

Research Article

On Using Fuzzy Linguistic 2-Tuples for the Evaluation of Human Resource Suitability in Software Development Tasks

Vassilis C. Gerogiannis,¹ Elli Rapti,² Anthony Karageorgos,³ and Panos Fitsilis¹

¹Department of Business Administration, Project Management Direction of Studies,
Technological Educational Institute of Thessaly, 41110 Larissa, Greece

²Centre for Research and Technology, Institute of Research and Technology of Thessaly, 38333 Volos, Greece

³Department of Wood and Furniture Design and Technology, Technological Educational Institute of Thessaly, 43100 Karditsa, Greece

Correspondence should be addressed to Vassilis C. Gerogiannis; gerogian@teilar.gr

Received 24 March 2015; Accepted 9 June 2015

Academic Editor: Phillip A. Laplante

Copyright © 2015 Vassilis C. Gerogiannis et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Efficient allocation of human resources to the development tasks comprising a software project is a key challenge in software project management. To address this critical issue, a systematic human resource evaluation and selection approach can be proven helpful. In this paper, a fuzzy linguistic approach is introduced to evaluate the suitability of candidate human resources (software developers) considering their technical skills (i.e., provided skills) and the technical skills required to perform a software development task (i.e., task-related skills). The proposed approach is based on qualitative evaluations which are derived in the form of fuzzy linguistic 2-tuples from a group of decision makers (project managers). The approach applies a group/similarity degree-based aggregation technique to obtain an objective aggregation of the ratings of task-related skills and provided skills. To further analyse the suitability of each candidate developer, possible skill relationships are considered, which reflect the contribution of provided skills to the capability of learning other skills. The applicability of the approach is demonstrated and discussed through an exemplar case study scenario.

1. Introduction

The problem of finding the “best” human resource in software projects is not always related to the optimal decision [1, 2]. For example, if a software programming task demands “high skill level for Java” and “low skill level for C++” then both of these requirements should be addressed by considering their respective levels of required knowledge. Thus, a human resource (i.e., a programmer in such a case) who meets both criteria is considered to be more suitable for the specific task, as opposed to another programmer with very high level skills in both Java and C++, since this resource could be assigned to a more demanding programming task. Moreover, human resource evaluation and allocation on software development projects are often based on experience, heuristic knowledge, and subjective judgments of experts (e.g., software team leaders and project managers). Although their judgments can be educated guesses that work in practice, managers lack appropriate tools to assess personnel and perform human resource

allocation [3, 4]. Subjective decisions may result in selecting inadequate and unsuitable resources which require more time for training, a fact that consequently affects negatively the schedule of projects. In skill-based working environments, such as software development projects, where the objective of the development tasks is to provide customized and usually nonrepetitive solutions to end-users/customers, the risk of utilizing inadequate resources is high, which further causes excessive project costs and high probability for developing unsuccessful software products/services [5].

Therefore, a key problem in software project management is to achieve an, as much as possible, objective and documented assessment of knowledge and technical 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 human 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 [6] can be used as an efficient conceptual base, due to the fact that most human reasoning, evaluation, and assessment forms are qualitative and approximate by their nature [1, 2, 5].

In this paper, we use the fuzzy linguistic 2-tuple representation/computation model [7] to build an evaluation and selection approach for candidate human resources (developers) in software development tasks, according to provided/required technical skills/competencies. The proposed approach is based on a group-based fuzzy multicriteria method that applies similarity degree-based aggregation [8] to derive an objective assessment for provided/required skills/competencies. Since technical skills/competencies in software development are not often independent of each other (i.e., prior knowledge in various skills can contribute to learning of other skills) [2], an important contribution of the suggested approach is that it also emphasizes deriving objective assessments of developers matching skills by considering possible skill relationships and dependencies. The approach presented in this paper is an extension and followup of our previous work in human resource evaluation and selection as presented in [9, 10]. Compared to these previous works, in this paper, a more detailed description of the approach is given along with an extended literature review and a presentation of case study with analytical calculations showing the approach applicability. The approach has been developed in the context of the SPRINT SMEs project [11], an R&D project that aims to suggest techniques for software process improvement tailored to the needs of small and medium sized software development organizations.

The remainder of the paper is structured as follows. In Section 2, we provide the background behind our work and an overview on the related literature. In Section 3, we describe the proposed approach for the evaluation and selection of developers in software development tasks. In Section 4, we discuss how the outputs of the approach (estimations of resource-task suitability values) can be used as an aid to indicate proper allocations of each developer effort to each development task. In Section 5, we present an exemplar case study scenario with arithmetic results, followed by a brief discussion in Section 6 upon the usefulness of the provided results. Finally, in Section 7, we conclude the paper and present our future research plans.

2. Background and Literature Overview

Software process improvement (SPI) in small medium sized enterprises (SMEs) is gaining momentum in software research and industry. SPI is a challenging endeavour for most software SMEs aiming at preventing project failures and delivering high quality software products/services consistent with end-customer needs. The SPRINT SMEs project (research in software process improvement methodologies for small and medium sized software development enterprises) [11] is an ongoing R&D project in which the authors

of the paper are involved with the objective to propose and develop a practical framework of methods for supporting the improvement of certain processes which take place in software SMEs. The framework of SPRINT SMEs methods focuses on suggesting rigorous decision making methods for supporting process assessment and improvement problems met by software SMEs. In particular, SPRINT SMEs methods concentrate on problems in selected process domains such as requirements engineering and prioritization and project planning and staffing [12, 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 [1, 2, 14]. While midsize to large companies are characterized by a large pool with a considerable number of available resources with different skills to choose from, small to midsize companies need to follow a tailored decision making process to effectively utilize the limited human resources available. The aim of SPRINT SMEs project is to take into consideration the particular needs of SMEs specialising in software development and propose easily applied and systematic approaches to help improve the domain of software project staffing processes. In the following, we present an overview of the research state of the art on resource evaluation, selection, and allocation in software projects focusing particularly on skill-based and fuzzy-based approaches. More interested readers are referred to [15] where a comprehensive literature review is presented on team composition criteria and team management approaches in software projects.

The problem of human resource allocation in a software project refers to the proper assignment of available human resources to the project tasks. The process usually followed by software development organizations concerns initially the division of project development effort into tasks, each one requiring specific skills, capabilities, competences, and previous experience. Once the various tasks to be performed have been defined, suitable candidates for each task need to be selected mainly according to skill-related requirements [1, 2, 16, 17]. However, assignment of roles and consequent formation of project teams rely often on project managers' subjective judgments and experience, project constraints (e.g., project budget and timing constraints), resource constraints (e.g., availability of human resources), and skill-related requirements. All this information is not usually systematically recorded and handled [14]. Proper resource evaluation and selection of human resources is an important decision in software project management. This decision is not just crucial for the generation of efficient software teams, but it is also of strategic importance in software development organizations since it can support them for developing a competitive advantage by exploiting long term competences of available personnel and implement successful projects [18]. Many approaches discussed in the literature, regarding resource selection and allocation, handle resources in a quantitative manner, where all resources are treated as equally skilled [1]. However, in human and skill-centered working environments, such as software development projects, personnel selection and assignment decisions considering individual level of skills of each available human resource

can clearly contribute to project success. For example, one expected benefit derived from improvements in a resource evaluation and allocation process is the decrease in project duration, an issue which further increases the organization productivity [2, 19].

Despite all research and advances in this research field, managing personnel in software projects still remains a very complicated problem due to the dynamic and complex context in which it takes place. Project managers/decision makers have to face a number of different possible staff combinations and many correlated and often conflicting decision variables, such as time, cost, and quality, and, respectively, make the best decision under pressure and high demands [16]. In addition, a major contributor to this complexity is the increased demand for specialized individual skills in the software development workforce, which results from high turnover rates and the fast pace at which new technologies and techniques are being developed. As a result of high demands, candidates with exact required skills to be allocated at software development tasks are usually not available [2].

Human resource evaluation is one critical task that determines to a large extent the organizational success, since the cost of human resources is usually the largest one in a software project [20]. The ability to properly evaluate the skill levels of available human resources is vital to achieve proper resource allocation in the development tasks comprising a software project. Besides skill-based assessment approaches, in the literature, there are also research works [3, 14] focusing on evaluations of various personality traits and different types of behavioural capabilities of candidate human resources, such as intrapersonal, organizational, interpersonal, and management capabilities. The main aim of these approaches is to properly assign available resources to project functional roles (e.g., project leader, analyst, designer, programmer, tester, etc.) and rather not to allocate resources to specific project development tasks, a decision that requires considering explicitly the capabilities of available resources regarding specific technical skills (e.g., level of knowledge in a particular programming language, expertise in applying an analysis/design methodology, competence in using a specific CASE tool, etc.). A central problem in software projects, in particular regarding the human resource allocation, is to evaluate accurately and objectively the capability of each developer/engineer with respect to required technical skills [5]. In the following, we review some representative approaches presented in the literature that attempt to manage the vague and imprecise problem of performing skill-based assessments to support resource allocation decisions in software projects.

For example, Tsai et al. in [21] proposed a method for selecting the most suitable set of human resources with the aim of completing a software project at minimum cost and duration. Their method adopts the critical resource diagram technique which focuses on resource, rather than activity scheduling, to represent task workflows and precedence relations. In addition, their method applies Taguchi's parameter design approach to optimise the selection of software engineers for tasks under dynamic/stochastic conditions. This approach considers technical skills (estimated as average numbers of lines of code per day) and tasks as controllable

and uncontrollable factors, respectively, and Taguchi's parameter design is performed to select personnel who achieved high performance. However, this method cannot be easily applied by immature software SMEs since it requires detailed and systematic recording of personnel productivity (i.e., to keep records of average numbers of lines of code produced per day by each software engineer) and estimations of optimistic, normal, and pessimistic values of resource personal abilities for each task type.

Otero et al. in [2] presented a promising methodology, termed Best-Fitted Resource (BFR) approach, where human resource capabilities are evaluated by considering relationships between skills. According to BFR, skill-relationship tables are formed with the aim of describing how prior knowledge in various skills contributes to learning of other skills. BFR calculates the suitability of each resource with each task and, consequently, the allocation of resources to multiple tasks is solved as a simple linear programming problem with the aim of maximizing the overall resource suitability. While the consideration of required skills, provided skills, and skill relationships makes the approach similar to the one presented in this paper, BFR does not take into account vagueness issues in characterizing capabilities of candidate human resources and required levels of expertise and it does not result in a group-based objective assessment for the resources suitability. In the approach presented in this paper, objective skill assessments can be derived by aggregating evaluations from decision makers in a group decision making setting.

In practice, human resource evaluation and allocation are usually based on the experience, intuition, and expert judgments of project managers. Software development projects tend to become more complex in time requiring more diversified skills to carry them out and multiple constraints to be satisfied. These complexities raise the need for the introduction of advanced computation methods to assist decision makers to achieve best possible allocation of the available human resources [22]. To this end, a number of analytical approaches are proposed in the literature which apply powerful optimisation techniques, such as constraint satisfaction [17, 23] and dynamic programming algorithms [18].

For example, Barreto et al. in [17] address project staffing as a constraint satisfaction problem [24], 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 as input the needs of the development tasks, the available human resources, and the constraints imposed by project managers. However, their approach does not consider differences in capability levels according to the required skills of different development tasks. Another constraint-based approach is the one proposed by Kang et al. in [23]. They identify human resources in software projects by individual and team-level constraints based both on the available literature and experts' judgment. Their approach estimates productivity of human resources by using the COCOMO II algorithmic effort estimation model and focuses on scheduling optimisation of human resource allocations by applying accelerated simulated annealing. However, the major limitation of

constraint-based approaches for software project staffing is their scalability, since these approaches become difficult to be implemented as the number of constraints increases. e Silva and Costa in [18] present another interesting optimisation approach based on dynamic programming. Their method determines the fit between the complete set of skills available from a candidate member of a project team and the skills required from project resource requirements. Besides skills, they also identify interpersonality factors of human resource allocation to project management and judge the personality factors on the capability for project development. However, they also fail to address vagueness in skill evaluations and relationships between various skills that contribute to learning other skills.

Since human decision related to personnel evaluation and selection can be characterised as qualitative and highly imprecise, fuzzy logic techniques have been proposed as feasible methods to handle these issues [25, 26]. Fuzzy logic was first introduced by Zadeh [6] and has been widely recognized as a suitable tool to describe vague personnel capabilities as well as imprecise project requirements [27]. There are some interesting approaches in the literature which address human resource evaluation and selection in software projects using fuzzy logic. For example, Ruskova in [28] proposed a fuzzy logic-based model for human resource selection and evaluation which is divided into three modules. The first one describes the evaluation of job positions, where requirements for each job position are defined. The second module deals with candidates' appraisal, conveying a description for each candidate that includes candidate characteristics and competences possessed among others. The third module of the proposed system describes how the former two modules are combined to suggest a list of candidate professionals that better fulfil the requirements of available job positions. A tool that implements the model is also presented. While this model is also based on fuzzy logic as the one presented in the current paper, it does not explicitly address the fact that skills/competencies in software development are often not independent of each other.

Dodangeh et al. in [29] also developed a decision making methodology that integrates group decision making and fuzzy linguistic evaluation. Decision makers are able to evaluate candidates using linguistic variables. The proposed methodology applies aggregation for subjective evaluations provided by the decision makers with the aim of performing more robust human resource selection procedures. However, the authors do not consider skill relationships reflecting how prior knowledge in various skills can contribute to the learning of other skills.

Callegari and Bastos in [1] proposed a fuzzy rule-based approach for human resource evaluation in software projects while also considering dependencies between skills to extract objective evaluations of human resources. Their main assumption 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 approach presented in this

paper is a group-based one that emphasizes deriving objective values for skill relationships and required/provided skill evaluations from corresponding subjective judgments expressed by decision makers/project managers. Another fuzzy rule-based approach is proposed in [5] by L. D. Otero and C. E. Otero. The authors propose a fuzzy rule-based expert system where personnel capabilities are modelled as fuzzy sets, relationships between skills are described as fuzzy rules, and capability assessments are computed by applying the Mamdani max-min method of fuzzy reasoning. However, the rules describing the skill relationships in the expert system knowledge base were developed in a rather ad hoc way by conducting experts/decision makers. This approach requires implementation and use of an expert system which cannot be easily adapted to the different skill-related requirements present in each development project that undertakes a software SME.

The underlying representation/computation of the approach presented in the current paper is the fuzzy 2-tuple linguistic model, presented by Herrera and Martínez [7]. Their model is based on the concept of symbolic translation and it is used to represent linguistic assessments in the form of 2-tuples composed of a linguistic term and a number. The main advantage of the fuzzy 2-tuple linguistic model is its continuity which overcomes the issue of information loss.

The fuzzy 2-tuple linguistic model has been compared with other symbolic linguistic computing models by Rodriguez and Martinez in [30]. It has been used in various application domains and in a number of approaches as a basis for experts to express their opinions [31]. Da and Xu in [32] introduced a proportional 2-tuple linguistic representation model and applied a group decision making/evaluation procedure in the context of complex construction projects. Dursun and Karsak in [33] proposed a fuzzy multicriteria decision making algorithm for personnel selection using the principles of fusion of fuzzy information, 2-tuple linguistic representation model, and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). Raoudha et al. in [34] used the fuzzy 2-tuple linguistic representation model to determine appraisers' opinions about both acquired and required competence levels. Given the resulting 2-tuple assessments, they suggested an extended version of the TOPSIS method to get the candidates' ranking based on a similarity degree between required and acquired competence levels. In addition, Espinilla et al. in [35] presented an integrated model for a group-based, 360-degree performance appraisal process that manages heterogeneous assessments and computes a final linguistic evaluation for each employee. Their model applies effective aggregation by considering the interaction among criteria and reviewers weights. Assessment values are expressed in linguistic 2-tuples to deal with heterogeneous information and obtain linguistic results close to human cognitive model. However, while all of the above approaches are computationally practical, they do not explicitly consider skill relationships or the subjectivity in experts' opinion. Moreover, the use of the ordered weighted averaging (OWA) operator [33, 35] and the Choquet integral [35] increases the computation complexity of the respective approaches.

3. Description of the Proposed Approach

In software projects, the resource selection process, where the resources considered are developers with a set of technical skills, typically involves decisions based on multiple criteria. While the various choices are often distinct for each manager, the basis of selecting the most appropriate resources for each project and task is considered to have one common ground: the evaluation of each task's requirements aligned with the skills of the available resources. Thus, the problem is reduced to the proper assignment of resources to tasks, even when the most desirable skills are not available in the resource pool.

To address human resource evaluation and selection in software development tasks, we suggest a group-based fuzzy multicriteria approach using the fuzzy linguistic 2-tuple representation model. The proposed approach consists of 3 different stages, which further include nine distinct steps as presented in the flowchart of Figure 1. Specifically, in stage 1, consisting of Steps 1 to 5, the task profile (with respect to the required skills) is estimated using linguistic evaluations provided by experts on the skills required for the particular task. In stage 2, which includes Steps 6 to 7, the capabilities of the available human resources are evaluated using linguistic evaluations provided by experts on their skills. Last, in stage 3, consisting of Steps 8 to 9, the capabilities of the available human resources are reevaluated using linguistic evaluations of skill relationships provided also by experts/decision makers. Steps 2 to 4 are common to the three different stages of the algorithm to ensure the objectivity of the experts' evaluations. The nine steps of the approach are described in detail in the following subsections.

Step 1 (group-based linguistic evaluation of required skills). Let T be a software development task in a software project. We assume that it is planned to be executed as a set of N individual development activities x ; that is, $T = \{x_1, x_2, \dots, x_N\}$. In total, the task activities require a set of M technical skills/competencies c , such as knowledge in Java and/or C++, denoted as $C = \{c_j \mid j = 1, 2, \dots, M\}$. The proposed approach applies a group-based decision approach by requiring from K project managers/evaluators e , $E = \{e_k \mid k = 1, 2, \dots, K\}$, to evaluate the level of each skill/competency required for each individual activity to be completed successfully. To express skill/competency requirements in a qualitative form we adopt the 2-tuple fuzzy linguistic terms approach [7]. 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/defuzzification of available information [7, 36].

To express their evaluations, project managers use a linguistic label set $S = \{s_i \mid i = 0, 1, 2, \dots, g\}$, where $g + 1$ is the granularity of the selected linguistic term set. A 2-tuple linguistic variable is thus denoted as (s_i, a_i) , where $s_i \in S$ corresponds to the central value of the i th linguistic term in the term set and $a_i \in [-0.5, 0.5]$ is the distance from s_i . Project managers may also select different linguistic term sets (i.e., sets having different granularities or

semantics) to express their assessments for the required skills. In this case, all heterogeneous linguistic assessments have to be transformed and expressed into a uniform linguistic term set by following the method proposed in [37, 38]. 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 e_k .

Step 2 (transformation of linguistic evaluations). Since project managers evaluate each development 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 the skill/competency c_j ($j = 1, 2, \dots, M$) is transformed into a 2-tuple of the form $(s_i, 0)$, according to the following transformation function [7]:

$$\theta : S \longrightarrow S \times [-0.5, 0.5], \quad \theta(s_i) = (s_i, 0), \quad s_i \in S. \quad (1)$$

Step 3 (similarity degree-based aggregation). By performing group-based linguistic evaluation, all skills required to perform the software development task are characterized by the subjective judgments of project managers, which may be biased towards each required skill/competency. Thus, in order to derive a more objective assessment, we adopt a similarity degree-based aggregation technique similar to the one introduced in [8]. The final aggregated assessments consider not only the relative importance levels-weights ε_k of project managers, but also similarities in their respective evaluations. Therefore, we expect the resulting assessments to reflect the collective judgments of project managers more reasonably and more objectively. The aggregation technique follows three distinct steps that are, respectively, presented in the following subsections.

3.1. Similarity Degree Calculation. The similarity degree, denoted as $\text{sim}(X_{ij}^k, X_{ij}^l) \in (0, 1]$, is calculated for any two skill evaluations X_{ij} provided by any two managers e_k and e_l ($k \neq l$, $k, l = 1, 2, \dots, K$) 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 δ between X_{ij}^k and X_{ij}^l evaluations is computed. To calculate this distance, the evaluations represented in 2-tuple linguistic form should be transformed into an equivalent numerical form. 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 $i = \text{round}(\beta)$ 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 [7]:

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

$$\Delta(\beta) = (s_i, a) = \begin{cases} s_i, & i = \text{round}(\beta) \\ \alpha = \beta - i, & a \in [-0.5, 0.5]. \end{cases} \quad (3)$$

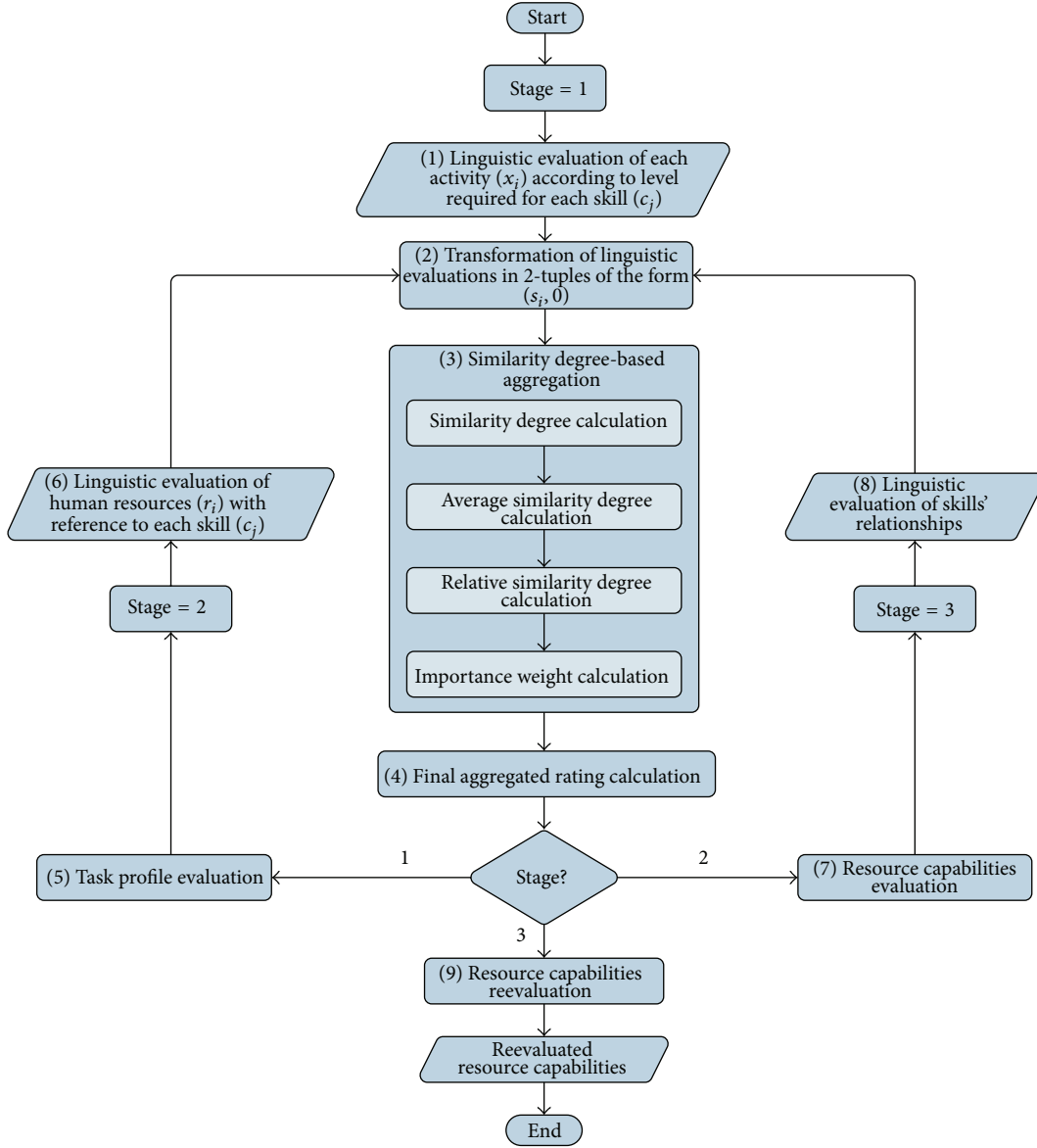


FIGURE 1: Flowchart of the proposed approach for the evaluation of human resources considering skill relationships.

Therefore, a 2-tuple linguistic variable is transformed into an equivalent number $\beta \in [0, g]$ by using the reverse function Δ^{-1} as follows [7]:

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

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

Then, the distance δ between X_{ij}^k and X_{ij}^l evaluations can be calculated using the following formula:

$$\delta = \left| \Delta^{-1}(X_{ij}^k) - \Delta^{-1}(X_{ij}^l) \right|, \quad (6)$$

where Δ^{-1} is the reverse function that transforms a 2-tuple linguistic variable into its equivalent numerical value [7].

Finally, the similarity degree value $\text{sim}(X_{ij}^k, X_{ij}^l)$ is computed according to the following [8]:

$$\text{sim}(x_{ij}^k, x_{ij}^l) = 1 - \left| \frac{\Delta^{-1}(x_{ij}^k) - \Delta^{-1}(x_{ij}^l)}{g} \right|, \quad (7)$$

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

3.2. Average and Relative Similarity Degree Calculation. The average similarity degree $\text{SM}_{ij}(e_k)$ and the relative similarity degree $\text{RSM}_{ij}(e_k)$ are calculated for each project manager 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 [8]

$$SM_{ij}(e_k) = \frac{\sum_{l=1, l \neq k}^K \text{sim}(x_{ij}^k, x_{ij}^l)}{K-1}, \quad (8)$$

$$RSM_{ij}(e_k) = \frac{SM_{ij}(e_k)}{\sum_{l=1}^K SM_{ij}(e_l)}. \quad (9)$$

3.3. 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 [8]:

$$w_{ij}^k = \frac{\varepsilon_k \times RSM_{ij}(e_k)}{\sum_{l=1}^K (\varepsilon_l \times RSM_{ij}(e_l))}. \quad (10)$$

Step 4 (calculation of final aggregated ratings). The objective aggregation for all activities' ratings is computed by utilizing the weighted average operator, as it is defined for fuzzy linguistic 2-tuples in [7]. 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 [7]:

$$\begin{aligned} \bar{x} &= \Delta \left(\frac{\sum_{i=1}^n (\Delta^{-1}(s_i, a_i) \times w_i)}{\sum_{i=1}^n w_i} \right) \\ &= \Delta \left(\frac{\sum_{i=1}^n (\beta_i \times w_i)}{\sum_{i=1}^n w_i} \right), \end{aligned} \quad (11)$$

where β_i is calculated using the reverse function Δ^{-1} as defined 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 (as it is given by formula (11)) on the linguistic evaluations of the activities and using as weights the previously calculated importance weights for the respective assessments. Thus, the final aggregated rating FX_{ij} is calculated as follows:

$$FX_{ij} = \Delta \left(\frac{\sum_{l=1}^K (\Delta^{-1}(X_{ij}^l, a_{ij}^l) \times w_{ij}^l)}{\sum_{l=1}^K w_{ij}^l} \right). \quad (12)$$

Step 5 (task profile evaluation). As stated above, a software development task is composed of a number of activities which result in the 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 desired level of skills/competencies required for the task's successful implementation, based on the project managers' evaluations. The task profile vector, denoted as tp , with respect to each required skill/competency $c \in C$, can be calculated by applying the weighted average operator (11)

to the previously calculated final aggregated ratings of skills (FX), using as weights the importance degree of each development activity $x \in T$ comprising the software development task. The importance degree I_i of each development activity x_i , $i = 1, 2, \dots, N$, may reflect the business value-criticality of the corresponding software component that is expected to result 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 business value compared to the development of the data base component. The task profile vector is calculated as follows:

$$tp_j = \Delta \left(\frac{\sum_{i=1}^N (\Delta^{-1}(FX_{ij}) \times I_i)}{\sum_{i=1}^N I_i} \right), \quad (13)$$

where N is the total number of activities comprising the development task.

Step 6 (linguistic evaluation of candidate human resources). 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 with respect to the task required skills C , Steps 2 to 4 are repeated. Specifically, each project manager evaluates, using a linguistic label set S , each candidate human resource r according to his/her previous experience and/or his/her level of knowledge in each required skill/competency $c \in C$. The linguistic evaluations of each resource according to each required skill are then transformed into 2-tuples in the form $(s_i, 0)$ according to (1). To derive an objective evaluation for these evaluations, the similarity degree value between any two project managers' evaluations is calculated using formula (7). The importance weights are calculated according to (10), using the average and relative similarity degree, as computed by (8) and (9), respectively. Then, a final aggregated rating FR_{ij} for each resource r_i ($i = 1, 2, \dots, Q$) with respect to each skill c_j ($j = 1, 2, \dots, M$) is calculated by applying the weighted average operator (11) on the linguistic evaluations of the resources using as weights the previously calculated importance weights for the respective assessments.

Step 7 (capabilities evaluation). The capability (i.e., suitability) cs_i of each resource r_i ($i = 1, 2, \dots, Q$) is computed by applying the weighted average operator (11) on the final aggregated ratings FR_{ij} , using as weights the previously calculated task profile assessments for each individual required skill. That is,

$$cs_i = \Delta \left(\frac{\sum_{j=1}^M (\Delta^{-1}(FR_{ij}) \times tp_j)}{\sum_{j=1}^M tp_j} \right), \quad (14)$$

where cs_i represents the capability of resource r_i ($i = 1, 2, \dots, Q$), FR_{ij} is the final aggregated rating for the resource r_i ($i = 1, 2, \dots, Q$) with respect to all skills c_j ($j = 1, 2, \dots, M$), and tp_j is the value in the task profile vector that corresponds to skill c_j ($j = 1, 2, \dots, M$).

Step 8 (linguistic evaluation of skill 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 [2]. 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 capability in one skill/competency contributes to the improvement (via learning and/or practical experience) of another skill/competency. To this end, each manager evaluates subjectively skill relationships and a skill-relationships table is constructed, where relationships between skills/competencies are expressed in linguistic terms using a linguistic label set S . To evaluate objectively the skill/competency relationships with respect to the task required skills c_j ($j = 1, 2, \dots, m$), Steps 2 to 4 are repeated. 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 (7). The importance weight is calculated according to (11), using the average and relative similarity degree calculated by (8) and (9), 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_i and c_j ($i, j = 1, 2, \dots, M$). Specifically, FSR_{ij} is calculated by applying the weighted average operator (11) on the linguistic evaluations of the skills/competencies using as weights the previously calculated importance weights for the respective assessments.

Step 9 (reevaluation of the capability/suitability of candidate human resources). As a last step of the approach, the capabilities of human resources in each required skill/competency need to be reevaluated according to the final aggregated rating of skill/competency relationships, which were calculated in Step 8. A new value FR_{ij}^{NEW} is computed for the capability of each human resource r_i ($i = 1, 2, \dots, Q$) with respect to each skill/competency c_j ($j = 1, 2, \dots, M$). This value is computed as the maximum value between the previously calculated capability in a skill and the weighted average contribution on that skill/competency from other skills/competencies, as follows:

$$FR_{ij}^{NEW} = \max \left(FR_{ij}; \frac{\sum_{h=1, h \neq j}^m (FR_{ih} \times FSR_{hj})}{\sum_{h=1, h \neq j}^m FSR_{hj}} \right). \quad (15)$$

According to formula (15) the capability of a human resource in a specific skill is recomputed only when the contribution of other skills results in a greater value compared to the initial capability that was calculated without considering skill relationships.

Consequently, the reevaluated final aggregated rating cs_i^{NEW} of each human resource r_i ($i = 1, 2, \dots, Q$) with respect to all skills/competencies c_j ($j = 1, 2, \dots, M$) is computed by applying the weighted average operator (11) on the reevaluated linguistic evaluations of the human resources, using as

weights the previously calculated task profile assessments for each individual skill/capability. This final rating reflects the overall suitability of each resource to perform the software development task and it is calculated as follows:

$$cs_i^{NEW} = \Delta \left(\frac{\sum_{j=1}^M (\Delta^{-1}(FR_{ij}^{NEW}) \times tp_j)}{\sum_{j=1}^M tp_j} \right). \quad (16)$$

4. Allocation of Resources to Software Development Tasks

Let us assume that a software company is undertaking R development tasks $\{T_1, T_2, \dots, T_R\}$ and at least one developer should be allocated to each task. Then, the above approach can be applied to calculate resource-task suitability values cs_{ij} for each human resource (developer) r_i ($i = 1, 2, \dots, Q$) to perform each development task T_j ($j = 1, 2, \dots, R$). The use of these resource-task suitability values is helpful to indicate proper allocations of developers to tasks. For example, if one developer is needed for one task, then it might be a good decision to assign in this task the developer with the highest corresponding resource-task suitability. If two developers are needed for one task, then these developers may be the two developers characterized with two highest resource-task suitability values.

In the literature, we can find some rather strict assumptions regarding resource allocation in software projects. For example, it is assumed that all resources cannot be allocated at a single task [18] or a resource may be assigned to at most one task [2]. In a small software development company, however, due to limited number of developers it is often the case that all available developers should be involved in all running development tasks, in order to take full advantage of their competences and deliver fast paced projects. In such a case, the resource-task suitability values are useful to indicate proper allocations of each developer effort to each development task.

An estimation of effort allocation EEA_{ij} (as a percentage % value) for each developer r_i to each task T_j can be calculated using formula (17) that divides the corresponding resource-task suitability value cs_{ij} by the sum of suitability values for all tasks in which the developer should be involved. We do recognize that this simple heuristic can be proven impractical to support complex resource allocation rules and availability constraints. In these situations the resource-task suitability values can be used as variables of a multiobjective optimization model, such as the one presented in [14]

$$EEA_{ij} = \frac{cs_{ij}}{\sum_{j=1}^R cs_{ij}}. \quad (17)$$

5. Exemplar Case Study

Let us consider a scenario where a software development company needs to assign personnel to a software development task T_1 , which is composed of four ($N = 4$) individual development activities x ; that is, $T_1 = \{x_1, x_2, x_3, x_4\}$. Thus,

TABLE 1: Project manager evaluations on skill/competencies required for each activity x .

Activity (x)	Project manager (e)/weight of importance (ε)	Levels of skills required for a task (c)			
		OO design (c_1)	C++ (c_2)	VB (c_3)	Java (c_4)
x_1	$e_1/(1/3)$	(H, 0)	(L, 0)	(VL, 0)	(VVH, 0)
	$e_2/(1/3)$	(VVH, 0)	(VL, 0)	(M, 0)	(VH, 0)
	$e_3/(1/3)$	(VVH, 0)	(VVL, 0)	(M, 0)	(M, 0)
x_2	$e_1/(1/3)$	(VL, 0)	(VL, 0)	(VH, 0)	(VL, 0)
	$e_2/(1/3)$	(VVL, 0)	(VL, 0)	(VVH, 0)	(L, 0)
	$e_3/(1/3)$	(VL, 0)	(L, 0)	(VH, 0)	(L, 0)
x_3	$e_1/(1/3)$	(VH, 0)	(VH, 0)	(L, 0)	(L, 0)
	$e_2/(1/3)$	(VVH, 0)	(M, 0)	(L, 0)	(VL, 0)
	$e_3/(1/3)$	(VVH, 0)	(VVH, 0)	(L, 0)	(VL, 0)
x_4	$e_1/(1/3)$	(L, 0)	(VVH, 0)	(VL, 0)	(VL, 0)
	$e_2/(1/3)$	(H, 0)	(VH, 0)	(VL, 0)	(VL, 0)
	$e_3/(1/3)$	(VH, 0)	(VH, 0)	(L, 0)	(VVL, 0)

TABLE 2: Similarity degree values between project manager evaluations.

	e_1/e_2				e_2/e_3				e_1/e_3			
	c_1	c_2	c_3	c_4	c_1	c_2	c_3	c_4	c_1	c_2	c_3	c_4
x_1	0.666667	0.833333	0.666667	0.833333	0.666667	0.666667	0.666667	0.5	1	0.833333	1	0.666667
x_2	0.833333	1	0.833333	0.833333	1	0.833333	1	0.833333	0.833333	0.833333	0.833333	1
x_3	0.833333	0.666667	1	0.833333	0.833333	0.833333	1	0.833333	1	0.5	1	1
x_4	0.666667	0.833333	1	1	0.5	0.833333	0.833333	0.833333	0.833333	1	0.833333	0.833333

we assume that these four development activities lead, respectively, to the development of four software components of a software product; that is, the company follows a component-based software development paradigm. It is also assumed that three project managers ($K = 3$) are asked to evaluate the needs for four specific skills/competencies ($M = 4$), namely, object oriented design, C++, Visual Basic, and Java programming, which are required to perform the activities of the software development task.

The linguistic term set S used by project managers includes the following terms: $s_0 = \text{VVL}$ (Very Very Low), $s_1 = \text{VL}$ (Very Low), $s_2 = \text{L}$ (Low), $s_3 = \text{M}$ (Medium), $s_4 = \text{H}$ (High), $s_5 = \text{VH}$ (Very High), and $s_6 = \text{VVH}$ (Very Very High). Table 1 presents the project managers' evaluations, in the form of 2-tuples, for the level of skills/competencies required for each one of the four development activities comprising the software development task T_1 . For example, according to project manager e_1 , a "Very Very High" level of competency in Java programming is required for activity x_1 , expressed by the 2-tuple (VVH, 0). As shown in Table 1, 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) involved in the case study 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 1), the similarity degree between these two evaluations, according to formula (7), is

$$\begin{aligned} \text{sim}(X_{21}^1, X_{21}^2) &= 1 - \left| \frac{\Delta^{-1}(X_{21}^1) - \Delta^{-1}(X_{21}^2)}{g} \right| \\ &= 1 - \left| \frac{1 - 0}{6} \right| = 0.833333. \end{aligned} \quad (18)$$

The values of the similarity degrees between any two project managers' evaluations for the level of skills required for each activity are computed in the same manner and they are presented in Table 2.

Having calculated the similarity degrees for each activity with respect to each skill, we proceed to calculate the average and relative similarity degrees. For example, the average similarity degree for activity x_2 with respect to skill c_1 (object oriented design) is calculated by applying formula (8) as follows:

$$\begin{aligned} \text{SM}_{21}(e_1) &= \frac{\text{sim}(X_{21}^1, X_{21}^2) + \text{sim}(X_{21}^1, X_{21}^3)}{K - 1} \\ &= \frac{0.83333 + 1}{3 - 1} = 0.916667. \end{aligned} \quad (19)$$

$\text{SM}_{21}(e_2)$ and $\text{SM}_{21}(e_3)$ are similarly calculated as 0.833333 and 0.916667, respectively. Thus, the relative similarity degree for project manager e_1 according to formula (9) is

$$\begin{aligned} \text{RSM}_{21}(e_1) &= \frac{\text{SM}_{21}(e_1)}{\text{SM}_{21}(e_1) + \text{SM}_{21}(e_2) + \text{SM}_{21}(e_3)} \\ &= \frac{0.916667}{0.916667 + 0.833333 + 0.916667} \\ &= 0.34375. \end{aligned} \quad (20)$$

Applying these calculations we can compute all relative similarity degree values for each project manager (Table 3).

Having assumed equal importance levels for all three project managers (i.e., each ε_k is equal to 1/3, Table 1), we

TABLE 3: Relative similarity degree values for each project manager e .

	e_1				e_2				e_3			
	c_1	c_2	c_3	c_4	c_1	c_2	c_3	c_4	c_1	c_2	c_3	c_4
x_1	0,285714	0,321429	0,285714	0,33333	0,357143	0,357143	0,357143	0,375	0,357143	0,321429	0,357143	0,291667
x_2	0,34375	0,34375	0,34375	0,3125	0,3125	0,34375	0,3125	0,34375	0,34375	0,3125	0,34375	0,34375
x_3	0,3125	0,375	0,33333	0,3125	0,34375	0,291667	0,33333	0,34375	0,34375	0,33333	0,33333	0,34375
x_4	0,291667	0,3125	0,34375	0,34375	0,375	0,34375	0,34375	0,34375	0,33333	0,34375	0,3125	0,3125

TABLE 4: Final aggregated ratings for each activity FX and task profile TP.

Activity (x)	Importance degree (I)	Required skills (c)			
		OO design (c_1)	C++ (c_2)	VB (c_3)	Java (c_4)
x_1	VL (1, 0)	5.4285/(VH, 0.43)	1/(VL, 0)	2.4285/(L, 0.43)	4.75/(VH, -0.25)
x_2	L (2, 0)	0.6875/(VL, -0.31)	1.3125/(VL, 0.31)	5.3125/(VH, 0.31)	1.6875/(L, -0.31)
x_3	M (3, 0)	5.6875/(VH, -0.31)	4.75/(VH, -0.25)	2/(L, 0)	1.3125/(VL, 0.31)
x_4	VVH (6, 0)	3.75/(H, -0.25)	5.3125/(VH, 0.31)	1.3125/(VL, 0.31)	0.6875/(VL, -0.31)
	Task profile (TP)	3.8638/(H, -0.13)	4.1458/(H, 0.14)	2.2440/(L, 0.24)	1.3489/(VL, 0.34)

apply formula (10) to compute the weight of the assessments of each project manager. For example, the weight of the assessment of project manager e_1 for activity x_2 with respect to skill c_1 is computed as follows:

$$\begin{aligned}
 w_{21}^1 &= \frac{\varepsilon_1 \times \text{RSM}_{21}(e_1)}{\varepsilon_1 \times \text{RSM}_{21}(e_1) + \varepsilon_2 \times \text{RSM}_{21}(e_2) + \varepsilon_3 \times \text{RSM}_{21}(e_3)} \\
 &= \frac{1/3 \times 0.34375}{1/3 \times 0.34375 + 1/3 \times 0.3125 + 1/3 \times 0.34375} \\
 &= 0.34375.
 \end{aligned} \quad (21)$$

The weights w_{21}^2 and w_{21}^3 of the assessments of project managers e_2 and e_3 are calculated in the same way and they are found equal to 0.3125 and 0.34375, respectively. It should be noted that assessment weights are found to be equal to the previously computed relative similarity degree values (Table 3) because we have assumed equal importance levels for all three project managers (i.e., each ε_k is equal to 1/3).

We continue by calculating the final aggregated rating for each activity with respect to each skill. For example, for activity x_2 with respect to skill c_1 using formula (12), the final aggregated rating is computed as follows:

$$\begin{aligned}
 \text{FX}_{21} &= \Delta \left(\frac{\Delta^{-1}(X_{21}^1) \times w_{21}^1 + \Delta^{-1}(X_{21}^2) \times w_{21}^2 + \Delta^{-1}(X_{21}^3) \times w_{21}^3}{w_{21}^1 + w_{21}^2 + w_{21}^3} \right) \\
 &= \Delta \left(\frac{1 \times 0.34375 + 0 \times 0.3125 + 1 \times 0.34375}{0.34375 + 0.3125 + 0.34375} \right) = \Delta(0.68675) \\
 &= (1, -0.31) = (\text{VL}, -0.31).
 \end{aligned} \quad (22)$$

The final aggregated ratings for all activities with respect to the various required skills/competencies are shown in Table 4.

Having calculated the final aggregated rating for each activity with respect to each skill, we proceed with the

calculation of the task profile vector. For example, for skill c_1 , namely, object oriented design (Table 4, column 3), the corresponding value in task profile vector is, according to formula (13), equal to

$$\begin{aligned}
 \text{tp}_1 &= \Delta \left(\frac{5.4285 \times 1 + 0.6875 \times 2 + 5.6875 \times 3 + 3.75 \times 6}{1 + 2 + 3 + 6} \right) \\
 &= \Delta(3.8638) = (4, -0.13) = (\text{H}, -0.13).
 \end{aligned} \quad (23)$$

The resulting task profile for all individual required skills is, therefore, calculated as a vector of linguistic 2-tuples and it is shown in the last row of Table 4. From this task profile, we can conclude that, for this specific task, high level of competence in both C++ programming and object oriented design is required (i.e., the corresponding 2-tuples are equal to (H, 0.14) and (H, -0.13), resp.) but a lower level of competency is required in Visual Basic and Java programming (i.e., the corresponding 2-tuples are equal to (L, 0.24) and (VL, 0.34), resp.).

Having calculated the task profile, we continue by evaluating candidate human resources (developers). Table 5 presents the judgments expressed by project managers for the level that characterizes each one of the four available developers ($Q = 4$) with respect to each required skill. For example, according to project manager e_1 , developer r_1 is characterized by a "Very Low" level of competence regarding object oriented design.

Following the same procedure as above (i.e., Steps 2–4 of the approach), we get the final aggregated ratings FR of each candidate resource with respect to each skill, as shown in columns 2–5 of Table 6. The capability/suitability of each resource c_s is calculated using formula (14) and the results are shown in the last column of Table 6.

In the next stage of the approach we perform reevaluation of the resource capabilities. Project managers provide their judgments on skill relationships, that is, how prior knowledge in various skills contributes to the learning of other skills, as

TABLE 5: Project managers' evaluation on skills/competencies of available human resources.

Resource (r)	Project manager (e)/weight of importance (ϵ)	Skills available from candidate resources (c)			
		OO design (c_1)	C++ (c_2)	VB (c_3)	Java (c_4)
r_1	$e_1/(1/3)$	(VL, 0)	(L, 0)	(H, 0)	(VH, 0)
	$e_2/(1/3)$	(L, 0)	(VL, 0)	(VH, 0)	(M, 0)
	$e_3/(1/3)$	(VL, 0)	(VVL, 0)	(VH, 0)	(L, 0)
r_2	$e_1/(1/3)$	(VL, 0)	(VL, 0)	(VH, 0)	(L, 0)
	$e_2/(1/3)$	(VL, 0)	(VL, 0)	(H, 0)	(L, 0)
	$e_3/(1/3)$	(VVL, 0)	(VL, 0)	(VH, 0)	(L, 0)
r_3	$e_1/(1/3)$	(M, 0)	(VH, 0)	(H, 0)	(L, 0)
	$e_2/(1/3)$	(H, 0)	(VH, 0)	(L, 0)	(VL, 0)
	$e_3/(1/3)$	(L, 0)	(VVH, 0)	(M, 0)	(VL, 0)
r_4	$e_1/(1/3)$	(VH, 0)	(VVH, 0)	(VL, 0)	(VL, 0)
	$e_2/(1/3)$	(VVH, 0)	(VH, 0)	(VL, 0)	(VL, 0)
	$e_3/(1/3)$	(VH, 0)	(VH, 0)	(VL, 0)	(VVL, 0)

TABLE 6: Final aggregated ratings for each resource FR and resource capabilities cs.

Resource (r)	Skills available from candidate resources (c)				Resource capabilities (cs)
	OO design (c_1)	C++ (c_2)	VB (c_3)	Java (c_4)	
r_1	1.3125/(VL, 0.31)	1/(VL, 0)	4.6875/(VH, -0.31)	3.25/(M, 0.25)	2.0788/(L, 0.07)
r_2	0.6875/(VL, -0.31)	1/(VL, 0)	4.6875/(VH, -0.31)	2/(L, 0)	1.7253/(L, -0.28)
r_3	3/(M, 0)	5.3125/(VH, 0.31)	3/(M, 0)	1.3125/(VL, 0.31)	3.6301/(H, -0.37)
r_4	5.3125/(VH, 0.31)	5.3125/(VH, 0.31)	1/(VL, 0)	0.6875/(VL, -0.31)	3.9407/(H, -0.06)

shown in Table 7. For example, according to project manager e_1 , existing competency in C++ programming contributes at a “Very Very High” level to improvement of skills in object oriented design. Following Steps 2–4 of the approach, as shown above, we get the final aggregated ratings FSR of skill relationships which are presented in Table 8.

Having calculated the final aggregated ratings of skill relationships, we continue by reevaluating the resources' final aggregated ratings using formula (15). For example, we use the final aggregated rating FR_{11} of resource r_1 with respect to skill/competency c_1 (object oriented design) that is found equal to (VL, 0.31) (see Table 6). FR_{11}^{NEW} is then calculated by using formula (15) as follows:

$$\begin{aligned}
 FR_{11}^{NEW} &= \max \left(1.3125; \right. \\
 &\quad \left. \frac{1 \times 5.4286 + 4.6875 \times 1.6875 + 3.25 \times 4.5714}{5.4286 + 1.6875 + 4.5714} \right) \quad (24) \\
 &= \max (1.3125; 2.4124) = 2.4124.
 \end{aligned}$$

The reevaluated final aggregated ratings of the resources are presented in Table 9.

Resource capabilities (i.e., resource-task suitability values) are computed using formula (16), as described in Step 9 of the approach, and the obtained results (final suitability values) are presented in the last column of Table 9. The comparison between the initial ratings (last column in Table 6) and the final ratings (last column in Table 9) of the candidate resources shows that definitely the most suitable candidate

human resource to be involved in the activities of task T_1 is resource r_4 .

By studying the results of the presented example, we can, for example, conclude that while in the initial evaluation the resources r_3 and r_4 are both highly suitable to cover the skill requirements of task T_1 , with ratings (H, -0.37) and (H, -0.06), respectively (see Table 6, last column), considering skill-relationships (see Table 9, last column) r_4 ends up being more suitable (rated as (VH, -0.04)) with greater difference from r_3 (rated as (H, 0.13)).

Let us now assume that the current workload of the software company comprises two tasks $\{T_1, T_2\}$. This means that in parallel with T_1 the software company is undertaking the development activities of an additional task T_2 . Performing the approach again results in resource-task suitability ratings of the available developers for the development activities of task T_2 . These ratings are shown in Table 10 (third column) along with the ratings of developers for the activities of task T_1 (second column). It should be noted that there would be no need to ask project managers to express again their judgments for the skill level characterizing the developers in case that a required skill for task T_2 is the same as a skill required for task T_1 . We can, therefore, apply formula (16) on these resource-task suitability values to derive estimations of the allocation of each developer effort to tasks T_1 and T_2 (shown in the fourth and fifth column of Table 10, resp.).

Some useful assertions can be derived by observing and comparing these estimated effort allocations. For example, although resource r_4 is slightly more suitable for task T_1 (VH, -0.04), compared with the corresponding suitability

TABLE 7: Project manager evaluations on skill/competency relationships.

Required skills/competencies (c)	Project manager (e)/weight of importance (ϵ)	Skills available from candidate resources (c)			
		OO design (c_1)	C++ (c_2)	VB (c_3)	Java (c_4)
OO design (c_1)	$e_1/(1/3)$	—	(VH, 0)	(M, 0)	(VH, 0)
	$e_2/(1/3)$	—	(H, 0)	(L, 0)	(VH, 0)
	$e_3/(1/3)$	—	(VVH, 0)	(M, 0)	(VH, 0)
C++ (c_2)	$e_1/(1/3)$	(VVH, 0)	—	(L, 0)	(VH, 0)
	$e_2/(1/3)$	(H, 0)	—	(M, 0)	(H, 0)
	$e_3/(1/3)$	(VVH, 0)	—	(M, 0)	(VH, 0)
VB (c_3)	$e_1/(1/3)$	(L, 0)	(M, 0)	—	(L, 0)
	$e_2/(1/3)$	(L, 0)	(L, 0)	—	(VL, 0)
	$e_3/(1/3)$	(VL, 0)	(M, 0)	—	(VL, 0)
Java (c_4)	$e_1/(1/3)$	(H, 0)	(VH, 0)	(VL, 0)	—
	$e_2/(1/3)$	(H, 0)	(H, 0)	(M, 0)	—
	$e_3/(1/3)$	(VVH, 0)	(H, 0)	(M, 0)	—

TABLE 8: Final aggregated ratings of skill/competency relationships FSR.

Skills/competencies (c)	Skills/competencies (c)			
	OO design (c_1)	C++ (c_2)	VB (c_3)	Java (c_4)
OO design (c_1)	—	5/(VH, 0)	2.6875/(M, -0.31)	5/(VH, 0)
C++ (c_2)	5.4286/(VH, 0.43)	—	2.6875/(M, -0.31)	4.6875/(VH, -0.31)
VB (c_3)	1.6875/(L, -0.31)	2.6875/(M, -0.31)	—	1.6875/(L, -0.31)
Java (c_4)	4.5714/(VH, -0.43)	4.3125/(H, 0.31)	2.4286/(L, 0.43)	—

of resource r_3 (H, 0.13), the allocation of r_4 effort to task T_1 is lower (45.70%) than the corresponding value of r_3 (50.18%). This is because resource r_4 should devote more time/effort (54.30%) to the development activities of task T_2 where the suitability resource r_4 (VVH, -0.11) is considerably higher than the corresponding suitability value of resource r_3 (H, 0.10).

6. Discussion

Since proper and systematic utilization of all available resources in a software development project is a vital issue, we proposed an approach that focuses on the efficient evaluation and selection of human resources in software development tasks. The approach contributes to and objectively supports project management in the difficult task of selecting and allocating human resources in software development tasks.

The proposed approach makes two important contributions to the literature. The first involves putting forward for use and development a framework for evaluating human resources to software development projects. We consider a group-based and multicriteria decision making mechanism for the assessment of human resources. We combine the 2-tuple fuzzy linguistic model with similarity degree-based aggregation to derive objective assessments for provided/required skills/competencies. The second contribution is that it emphasizes deriving objective evaluations of human resources matching skills required by software tasks by considering possible skill relationships and dependencies.

Using subjective evaluations (expressed by project managers in qualitative linguistic forms) on skills/competencies required for specific task activities, available human resource skills/competencies, and relationships between the various skills/competencies, 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 of the available resources for a specific task can be intensified, thus better indicating the most appropriate candidate for a specific task.

The above contributions are reflected through the results provided by the exemplar case study scenario. The results show that by using subjective evaluations provided by projects managers we can obtain objective results through the similarity degree-based aggregation technique. The reevaluation of human resource suitability through the consideration of skill/competency relationships can significantly enhance the differences between the available human resources. In addition, we have shown that suitability values are practically useful to estimate effort allocation of resources to development tasks.

We should finally mention that the computations underlying the approach are not apparent to the project managers/decision makers which only provide linguistic terms to state their evaluations. Actually, we implemented all required computations in a spreadsheet program that helps to effectively and practically apply the approach with a variety of inputs.

TABLE 9: Reevaluated final aggregated ratings of resources FR^{NEW} and final values of resource capabilities cs.

Resource (r)	Skills available from candidate resources (c)				Resource capability/suitability (cs) (reevaluated)
	OO design (c_1)	C++ (c_2)	VB (c_3)	Java (c_4)	
r_1	2.4124/(L, 0.41)	2.7646/(M, -0.24)	4.6875/(VH, -0.31)	3.25/(M, 0.25)	3.07569/(M, 0.08)
r_2	1.9235/(L, -0.08)	2.0550/(L, 0.06)	4.685/(VH, -0.31)	2/(L, 0)	2.5139/(M, -0.49)
r_3	3.4140/(M, 0.41)	5.3125/(VH, 0.31)	3.2712/(M, 0.27)	3.9529/(H, -0.05)	4.1274/(H, 0.13)
r_4	5.3125/(VH, 0.31)	5.3125/(VH, 0.31)	3.8731/(H, -0.13)	4.6727/(VH, -0.33)	4.9597/(VH, -0.04)

TABLE 10: Resource capabilities cs to perform the development activities of two tasks and allocation of developers' effort EEA to tasks.

Resource (r)	Capability/suitability cs for task T_1	Capability/suitability cs for task T_2	Allocation of effort to task T_1	Allocation of effort to task T_2
r_1	3.07569/(M, 0.08)	5.58151/(VVH, -0.42)	35.53%	64.47%
r_2	2.5139/(M, -0.49)	4.42354/(H, 0.42)	36.24%	63.76%
r_3	4.1274/(H, 0.13)	4.0976/(H, 0.10)	50.18%	49.82%
r_4	4.9597/(VH, -0.04)	5.89241/(VVH, -0.11)	45.70%	54.30%

7. Conclusion

A fuzzy linguistic description approach based on 2-tuple fuzzy linguistic terms to evaluate human resources involved in software development tasks is presented in this paper. The proposed approach follows a group and similarity degree-based aggregation method to obtain an objective assessment of the ratings of multiple required task-related skills and provided skills from the available human resources. In addition, we consider skill relationships, provided as 2-tuple fuzzy linguistic terms, to represent dependencies between the various task-related skills reflecting the contribution of one skill to the learning of other skills. Through the introduction of skill relationships we ensure the objective assessments of human resources matching skills required by software tasks.

To demonstrate the applicability of the proposed approach we presented an exemplar case study scenario with numerical results, which was implemented to assess the results obtained. The numerical results provide indications that the proposed approach can effectively support management decisions to objectively evaluate available human resources considering a set of skills and a set of project tasks. We are currently executing a case study in software SME with the aim of thoroughly validating the applicability of the proposed approach. In particular, since the approach utilizes and objectively aggregates multiple qualitative judgments for task-related and provided skills, our current interest is to apply the approach in the context of a software development company that follows 360-degree appraisal [35] to evaluate its personnel, where the judgment of each company employee about his/her level of competence is also considered.

Our future work aims at further improving 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 multiobjective optimisation (e.g., according to budget and availability constraints).

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research has been cofinanced 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."

References

- [1] D. A. Callegari and R. M. Bastos, "A multi-criteria resource selection method for software projects using fuzzy logic," in *Enterprise Information Systems*, J. Filipe and J. Cordeiro, Eds., vol. 24 of *Lecture Notes in Business Information Processing*, pp. 376–388, Springer, Berlin, Germany, 2009.
- [2] L. D. Otero, G. Centeno, A. J. Ruiz-Torres, and C. E. Otero, "A systematic approach for resource allocation in software projects," *Computers and Industrial Engineering*, vol. 56, no. 4, pp. 1333–1339, 2009.
- [3] S. T. Acuña, N. Juristo, and A. M. Moreno, "Emphasizing human capabilities in software development," *IEEE Software*, vol. 23, no. 2, pp. 94–101, 2006.
- [4] F. Padberg, "Scheduling software projects to minimize the development time and cost with a given staff," in *Proceedings of the Software Engineering Conference (APSEC '01)*, pp. 187–194, 2001.
- [5] 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, no. 1, pp. 654–662, 2012.
- [6] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

- [7] F. Herrera and L. Martínez, "A 2-tuple fuzzy linguistic representation model for computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 6, pp. 746–752, 2000.
- [8] X. Liao, Y. Li, and B. Lu, "A model for selecting an ERP system based on linguistic information processing," *Information Systems*, vol. 32, no. 7, pp. 1005–1017, 2007.
- [9] V. C. Gerogiannis, E. Rapti, A. Karageorgos, and P. Fitsilis, "Human resource assessment in software development projects using fuzzy linguistic 2-tuples," in *Proceedings of the 2nd International Conference on Artificial Intelligence, Modelling & Simulation (AIMS '14)*, pp. 217–222, IEEE, Madrid, Spain, November 2014.
- [10] V. C. Gerogiannis, E. Rapti, A. Karageorgos, and P. Fitsilis, "A fuzzy linguistic approach for human resource evaluation and selection in software projects," in *Proceedings of the 5th International Conference on Industrial Engineering and Operations Management (IEOM' 15)*, pp. 1–9, Dubai, UAE, March 2015.
- [11] SPRINT SMEs Project, January 2015, <http://sprint.teilar.gr/>.
- [12] V. C. Gerogiannis, G. Kakarontzas, L. Anthopoulos, S. Bibi, and I. Stamelos, "The SPRINT-SMEs approach for software process improvement in small-medium sized software development enterprises," in *Proceedings of the Scientific Workshop of R&D Projects in ARCHIMEDES III Research Programme*, Technological Education Institute of Thessaly, Larissa, Greece, November 2013.
- [13] S. Bibi, V. C. Gerogiannis, G. Kakarontzas, and I. Stamelos, "Ontology based bayesian software process improvement," in *Proceedings of the 9th International Conference on Software Engineering and Applications (ICSOFTEA '14)*, pp. 568–575, Vienna, Austria, August 2014.
- [14] 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, no. 3, pp. 259–275, 2011.
- [15] H. I. Ahmad, *A systematic review of selection and management of a project team [M.S. thesis]*, Joensuu School of Computing, Faculty of Science and Forestry, University of Eastern Finland, 2013.
- [16] T. J. A. Santos, A. M. Lima, C. A. Reis, and R. Q. Reis, "Automated support for human resource allocation in software process by cluster analysis," in *Proceedings of the 4th International Workshop Recommendation Systems for Software Engineering*, pp. 30–31, Hyderabad, India, June 2014.
- [17] A. Barreto, M. D. O. Barros, and C. M. L. Werner, "Staffing a software project: a constraint satisfaction and optimization-based approach," *Computers and Operations Research*, vol. 35, no. 10, pp. 3073–3089, 2008.
- [18] 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, no. 1, pp. 100–108, 2013.
- [19] R. Kolisch, "Resource allocation capabilities of commercial project management software packages," *Interfaces*, vol. 29, no. 4, pp. 19–31, 1999.
- [20] 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, no. 5, pp. 413–421, 2010.
- [21] H.-T. Tsai, H. Moskowitz, and L.-H. Lee, "Human resource selection for software development projects using Taguchi's parameter design," *European Journal of Operational Research*, vol. 151, no. 1, pp. 167–180, 2003.
- [22] M. Yoshimura, Y. Fujimi, K. Izui, and S. Nishiwaki, "Decision-making support system for human resource allocation in product development projects," *International Journal of Production Research*, vol. 44, no. 5, pp. 831–848, 2006.
- [23] D. Kang, J. Jung, and D.-H. Bae, "Constraint-based human resource allocation in software projects," *Software: Practice and Experience*, vol. 41, no. 5, pp. 551–577, 2011.
- [24] 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 (IEA/AIE '88)*, pp. 116–125, 1988.
- [25] W. Kwak, Y. Shi, and K. Jung, "Human resource allocation in a CPA firm: a fuzzy set approach," *Review of Quantitative Finance and Accounting*, vol. 20, no. 3, pp. 277–290, 2003.
- [26] M. L. Hussein and M. A. Abo-Sinna, "A fuzzy dynamic approach to the multicriterion resource allocation problem," *Fuzzy Sets and Systems*, vol. 69, no. 2, pp. 115–124, 1995.
- [27] D. Strnad and N. Guid, "A fuzzy-genetic decision support system for project team formation," *Applied Soft Computing Journal*, vol. 10, no. 4, pp. 1178–1187, 2010.
- [28] N. A. Ruskova, "Decision support system for human resources appraisal and selection," in *Proceedings of the 1st International IEEE Symposium on Intelligent Systems (IS '02)*, pp. 354–357, IEEE, Varna, Bulgaria, 2002.
- [29] J. Dodangeh, S. Sorooshian, and A. R. Afshari, "Linguistic extension for group multicriteria project manager selection," *Journal of Applied Mathematics*, vol. 2014, Article ID 570398, 8 pages, 2014.
- [30] R. M. Rodriguez and L. Martinez, "An analysis of symbolic linguistic computing models in decision making," *International Journal of General Systems*, vol. 42, no. 1, pp. 121–136, 2013.
- [31] L. Martinez and 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.
- [32] T. Da and Y. Xu, "Evaluation on functions of urban waterfront redevelopment based on proportional 2-tuple linguistic," *International Journal of Computational Intelligence Systems*, vol. 7, no. 4, pp. 796–808, 2014.
- [33] M. Dursun and E. E. Karsak, "A fuzzy MCDM approach for personnel selection," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4324–4330, 2010.
- [34] H. Raoudha, D. E. Mouloudi, H. Selma, and E. M. Abderahman, "A new approach for an efficient human resource appraisal and selection," *Journal of Industrial Engineering and Management*, vol. 5, no. 2, pp. 323–343, 2012.
- [35] M. Espinilla, R. de Andrés, F. J. Martinez, and L. Martinez, "A 360-degree performance appraisal model dealing with heterogeneous information and dependent criteria," *Information Sciences*, vol. 222, pp. 459–471, 2013.
- [36] 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, no. 1, pp. 33–48, 2001.
- [37] F. Herrera, E. Herrera-Viedma, and J. L. Verdegay, "A rational consensus model in group decision making using linguistic

assessments,” *Fuzzy Sets and Systems*, vol. 88, no. 1, pp. 31–49, 1997.

- [38] F. Herrera, L. Martínez, and P. J. Sánchez, “Managing non-homogeneous information in group decision making,” *European Journal of Operational Research*, vol. 166, no. 1, pp. 115–132, 2005.

