importance levels ε_k. However, in the general case involving project managers with different importance levels, the weights of their assessments can be unequal.

Step 3: 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 [10]. In particular, for a set of linguistic 2-tuples $\{(s_1, a_1), (s_2, a_2), \dots, (s_n, a_n)\}$ and their corresponding weights $\{w_1, w_2, \dots, w_n\}$, the 2-tuple weighted average operator $\bar{x}$ is computed as follows [10]:

$$\bar{x} = \Delta\left(\frac{\sum_{l=1}^{n}(\Delta^{-1}(s_i, a_i) \times w_i)}{\sum_{l=1}^{n} w_i}\right) = \Delta\left(\frac{\sum_{l=1}^{n}(\beta_i \times w_i)}{\sum_{l=1}^{n} w_i}\right) \quad (10)$$

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/competency 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 weights for these assessments. Thus, according to (10) the final aggregated rating FX_{ij} is calculated as follows:

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

For example, final aggregated rating for activity x_2 with respect to skill c_1 is:

$$FX_{21} = \Delta\left(\frac{\Delta^{-1}(X_{21}^1) \times w_{21}^1 + \Delta^{-1}(X_{21}^1) \times w_{21}^2 + \Delta^{-1}(X_{21}^3) \times w_{21}^3}{0.33+0.33+0.33}\right) = \Delta\left(\frac{1 \times 0.33 + 0 \times 0.33 + 1 \times 0.33}{0.33+0.33+0.33}\right) = \Delta(0.66) = (1, -0.34) = (VL, -0.34)$$

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

Step 4: Evaluation of the task profile with respect to skill requirements

Since a software development task is composed by a number of activities which result in development of different software components, 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/competency c_j can be calculated by applying the weighted average operator (10) to the previously calculated final aggregated ratings of skills 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. For example, we can assume that project managers agree that for the overall success of the software development task the development of the user interface component is an activity of higher importance compared to the development of the data base component.

Having calculated the final aggregated rating for each activity with respect to skill c_1, object oriented design (Table II, column 3), the task profile with respect to skill c_1 is equal to:

$$tp_2 = \Delta\left(\frac{5.43 \times 1 + 0.66 \times 2 + 5.69 \times 3 + 3.75 \times 6}{1 + 2 + 3 + 6}\right) = \Delta(3.87) = (4, -0.13) = (H, -0.13)$$

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 task profile, we can then conclude that for the 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 5: Linguistic evaluation of skills available from candidate human resources

After having calculated the task profile, we continue by evaluating candidate human resources (e.g., software programmers/developers) according to their available skills/competencies with respect to the specific task required skills. To consider and evaluate objectively the capability/suitability of q candidate human resources r_i $(i = 1, 2, \ldots, q)$, with respect to the task required skills c_j $(j = 1, 2, \ldots, m)$, steps 1 to 4 are repeated. Specifically, each project manager evaluates all candidate human resources according to their level of knowledge on different required skills/competencies using a linguistic label set $s = \{s_0, s_1, \ldots, s_g\}$. 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/knowledge of each one from the four candidate human resources $(q = 4)$ 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 weight 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 $(i = 1, 2, \ldots, q)$ with respect to each skill c_j $(j = 1, 2, \ldots, m)$ is calculated according to (11). Finally, the capability/suitability of each resource cs_i $(i = 1, 2, \ldots, q)$ 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 $(j = 1, 2, \ldots, m)$ 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 last column of Table III.

Step 6: 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 [6]. 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/competency contributes to the improvement (via learning) of another skill/competency. To this end, each manager evaluates subjectively skill relationships and a skill/competency-relationships table is constructed, where relationships between skills/competencies are expressed in linguistic terms using a linguistic label set $s = \{s_0, s_1, \ldots, s_g\}$, as shown in Table IV. 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/competency-relationships with respect to the task required skills c_j $(j = 1, 2, \ldots, m)$, steps 1 to 4 are repeated again. Initially, the linguistic evaluations c_{ij} of skill/competency-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 weight is calculated according to (9), using the average and relative similarity degree calculated by (7) and (8) respectively. Finally, the objective skill/competency-relationships are extracted through the final aggregated rating FSR_{ij} of each relationship between any two skills/competencies $(c_j, j = 1, 2, \ldots, m)$. The final aggregated rating FSR_{ij} of skill/competency-relationships is calculated according to (11) as presented in Table V.

Step 7: Re-evaluation of the capabilities of human resources

As a last step of the approach, the capabilities of human resources on each required skill/competency need to be re-evaluated according to the final aggregated rating of skill/competency-relationships, which were calculated in step 6. 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/competency from other skills/competencies as follows:

$$FR_{ij}^{NEW} = max\left(FR_{ij}; \frac{\sum_{h=1,h\neq j}^{m}\left(FR_{ih} \times FSR_{hj}\right)}{\sum_{h=1,h\neq j}^{m} FSR_{hj}}\right) \quad (12)$$

For example, the final aggregated rating FR_{11} of resource r_1 with respect to skill/competency c_1 is equal to 1.31 (Table III). According to (12) the re-evaluated FR_{11} according to the skill/competency relationships is higher (Table VI):

$$FR_{11}^{NEW} = max\left(1.31; \frac{1 \times 5.43 + 4.69 \times 1.69 + 3.25 \times 4.57}{5.43 + 1.69 + 44.57}\right) = max(1.31; 2.41) = 2.41$$

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/competency 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/capability. The re-evaluated resource capabilities are presented in the last column of Table VI. The comparison between the initial ratings (last column in Table III) and the final ratings (last column in Table VI) 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

Proper and systematic utilization of every available resource in a software development project is very important issue. The proposed approach introduces a method for the efficient evaluation and selection of human resources in software development projects. The approach contributes and objectively supports project managers in the difficult task of selecting and allocating human resources in software development tasks.

Using subjective evaluations on skills/competencies required for specific task activities, available human resource skills/competencies and relationships between the various skills/competencies – as they are 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 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. By studying the results of the presented example, we can, for example, conclude that while in the initial evaluation of the candidate human resources (Table III, last column) resources r_3 and r_4 are both highly suitable for the task skill requirements, with ratings 3.64 and 3.94 respectively, 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 VI, last column).

V. RELATED WORK

A. Evaluating Human Resources for Software Development Tasks

Human resource management is a critical task that determines organizational success, since the cost of human resources is usually the largest one in a software project [17]. Many approaches have been proposed in the literature to support the evaluation and allocation of personnel in software development projects, using different techniques, such as simulation, genetic algorithms and fuzzy theory, aiming to overcome difficulties that force project managers (decision makers) to assign resources subjectively to development tasks.

Barreto et al. in [2] address project staffing as a constraint satisfaction problem [18], based on utility functions that should be maximized or minimized by the selected development team in order to provide greater value for a software development organization. Their approach takes into account the characteristics of the activities of the project, the available human resources, and the constraints established by the management of a software development organization. However, their approach does not consider differences in capability levels according to the required skills of different development tasks.

Otero et al. in [6] present a methodology called Best-Fitted Resource (BFR) which aims to assign staff to tasks for which someone with a set of first-class skills is not available. The proposed methodology is a systematic approach to determining the fit between a candidate's set of skills and the skills required for the tasks, while also incorporating relationship-ability tables to describe how prior knowledge in various skills contributes to the learning of other skills. This approach, although similar with the presented one in the current paper, does not take into account fuzziness and vagueness issues in characterizing capabilities of candidates in required skills and the levels of expertise required.

Silva and Costa in [5] present a framework based on dynamic programming with a methodology that determines the fit between the complete set of skills available from a candidate member of a project team and the skills required for that project so as to assist the process of allocating human resources in IS projects. They identify interpersonality factors of human resource allocation to information system projects management and judge the personality factors on the capability for project development. However, they also fail to address vagueness in skill evaluations and the relationship between the various skills that contributes to the learning of other skills.

Ruskova in [19] presents a fuzzy logic-based system for human resource selection and evaluation. In particular, a model is suggested in [19] that is divided into three parts. The first one describes the evaluation of job positions, where requirements for each job position are defined. The second part deals with candidates' appraisal, conveying a description for each candidate (characteristics, competences possessed, among others). The third part of the model describes how the former two parts are combined to suggest a list of candidate professionals that better fulfill the requirements of available job positions. A tool that implements the model is also presented. The model suggested by Ruskova [19] is also based on fuzzy logic as the one presented in the current paper; however, this model does not address the fact that skills/competencies in software development are often not independent of each other.

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 representative approach is the one suggested in [3]. However, the main assumption of the approach 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 judgments expressed by decision makers/project managers.

B. Fuzzy Linguistic Methods

A fuzzy linguistic description approach performs well when cases are described based on qualitative aspects instead of quantitative aspects, and/or when the information is vague or imprecisely described. Herrera and Martinez [10] proposed a 2-tuple fuzzy linguistic approach which is a continuous model that carries out the "computing with words" paradigm, namely it processes qualitative linguistic descriptions without losing information. Their model has been applied to several application areas [20]. For example, in [21] Li et al. used the fuzzy linguistic model to recommend relevant human experts in a knowledge management system, while Porcel et al. in [22] applied the proposed model to assist researchers in obtaining information from various research sources. Moreover, Gerogiannis et al. in [23] used 2-tuple fuzzy linguistic terms to provide appropriate recommendations of high involvement products, such as furniture, to end-users.

VI. CONCLUSION

In this paper we presented a fuzzy linguistic description approach based on 2- tuple fuzzy linguistic terms to evaluate human resources involved in software development tasks. The proposed approach 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, we calculate skill relationships 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.

Our future work aims to further improve the suggested approach by determining resource teams based on substitution and complementarity associations between candidate human resources. Another important issue to be addressed is 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 – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - in the context of the research project "SPRINT SMEs", Act: "ARCHIMEDES III" in the Operational Program "Education and Lifelong Learning 2007-13". Investing in knowledge society through the European Social Fund.

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] M. André, M. G. Baldoquín, and S. T. Acuña, "Formal model for assigning human resources to teams in software projects," *Information and Software Technology,* vol. 53, pp. 259-275, 2011.

[5] 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.

[6] 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.

[7] R. Kolisch, "Resource allocation capabilities of commercial project management software packages," *Interfaces,* vol. 29, pp. 19-31, 1999.

[8] L. D. Otero and C. E. Otero, "A fuzzy expert system architecture for capability assessments in skill-based environments," *Expert Systems with Applications,* vol. 39, pp. 654-662, 2012.

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

[10] 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.

[11] 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.

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

[13] S. Bibi, V. C. Gerogiannis, G. Kakarontzas, and I. Stamelos, "Ontology based Bayesian Software Process Improvenent," To be published in *Proceedings of the 9th International Joint conference on Software Technologies (ICSOFT 2014), 2014.*

[14] 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.

[15] 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.

[16] 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.

[17] O. Zwikael and E. Unger-Aviram, "HRM in project groups: the effect of project duration on team development effectiveness," *International Journal of Project Management,* vol. 28, pp. 413-421, 2010.

[18] W. W. Chu and P. Ngai, "A dynamic constraint-directed ordered search algorithm for solving constraint satisfaction problems," in *Proceedings of the 1st international conference on Industrial and engineering applications of artificial intelligence and expert systems-Volume 1*, 1988, pp. 116-125.

[19] 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.

[20] F. Herrera, "An overview on the 2-tuple linguistic model for computing with words in decision making: extensions, applications and challenges," *Information Sciences,* vol. 207, pp. 1-18, 2012.

[21] M. Li, L. Liu, and C.-B. Li, "An approach to expert recommendation based on fuzzy linguistic method and fuzzy text classification in knowledge management systems," *Expert Systems with Applications,* vol. 38, pp. 8586-8596, 2011.

[22] C. Porcel, A. G. López-Herrera, and E. Herrera-Viedma, "A recommender system for research resources based on fuzzy linguistic modeling," *Expert Systems with Applications,* vol. 36, pp. 5173-5183, 2009.

[23] V. C. Gerogiannis, A. Karageorgos, L. Liu, and C. Tjortjis, "Personalised Fuzzy Recommendation for High Involvement Products," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, 2013, pp. 4884-4890.

BIOGRAPHY

Vassilis C. Gerogiannis holds a Diploma and a PhD in Software Engineering from the University of Patras, Greece. He is an Associate Professor at the Business Administration Dept. of the Technological Education Institute of Thessaly, Greece. He is also an adjunct Professor at the Hellenic Open University. He has published a number of papers in international journals and conference proceedings. His research interests include software engineering and project management. He is currently the coordinator of the SPRINT SMEs project, an R&D project focusing on the improvement of software development processes in Small & Medium-sized SW Enterprises. Gerogiannis has been involved as project manager in a number of EU and national R&D projects.

Elli Rapti is a PhD candidate at the Dept. of Electrical and Computer Engineering, University of Thessaly, Greece. She received her Diploma degree in Electrical and Computer Engineering from the Democritus University of Thrace, Xanthi, Greece, in 2010, and her Master's degree in Science and Technology of Computers and Telecommunications from the University of Thessaly, Greece, in 2014. She is currently a research assistant at the Institute for Research and Technology, Thessaly, Greece specializing in the area of distributed intelligent systems.

Anthony Karageorgos holds a BS. in Applied Mathematics, an M.Sc in Computer Science and a PhD in Agent-based Software Engineering. He has participated in numerous research projects in Greece and UK, which resulted in patents and several international journal and conference publications. Since 2014 he is an Assistant Professor with the Dept. of Wood and Furniture Design and Technology, TEI of Thessaly, Greece. Karageorgos is an Associate Editor of the ACM Transactions on Autonomous Adaptive Systems. His research interests include Intelligent Software, Self-Organising and Service-Oriented Systems, Multi-Agent Applications, and Simulation Modelling. Dr Karageorgos is a member of IEEE Computer Society and ACM.

Panos Fitsilis is full Professor at TEI of Thessaly, Dept. of Business Administration, Greece. Fitsilis has extensive project management experience with the development and deployment of large IT systems and extensive experience in various senior management positions such as Director of School of Business and Economics, Vice President of TEI Thessaly Research Committee, Head of Project Management Dept. at TEI Thessaly, Head of Business Unit at international IT company, etc. He is author of three books and many articles published in scientific journals and international conferences. His research interests include Project Management, Software Engineering and Business Process Reengineering.

TABLE I. PROJECT MANAGER EVALUATIONS ON SKILLS/COMPETENCIES (x_{ij}/r_{ij})

Activity (x_i)/ Resource (r_i)	**Project Manager (e_k)/Weight of Importance (ε_k)**	**Levels of Skills Required for a Task / Skills Available from Candidate Resources (c_j)**			
		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_{ij} AND TASK PROFILE tp_j

Activity (x_i)	Importance Degree (I_i)	Required Skills (c_j)				Task Profile (tp_j)
		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_{ij} AND RESOURCE CAPABILITIES ASSESSMENT cs_i

Resource (r_i)	Task Profile	Required Skills (c_j)				Resource Capabilities (cs_i)
		OO design (c_1)	*C++* (c_2)	*VB* (c_3)	*Java* (c_4)	
r_1	H(4, −0.13)	1.31/(1, 0.31)	1/(1, 0)	4.69/(5, −0.31)	3.25/(3, 0.25)	2.07/L(2, 0.07)
r_2	H(4, 0.14)	0.69/(1, −0.31)	1/(1, 0)	4.69/(5, −0.31)	2/(2, 0)	1.73/L(2, −0.27)
r_3	L(2, 0.24)	3/(3, 0)	5.31/(5, 0.31)	3/(3, 0)	1.31/(1, 0.31)	3.64/H(4, −0.36)
r_4	VL(1, 0.34)	5.31/(5, 0.31)	5.31/(5, 0.31)	1/(1, 0)	0.69/(1, −0.31)	3.94/H(4, −0.06)

TABLE IV. PROJECT MANAGER EVALUATIONS ON SKILL/COMPETENCY RELATIONSHIPS c_{ij}

Required Skills/ Competencies (c_j)	Project Manager (e_k)/Weight of Importance (ε_k)	Required Skills (c_j)			
		OO design (c_1)	*C++* (c_2)	*VB* (c_3)	*Java* (c_4)
OO design	$e_1/(1/3)$	--	$VH(5,0)$	$M(3,0)$	$VH(5,0)$
	$e_2/(1/3)$	--	$H(4,0)$	$L(2,0)$	$VH(5,0)$
	$e_3/(1/3)$	--	$VVH(6,0)$	$M(3,0)$	$VH(5,0)$
C++	$e_1/(1/3)$	$VVH(6,0)$	--	$L(2,0)$	$VH(5,0)$
	$e_2/(1/3)$	$H(4,0)$	--	$M(3,0)$	$H(4,0)$
	$e_3/(1/3)$	$VVH(6,0)$	--	$M(3,0)$	$VH(5,0)$
VB	$e_1/(1/3)$	$L(2,0)$	$M(3,0)$	--	$L(2,0)$
	$e_2/(1/3)$	$L(2,0)$	$L(2,0)$	--	$VL(1,0)$
	$e_3/(1/3)$	$VL(1,0)$	$M(3,0)$	--	$VL(1,0)$
Java	$e_1/(1/3)$	$H(4,0)$	$VH(5,0)$	$VL(1,0)$	--
	$e_2/(1/3)$	$H(4,0)$	$H(4,0)$	$M(3,0)$	--
	$e_3/(1/3)$	$VVH(6,0)$	$H(4,0)$	$M(3,0)$	--

TABLE V. FINAL AGGREGATED RATINGS OF SKILL/COMPETENCY RELATIONSHIPS FSR_{ij}

Skills/ Competencies (c_i)	Task Profile (tp_j)	Skills (c_j)			
		OO design (c_1)	*C++* (c_2)	*VB* (c_3)	*Java* (c_4)
OO design	H(4, −0.13)	--	5/VH(5, 0)	2.69/M(3, −0.31)	5/VH(5, 0)
C++	H(4, 0.14)	5.43/VH(5, 0.43)	--	2.69/M(3, −0.31)	4.69/VH(5, −0.31)
VB	L(2, 0.24)	1.69/L(2, −0.31)	2.69/M(3, −0.31)	--	1.69/L(2, −0.31)
Java	VL(1, 0.34)	4.57/VH(5, −0.43)	4.31/H(4, 0.31)	2.43/L(2, 0.43)	--

TABLE VI. RE-EVALUATED FINAL AGGREGATED RATINGS OF RESOURCES FR_{ij}^{NEW} AND RESOURCE CAPABILITIES cs_i RE-EVALUATION

Resource (r_i)	Task Profile (tp_j)	Required Skills (c_j)				Resource Capabilities (cs_i) (re-evaluated)
		OO design (c_1)	*C++* (c_2)	*VB* (c_3)	*Java* (c_4)	
r_1	H(4, −0.13)	2.41/(2, 0.41)	2.76/(3, −0.24)	4.69/(5, −0.31)	3.25/(3, 0.25)	3.08/M(3, 0.08)
r_2	H(4, 0.14)	1.92/(2, −0.08)	2.06/(2, 0.06)	4.69/(5, −0.31)	2/(2, 0)	2.51/M(3, −0.49)
r_3	L(2, 0.24)	3.41/(3, 0.41)	5.31/(5, 0.31)	3.27/(3, 0.27)	3.95/(4, −0.05)	4.13/H(4, 0.13)
r_4	VL(1, 0.34)	5.31/(5, 0.31)	5.31/(5, 0.31)	3.87/(4, −0.13)	4.67/(5, −0.33)	4.96/VH(5, −0.04)