An ontology-based approach to assigning human resources to Software projects

Mario Andrés Paredes-Valverde
María del Pilar Salas-Zárate
Ricardo Colomo-Palacios
Juan Miguel Gómez-Berbís
Rafael Valencia-García

Abstract

Human resources play a critical role in the success of software projects. Ensuring the correct assignment of them to a specific project is, therefore, an immediate requirement for Software development organizations. Within this context, this work explores the use of ontologies in the building of a decision support system that will help human resources managers or project leaders to select those employees who are best suited to participating in a new software development project. Ontologies allow the system to discover semantic relatedness among new and previous software projects by means of its requirements specification. The system can, therefore, suggest those people who have participated on similar projects. We have proved the effectiveness of our approach by conducting an evaluation in a software development organization. Our findings confirm the success of our approach and reveal that it may bring considerable benefits to the software development process.

Keywords

Software project; ontologies; semantic indexing; assigning human resources

1. Introduction

The need for software development process improvement has always existed in the industry, but it has become even more pressing within the current economic context since software development organizations are increasingly demanding better practices that consider not only factors such as budget and time, but also the competencies of its personnel. The term competence refers to the state or quality of being adequately or well qualified [1]. Human resources play a critical role in software project success [2,3]. Hence ensuring the correct assignment of them to a specific software project is an immediate requirement in which software development organizations must put special attention. Nowadays, software development teams are formed on the basis of human resources manager's experience about people, constraints (e.g. availability), and skill requirements, but this experience is not systematically recorded [4]. This practice becomes more complex and even impossible in big organizations and SMEs (Small and Medium-sized Enterprises) because of the number of available employees.

A software development process can be defined as an environment of capable interrelated resources managing a sequence of activities using appropriate methods and practices to develop a software product that conforms to customer's requirements [5]. Software development process involves several activities such as requirement analysis, software design, implementation, testing, integration, deployment, and maintenance. Software requirements specification (SRS) is one of the keys to success in software development. An SRS is a description of a software product to be developed. It establishes the basis for an agreement between customers and suppliers on what the software product will, and, if necessary, will not do [6]. The SRS document enlists enough and necessary requirements that are required for the project development [7]. Hence, based on this document it is possible to determine what are the technological knowledge and skills that a person must have to be integrated into the development team in charge of a project.

Semantic Web aims to provide to Web information with a well-defined meaning and make it understandable not only by humans but also by computers [8]. In the Semantic Web, the ontologies are the fundamental technology for domain modelling. An ontology is a formal and explicit specification of a shared conceptualization [9], i.e. it provides a formal, structured knowledge representation, with the advantage of being reusable and shareable. This knowledge is mainly formalized through five kinds of components, namely classes, relations, attributes, axioms, and instances. The ontologies have been successfully applied to numerous domains, including finances [10], question and answering [11], human perception [12], and cloud services [13], among others.

This work presents an ontology-based decision support system capable of automatically suggesting the human resources who are best suited for participating in a new software development project. This system uses ontologies to model the domain where it will be implemented, thus allowing to formalize both the software requirements specification and the personnel competencies. In this way, the proposed system is able to recommend personnel that best fit with the software requirements specification, based on the semantic relatedness among them.

The remainder of this paper is organized as follows. Section 2 outlines the state of the art of technologies involved in this work. The components that take part in the system proposed and its overall architecture are described in Section 3. In section 4, a use case scenario in the software development domain is presented. Finally, conclusions are forward in Section 5.

2. Related work

Nowadays, there are prominent efforts to provide systems that assist the process of assignment of human resources to software projects. For instance, in [4] presents a formal model for assigning human resources to software projects based on the Delphi method, an expert consultation method whose essence is to organize an anonymous dialogue among experts consulted individually by means of questionnaires. This method was applied to determine the criteria for selecting experts and to draft a proposal of fixed roles and competences required to tackle software projects. The work presented in [14] proposes a methodology to assign resources to tasks when optimum skill sets are not available. This methodology considers existing capabilities of candidates, required levels of expertise, and priorities of requires skills for the task. In [15], the authors present a release planning method for developing software incrementally. This method assigns human resources based on their productivity executing different kinds of tasks. However, this method is only applicable in mature software organizations that have well-defined and measured processes. Another work [16] proposes a flexible and effective model for software project planning, this model is based on an event-based scheduler (EBS) and an ant colony optimization (ACO) algorithm. The proposed model represents a plan by a plan list and a planned employee allocation matrix to consider task scheduling and employee allocation. The work presented in [17] propose a methodology based on dynamic programming, to assign human resources to software development projects. This methodology considers the complexity of each project, the existing capabilities of staff and the skills required for the project. In [18] the authors present an optimisation model to address the problem of assigning project work to multi-skilled internal and external human resources while considering learning, depreciation of knowledge and company skill level targets. In [19] authors developed a method to select the proper set of human resources to form a software development team to complete the project at minimum cost and cycle time. The proposed method uses Taguchi's parameter design approach and the critical resource diagram (CRD), which focuses on resource rather than activity scheduling. The method considers the technical skills (these were estimated as average numbers of software lines of code per day) and the salary of the candidates. On the other hand, there are works focused on the topic of roles. For example, in [20] the authors stated the importance or roles in human resources assignment. They propose a role-based software engineering process using primitive roles. Furthermore, they developed a prototype tool to dynamically add, specify and modify roles. In the same context, in [21] the author proposed a human resources management process for software development projects based on the reuse of organizational knowledge of competences and previous role assignments. According to this process, the project leaders assign people to each project task considering the defined profile.

The works above propose a variety of solutions to the human resources assignment problem. All these works have in common that they somehow consider the employees' competencies in the human resources assignment. However, none of them take into account the previous participation of the employees in similar software projects. The research effort presented in this work provides a decision support system that aims to assists human resources managers to assign the workers who are best suited for participating in a new software development project. This system has been designed to take advantage of well-known XML vocabularies to describe the personnel competences as well as their experience on previous software projects. Furthermore, our approach integrates semantic technologies, more specifically ontologies, to relate the information resources used by the system (personnel profile and SRS documents) and then be able to suggest personnel based on their previous participation in similar projects. In the following sections, the architecture design and component functionality of the present approach are described in detail.

3. System architecture

The main goal of the decision support system proposed here is to assists in the human resources assignment to new software development projects. The architecture of this system relies on four main elements: information resources (SRS documents and employees' profile), ontologies repository, semantic indexing module, and the decision support module. Figure 1 shows the complete architecture of the system.

In a nutshell, the system works as follows. Firstly, a semantic profile of each employee of the organization is generated. This profile includes, among other things, information about the employee's experience in previous software development projects. Then, SRS documents of previous projects are semantically annotated by the semantic annotation module. For this purpose, the ontological model describing the ICT domain is used. Based on the annotations generated, the semantic indexing module creates a semantic index by using the vector space model. This semantic index is stored in a database. Once the information resources have been processed, the system is able of automatically suggesting the human resources who are best suited for participating in a new software development project.

When a new project is generated and its corresponding SRS document is created, this last annotated by the semantic annotation module. Then, the projects similarity module generates a vector representation of the SRS document from the resulting annotations. This vector is compared to the ones stored in the database by using the cosine metric. The previous task aims to obtain the most similar software projects to the new one. Once these projects have been selected, the HR suggesting module obtains all employees that have participated in those projects. Finally, this module ranks the employees obtained and suggests those ones that best fit with the software requirements specification provided.

Next sections provide a description in detail of each component of the architecture, as well as its main contribution.

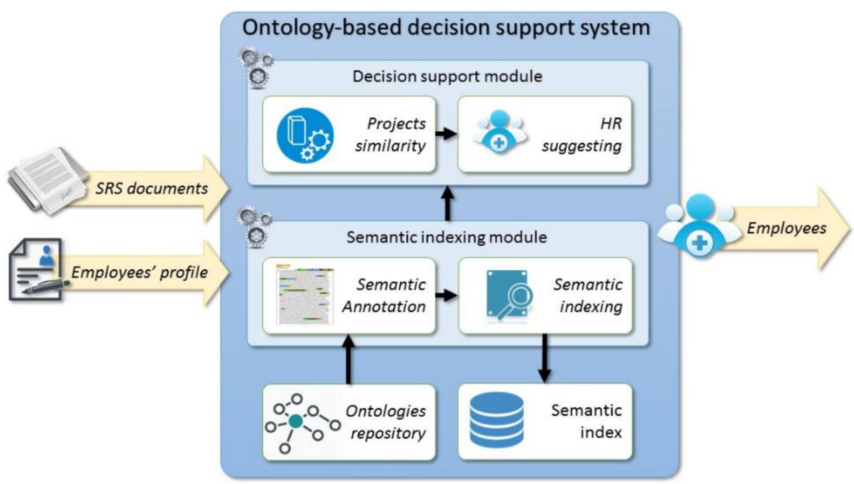

Figure 1. System architecture.

3.1. Information resources

The information resources used by our system are the SRS documents and the semantic profile of each person that could has the responsibility to execute work units along software development process. The SRS documents used in this work are based on requirements specification standards, specifically IEEE [22,23]. Regarding the employees' profile, we established a semantic profile for describing it, stressing the knowledge and skills of employees on software development technologies as well as their experience on previous software projects. The SRS document and the employees' semantic profile are described in detail below.

3.1.1. SRS documents

The SRS are represented textually through documents organised according to the IEEE Recommended Practice for Software Requirements Specifications [22] and the IEEE Guide for Developing System Requirements Specifications [23]. The SRS document used in this work consists of several sections with its corresponding subsections. However, next we present a general description of five main sections:

(1) Introduction. This section should provide an overview of the entire SRS. Also, it must identify the software product by name, explain what the software product will, and, if necessary, will not do, describe the benefits, objectives and goals, among other issues.
(2) Overall description. It should describe the general factors that affect the product and its requirements. It provides a background regarding product perspective, functions, user classes and characteristics, operating environment, design and implementation constraints, and assumptions and dependencies.
(3) External interface requirements. It should provide information related to user interfaces, hardware interfaces, software interfaces, and communication interfaces.
(4) System features. It should contain all software requirement to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and tester to test that the system satisfies those requirements. This section can be organized by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these.
(5) Other non-functional requirements. This section describes performance requirements for the product under various circumstances. This section includes information regarding security requirements, software quality attributes, and business roles.

The SRS document should be produced such way that all the participants can understand it. Furthermore, it is the basis for all other activities of development and its quality is fundamental for the project success [24]. On the basis of this understanding, we believe that this document is crucial for determining the human resources who are the best suited for participating in a specific software development project since it describes all requirements through concepts that make reference to technologies and markets in which the organization employees must be competent.

3.1.2. Employees' profile

The semantic employees' profile aims to describe the competencies of the organization personnel emphasizing in their knowledge and skills regarding software development technologies as well as their experience on previous software projects. Nowadays there are RDF schemas and XML vocabularies that helps to describe curriculum vitae. For instance, the ResumeRDF [25] is an ontology developed to express information contained in a personal resume or curriculum vitae on the Semantic Web. This ontology allows describing the courses, education, language skills, work history, among other things. Another example is FOAF (Friend of a Friend) [26], a widely accepted standard vocabulary for information about people and their social connections. FOAF contains terms for describing personal information, membership in groups, and social connections.

Although these approaches represent a great effort to address the competencies description issue, the vocabulary proposed by them is insufficient for the objectives of this work. For example, these vocabularies do not consider technological skills, business markets skills, or the experience in previous software projects within a specific organization. Because of this understanding, an RDF schema has been proposed by extending the vocabulary provided by the XML vocabularies above presented. The main objective of this RDF schema is to improve the semantic description of employees' competences within the context of software development. In Figure 3, an example of employee's competencies description by using the RDF schema proposed in this study is shown. In summary, the semantic employees' profile describes the language, technology and markets skills, the education, courses, and the projects in which he/she has been involved.

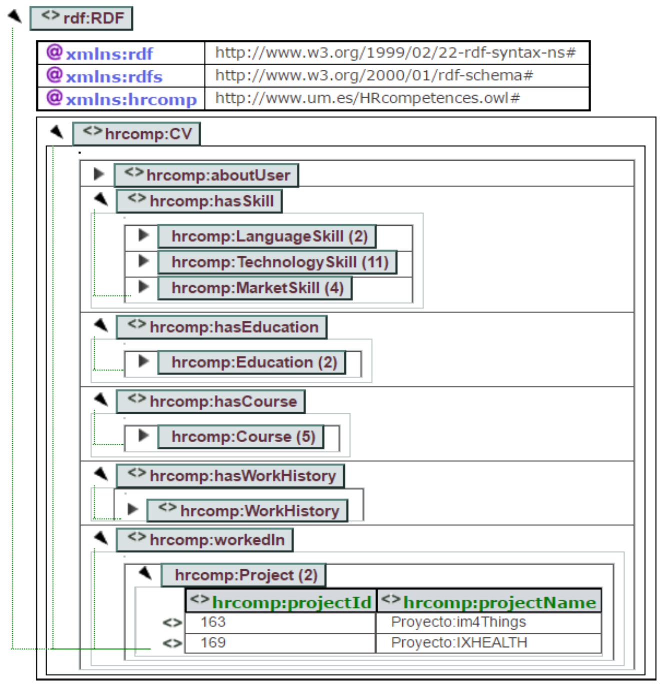

Figure 2. Excerpt of the employees' profile.

3.2. Ontologies repository

The decision support system proposed in this work requires a formal representation of the domain where it will be implemented. In this sense, the ontologies repository stores all ontological models that will be used by the semantic annotation module to perform its corresponding tasks. To achieve the main goal of this work, an ontology describing Information and Communications Technology (ICT) has been designed and implemented using the OWL 2 Web Ontology Language [27] and the Protégé software [28]. Next, this ontology is described.

3.2.1. ICT's ontology

This ontology offers a common vocabulary for the ICT domain. The ICT is a term that includes any communication device, application, as well as the services associated with them that enable users to access, store, transmit, and manipulate information in a digital form. The design and development of this ontology are based on the work presented in [29] which involves the use of the Wikipedia website as the main information source. Wikipedia is a multilingual, web-based, free-content encyclopaedia project based on a model of openly edit content, i.e., it is written collaboratively by the people who use it. We have performed a concepts' extraction process taking advantage of the categorization function in Wikipedia. This process results in an ontology consisting of 2134 concepts that form a hierarchy that can be used by different kind of applications or tasks. Furthermore, this ontology contains 155 object properties and 39 datatype properties. The terms contained in this ontology enable to describe software development industry-needed skills through different areas such as software engineering, web programming, databases, mobile development, to mention but a few. Figure 3 shows an excerpt of the ICT's ontology with some of the generic terms contained within it.

3.3. Semantic indexing module

3.3.1. Semantic annotation

The information resources on which relies the semantic indexing module are the SRS documents and the personnel semantic profiles. Firstly, the SRS documents are processed by means of natural language processing (NLP) techniques to obtain a set of semantic annotations in accordance with the ontologies stored in the ontologies repository. The goal of this process is to align a resource, in this case the SRS document, with a description of some of its properties and characteristics with respect to a formal conceptual model [30], in this case, the ICT's ontology. This alignment will allow the decision support module to found semantic similarities between the software projects and between these projects and the personnel's profile. The semantic annotation process is performed using the GATE framework [31], an infrastructure for developing and deploying software components that process human language.

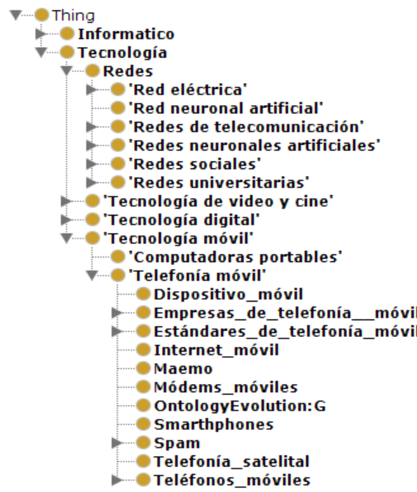

Figure 3. Excerpt of the ICT's ontology.

3.3.2. Semantic indexing

Once information resources have been semantically annotated, this module assigns a weight to the annotations aiming to reflect how relevant the ontological entity is for the SRS document meaning. The weight assignment is fundamental to computing semantic relatedness among the information resources involved in this work, thus allowing the decision support module to suggest personnel selection that best fit with a specific software project. This process is based on the work presented in [32] which provides an adaptation of the classic vector-space model [33]. In this approach, weights are computed automatically by an adaptation of the TF-IDF algorithm [33], based on the frequency of occurrence of the ontological entity in each SRS document. More specifically, the weight of each ontological entity is computed by using equation 1.

$$(tf - idf)_{i,p} = \frac{n_{i,p}}{\sum_k n_{k,p}} * \log \frac{|P|}{N_i} \tag{1}$$

where $n_{i,p}$ is the number of occurrences on the ontological entity *i* in the SRS document *p*, $\sum_k n_{k,p}$ is the sum of the occurrences of all the ontological entities identified in the SRS document *p*, $|P|$ is the set of all SRS documents, and N_i is the number of all SRS documents annotated with *i*.

The annotations weighting process used until now considers only those ontological entities that explicitly appear in the SRS documents. To enrich the semantic meaning of these documents, we use an extension of the algorithm above to consider not only those entities that explicitly appear in the text description but also those ontological entities that have a taxonomic relation with the first ones. This extension has been implemented using the Dijkstra's algorithm [34] which allows finding the shortest paths between nodes in a graph. In this case, Dijkstra's algorithm is used to determine the distance between two classes of the ontology based on the taxonomic relations of them. Therefore, this module also assigns a weight to those ontological concepts whose distance with respect to the annotated concept is less than 2. More specifically, the weight of this kind of annotations is computing through equation 2.

$$v_i = \sum_{j=1}^{n} \frac{tf-idf_{j,p}}{e^{dist(i,j)}} \tag{2}$$

where *dist(i,j)* is the semantic distance between concept *i* and concept *j* in the domain ontology. As has been mentioned earlier, the distance is calculated considering the taxonomic relations between concepts, i.e., the relations of type subclass of. Thus, the distance between a concept and itself is 0, the distance between a concept and its taxonomic parent or child is 1, and so on and so forth.

3.4. Decision support module

The decision support module benefits from the vector representations of information sources used in this work to suggest the personnel selection to a new software project considering the semantic relatedness among them. In this way, when a new software project is created, it is necessary to establish its requirements by means of the SRS document. Once this resource is created, the semantic indexing module process the SRS document to obtain a vector representation of it by means of equation 2. From such vector, this module suggests human resources for participating in that project. To this end, this module needs to rank all employees existing in the knowledge base for finding the most appropriate ones, considering their participation in similar software projects. This process consists of two main tasks: measuring the similarity between software projects and obtaining personnel suggestions. Next, these tasks are explained in detail.

3.4.1. Projects similarity

It is possible that a great number of software projects within an organization might be focused on the same topics, hence they can have certain similarity degree. Based on this fact, the aim of this module is to compute the similarity level of the new project with previous projects. In this work, the evaluation of the correlation between software projects considers both the semantic information and annotations previously retrieved by the semantic indexing module and the vector representation of the SRS document of the new project. Therefore, the semantic similarity of software projects is calculated comparing their vectors in the space by using the cosine metric [35] which is commonly used to determine the similarity of documents in the vector space model approach.

The cosine metric calculates the similarity value between the vectors representing the new project and each previous project. It should be remembered that these vectors are calculated by using equation 2. Having said this, semantic similarity is computed using equation 3.

$$sim(P, P') = cos\theta = \frac{P \ x \ P'}{|P| \ x \ |P'|} \qquad (3)$$

where *P* is the vector representation of the new software project, and *P'* is the vector representation of a software project already stored in the system. The θ symbol is the angle separating both vectors and it represents the degree of similarity between them. In this case, the cosine similarity between two software projects ranges from 0 to 1, since the term frequencies (tf-idf weights assigned to each ontological entity by the semantic indexing module) cannot be negative. When two software projects are very similar their vectors overlap, and when they have less similarity, their vectors start to diverge, i.e., there is a bigger angle between them.

As we can see, equation 3 helps to determine the similarity between two software projects. However, it is necessary to obtain the similarity level between all projects stored in the system. To achieve this aim, the present module obtains all these similarities rates and stored them in a matrix called Project2Project matrix. Figure 4 provides a representation of this matrix, where each $sim_{i,j}$ is a float number that represents the similarity between the projects *i* and *j*.

	SP_1	SP_2	...	SP_n
SP_1	1	$sim_{1,2}$	$sim_{...}$	$sim_{1,n}$
SP_2	$sim_{2,1}$	1	$sim_{...}$	$sim_{2,n}$
...	$sim_{...}$	$sim_{...}$	1	$sim_{...}$
SP_n	$sim_{n,1}$	$sim_{n,2}$	$sim_{...}$	1

Figure 4. Software projects similarity matrix.

3.4.2. HR suggesting

The second task performed by the decision support module consists in suggesting people that can be involved in the new software project in accordance with their experience in previous projects. With this aim, this module obtains, from the Project2Project matrix, the most similar projects to the new one based on a threshold. To calculate this threshold, we use the statistical method of percentile (deciles, quartiles, etc.). The percentile indicates the value of a variable (in this case, the similarity rate) below which a given percentage of observations in a group of observations fall. In this case, this module estimates the threshold value as the similarity rate that separates 80% of the levels of similarity, represented by

P80, i.e. the similarity rate that cuts apart 80% of the software projects, leaving the other 20%, which represents meaningful similar projects. Percentile is calculated by using equation 4, which was presented in [36].

$$k = \frac{p\,x\,n}{100} \tag{4}$$

where k is the position corresponding to the similarity rate that separates the sample (in which the similarity rates are sorted in ascending order) in p percentile, p represents the percentile we are looking for, and n represents the sample size.

Once the set of similar software projects has been obtained, it is necessary to obtain all participants of these projects. This module obtains such data from the ontology repository by means of SPARQL queries. Having obtained this information, the system needs to associate a score with each person aiming to determine the personnel that best fit with the new software project. Hence, the score of each person is obtained by summing the similarity rates of the most similar projects in which he/she has participated, which were previously obtained by means of the percentile measure. This algorithm is shown in equation 5.

$$peopleScore(A,P) = \sum_{workedIn(A,P_i)} projectSim(P,P_i) \tag{5}$$

where $projectSim(P,P_i)$ refers to the similarity rate between the new project (P) and one of the most similar projects (P_i).

Finally, all employees are sorted by their score. Furthermore, for each person, the system provides information concerning the projects he/she has worked on, as well as the roles that he/she has played in those projects. To prove the effectiveness of the approach presented in this work, we conducted an evaluation, which will be explained in the next section.

4. Case study

4.1. Method

In this work, we have evaluated our proposal aiming to measure its effectiveness regarding software project similarity measurement and assignment human resources to Software projects. To this end, our system was implemented in a software development SME which offers solutions and services for different markets such as public administration, healthcare, finance, among others. This organization is characterized by the use of agile methodologies attached to international software development standards such as the IEEE recommendations mentioned in previous sections. The overall evaluation process is described below.

To obtain the semantic meaning of all information resources to be used by the system, it is necessary to have a formal representation of the domain where it will be implemented. Considering that a great number of software projects developed by the SME under consideration relies on the domain of ICT, we have used the ontology presented in section 3.2.1.

The software development processes followed by the organization are attached to international standards which allow the organization to provide their clients high-quality software products and services. Among the standards used by this organization are the IEEE recommendations for requirements specification. Therefore, the SRS of each project follows the document template presented in section 3.1.1, a practice that the organization has used for many years. As was explained in that section, this template includes all information related to different kinds of requirements such as functional requirements, software system attributes, external interfaces requirements, to mention but a few. Around 50 SRS documents within the ICT's domain were processed by the semantic indexing module in order to represent each of them through vectors, as well as to create the Project2Project matrix, which contains the similarity rates between all projects.

Personnel information related to their experience on previous software projects within the organization was represented by means of the semantic profile presented in section 3.1.2. Also, this information was stored in the ontology repository, which was implemented using Virtuoso [37], an SQL-ORDBMS and Web Application Server that provides SQL, XML, and RDF data management. A total of 38 profiles were introduced into the system. It should be mentioned that an average of 7 people was involved in each project.

Once all information resources have been introduced, the experiment took place. Then, 10 new software projects within the context of ICT were proposed and described by means of the SRS document template. Then, human resources managers of the organization manually selected, by each new project, the most related projects as well as the personnel that best fit with the project. Finally, the set of selected projects and the personnel suggested by the experts of the organization were compared to the results obtained automatically by our system. In this evaluation, we used the primary metric precision, recall and their harmonic mean, known as the F-measure. These alternative metrics of inter-rater agreement have been applied by researchers in the context of information retrieval [38].

$$precision = \frac{correct\ suggestions}{total\ suggestions} \tag{6}$$

$$recall = \frac{correct\ suggestions}{number\ of\ relevant\ items} \tag{7}$$

$$F - measure = 2 * \frac{precision * recall}{precision + recall} \tag{8}$$

The precision metric refers to the proportion of suggested items (software projects or personnel) that the system classified as relevant. This score was obtained by dividing the number of correct suggestions obtained by the system by the total suggestions obtained by the system. The recall metric refers to the proportion of relevant items retrieved by the system. This score was obtained by dividing the number of correct suggestions retrieved by the system by the total number of items suggested by the experts of the organizations. Finally, the F-measure score is the weighted average of the precision and recall. It represents twice the product of precision and recall divided by the sum of precision and recall; F-measure score reaches its best value at 1 and the worst score at 0.

4.2. Discussion of the evaluation results

The global experiment results for software projects similarity and assignment human resources to Software projects are reported in Table 1. Regarding project similarity, the present approach obtained an average rate of 0.7940 for the precision metric, 0.8182 for the recall metric, and 0.8037 for the F-measure metric. The best result was obtained for project 6, with an F-measure value of 0.8571. Meanwhile, the worst result was obtained for the project 10, with an F-measure score of 0.7059. With regards to personnel suggestions, the system got an average of 0.7541 for the precision metric, 0.7464 for the recall metric, and 0.7456 for the F-measure metric. The best result was obtained for project 3, with an F-measure score of 0.8421. Meanwhile, the worst result was obtained for the project 10 again, with an F-measure score of 0.5455.

Table 1. Evaluation results for projects similarity and human resources suggestions.

	Projects			Human resources		
Project	Precision	Recall	F-measure	Precision	Recall	F-measure
1	0.8000	0.8000	0.8000	0.8000	0.6667	0.7273
2	0.8333	0.8333	0.8333	0.8889	0.7273	0.8000
3	0.7778	0.8750	0.8235	0.8000	0.8889	0.8421
4	0.8182	0.8571	0.8372	0.7143	0.8333	0.7692
5	0.8000	0.8571	0.8276	0.8571	0.7500	0.8000
6	0.8400	0.8750	0.8571	0.8333	0.7143	0.7692
7	0.7500	0.8182	0.7826	0.6667	0.7500	0.7059
8	0.8889	0.8000	0.8421	0.7143	0.8333	0.7692
9	0.8000	0.6667	0.7273	0.6667	0.8000	0.7273
10	0.6316	0.8000	0.7059	0.6000	0.5000	0.5455
Avg.	0.7940	0.8182	0.8037	0.7541	0.7464	0.7456

The results obtained by our approach seem promising. Furthermore, as can be observed in Table 1, there is no a big difference between the F-measure scores obtained for all software projects. However, based on a detailed analysis of all software projects (the old and new ones) and the profile of each employee, we ascribe these variations to next main reasons: software projects focused on out-of-context topics and short experience of personnel in some technology domains.

Concerning out-of-context topics, even though our approach has an ontology that model the ICT domain, a domain in which most organization projects fall, the SRS documents contains several concepts that are not described by this ontology. This is mainly because of the organization includes several new markets within its catalogue of products and services, which were not considered by the ontology above. For instance, projects 7 and 10 obtained a low precision (0.7500 and 0.6316, respectively) for projects similarity suggestion because these works are focused in the development of solutions on e-health, a domain that was not described in depth by the ontology. This last fact caused that their corresponding SRS documents were not correctly described, i.e. the number of annotations generated for each one was a little bit low. Also, there are not similar projects in the system because this domain has been recently adopted by the organization. In this sense, it is necessary to improve the support for projects whose topics might be different from the domain described in the ontology. From this perspective, we are convinced that our approach can be extended to provide a more robust behaviour with respect to out-of-context domain projects by incorporating new ontologies that model the corresponding contexts. Furthermore, the ontologies already available in this repository can evolve by means of ontology evolution techniques such as the presented in [29].

Regarding second reason given above, we perceive that personnel suggestions results were lower than those obtained for software projects similarity. This fact is mainly due to the employees have not enough experience in the domains that organization has recently adopted such as e-health. Again, this last fact results in a small number of projects related to those domains. Regarding personnel suggestion, human resources managers provided suggestions about the global functionality of the system. One of the most outstanding observation regards to the inclusion of employees that have low experience to non-critical software projects. In this sense, we are convinced that our approach can be enhanced through the incorporation of new weighting mechanisms that consider variables such as the critical level of the project or any other variable that can emerge along the software development process of the organization.

5. Conclusions

In this piece of research, we have presented a semantic-based system able to suggest which personnel could be incorporated into a new software project considering their experience in previous projects. Our proposal obtained encouraging results with an average F-measure score of 0.8037 for similarity projects and 0.7456 for personnel suggestion. The main contribution of our research effort is twofold. First, we propose a semantic-based profile focused on describing the competences and experience of personnel within an organization. This profile integrates well-known XML vocabularies within the context. Second, the integration of semantic techniques to the personnel assignment process within a software development organization can help human resource manager to speed up this task, and even, make it completely automated. All this, with a high level of confidence that the personnel selected by the system is the most appropriate for the project. Also, the method here proposed can be applied to current software development process of the organization without requiring any changes to it. Furthermore, our approach has been designed to be domain-independent, i.e. our system can be implemented in different organizations with the only requirement of providing the system with an ontology that models the corresponding context. The effectiveness of our approach will be evaluated by means of the implementation of it in other software development SMEs whose business markets do not fall into the ICT domain. Finally, it is important to remark that the incorporation of semantic technologies within the project, helps to improve data quality and consistency, thus allowing this information to be used for other purposes, since one of the main goals of semantic web, and more specifically of the ontologies, is to serve as a reference for communication not only among humans but also among computers.

Funding

This work has been supported by the Spanish National Research Agency (AEI) and the European Regional Development Fund (FEDER / ERDF) through project KBS4FIA (TIN2016-76323-R). María del Pilar Salas-Zárate and Mario Andrés Paredes-Valverde are supported by the National Council of Science and Technology (CONACYT), the Secretariat of Public Education (SEP) and the Mexican government.

References

[1] Mifflin EH. The American Heritage Dictionary. 4 edition. Turtleback; 2001.
[2] Gorla N, Lam YW. Who Should Work with Whom? Building Effective Software Project Teams. Communications of the ACM. 2004 Jun;47(6):79–82.
[3] Acuña ST, Gómez M, Juristo N. How do personality, team processes and task characteristics relate to job satisfaction and software quality? Information and Software Technology. 2009 Mar;51(3):627–39.
[4] André M, Baldoquín MG, Acuña ST. Formal model for assigning human resources to teams in software projects. Information and Software Technology. 2011 Mar;53(3):259–75.

[5] Zeineddine R, Mansour N. Software Quality Improvement Model for Small Organizations. In: Yazıcı A, Şener C, editors. Computer and Information Sciences - ISCIS 2003 [Internet]. Springer Berlin Heidelberg; 2003 [cited 2016 Dec 11]. p. 1027–34. (Lecture Notes in Computer Science). Available from: http://link.springer.com/chapter/10.1007/978-3-540-39737-3_127
[6] Committee ICSSES, Board I-SS. IEEE Recommended Practice for Software Requirements Specifications. In Institute of Electrical and Electronics Engineers; 1998.
[7] Pressman RS, Maxim BR. Software Engineering: A Practitioner's Approach. Edición: 8. New York, NY: McGraw-Hill Education; 2014. 976 p.
[8] Berners-Lee T, Hendler J, Lassila O, others. The semantic web. Scientific American. 2001;284(5):28–37.
[9] Studer R, Benjamins VR, Fensel D. Knowledge engineering: Principles and methods. Data & Knowledge Engineering. 1998 Mar 1;25(1):161–97.
[10] Salas-Zárate M del P, Valencia-García R, Ruiz-Martínez A, Colomo-Palacios R. Feature-based opinion mining in financial news: An ontology-driven approach. Journal of Information Science. 2016 May 11;165551516645528.
[11] Paredes-Valverde MA, Valencia-García R, Rodríguez-García MÁ, Colomo-Palacios R, Alor-Hernández G. A semantic-based approach for querying linked data using natural language. Journal of Information Science. 2015 Nov 19;165551515616311.
[12] Prieto-González L, Stantchev V, Colomo-Palacios R. Applications of Ontologies in Knowledge Representation of Human Perception. International Journal of Metadata, Semantics and Ontologies. 2014 Feb;9(1):74–80.
[13] Rodríguez-García MÁ, Valencia-García R, García-Sánchez F, Samper-Zapater JJ. Ontology-based annotation and retrieval of services in the cloud. Knowledge-Based Systems. 2014 january; 56:15–25.
[14] Otero LD, Centeno G, Ruiz-Torres AJ, Otero CE. A systematic approach for resource allocation in software projects. Computers & Industrial Engineering. 2009;56(4):1333–1339.
[15] Ngo-The A, Ruhe G. A systematic approach for solving the wicked problem of software release planning. Soft Computing. 2008 Jan 1;12(1):95–108.
[16] Chen WN, Zhang J. Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler. IEEE Transactions on Software Engineering. 2013 Jan;39(1):1–17.
[17] e Silva LC, Costa APCS. Decision model for allocating human resources in information system projects. International Journal of Project Management. 2013 Jan;31(1):100–8.
[18] Heimerl C, Kolisch R. Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets. International Journal of Production Research. 2010 Jul 1;48(13):3759–81.
[19] Tsai H-T, Moskowitz H, Lee L-H. Human resource selection for software development projects using Taguchi's parameter design. European Journal of Operational Research. 2003 Nov 16;151(1):167–80.
[20] Zhu H, Zhou M, Seguin P. Supporting Software Development with Roles. IEEE Transactions on Systems Man and Cybernetics. 2006 Nov;36(6):1110–23.
[21] Carvalho Schnaider LR. Planejamento da Alocação de Recursos Humanos em Ambientes de Desenvolvimento de Software Orientados à Organização [PhD Thesis]. [Brazil]: Universidade Federal do Rio de Janeiro; 2003.
[22] IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998. 1998 Oct;1–40.
[23] IEEE Guide for Developing System Requirements Specifications. 1998 Ed IEEE Std 1233. 1998 Dec;1–36.
[24] Soares HA, Moura RS. A methodology to guide writing Software Requirements Specification document. In: 2015 Latin American Computing Conference (CLEI). 2015. p. 1–11.
[25] Bojārs U, Breslin JG. ResumeRDF: Expressing skill information on the Semantic Web. In: 1st International ExpertFinder Workshop. 2007.
[26] Brickley D, Miller L. FOAF vocabulary specification 0.98. Namespace Doc. 2012;9.
[27] Grau BC, Horrocks I, Motik B, Parsia B, Patel-Schneider P, Sattler U. OWL 2: The next step for OWL. Journal of Web Semantics: Science, Services and Agents on the World Wide Web. 2008 Nov;6(4):309–22.
[28] Musen MA. The Protégé Project: A Look Back and a Look Forward. AI Matters. 2015 Jun;1(4):4–12.
[29] Rodríguez-García MÁ, Valencia-García R, García-Sánchez F. An Ontology Evolution-Based Framework for Semantic Information Retrieval. In: Herrero P, Panetto H, Meersman R, Dillon T, editors. On the Move to Meaningful Internet Systems: OTM 2012 Workshops [Internet]. Springer Berlin Heidelberg; 2012 [cited 2016 Dec 12]. p. 163–72. (Lecture Notes in Computer Science). Available from: http://link.springer.com/chapter/10.1007/978-3-642-33618-8_25
[30] Andrews P, Zaihrayeu I, Pane J. A classification of semantic annotation systems. Semantic Web. 2012;3(3):223–248.
[31] Cunningham H, Tablan V, Roberts A, Bontcheva K. Getting more out of biomedical documents with GATE's full lifecycle open source text analytics. PLoS Computational Biology. 2013;9(2), e1002854.
[32] Castells P, Fernandez M, Vallet D. An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. IEEE Trans Knowl Data Eng. 2007 Feb;19(2):261–72.
[33] Salton G, McGill MJ. Introduction to Modern Information Retrieval. New York, NY, USA: McGraw-Hill, Inc.; 1986.
[34] Skiena S. Dijkstra's algorithm. Implement Discrete Math Comb Graph Theory Math Read MA Addison-Wesley. 1990;225–227.
[35] Zobel J, Moffat A. Exploring the Similarity Space. SIGIR Forum. 1998 Apr;32(1):18–34.
[36] Mendenhall WM, Sincich TL, Boudreau NS. Statistics for Engineering and the Sciences. CRC Press; 2016.
[37] Virtuoso O. Universal server platform for the real-time enterprise. 2009.

[38] Hripcsak G, Rothschild AS. Agreement, the F-Measure, and Reliability in Information Retrieval. Journal of the American Medical Informatics Association. 2005 May 1;12(3):296–8.