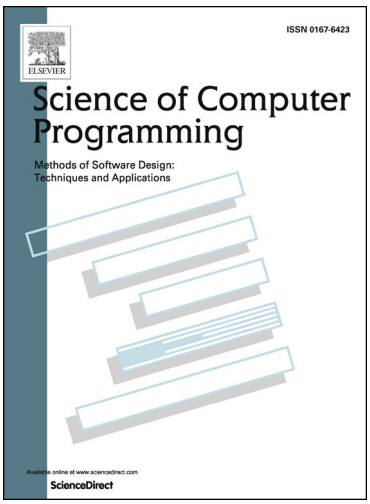
Accepted Manuscript



An ontology-based approach with which to assign human resources to Software projects

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

PII: S0167-6423(18)30004-2
DOI: https://doi.org/10.1016/j.scico.2018.01.003
Reference: SCICO 2184

To appear in: *Science of Computer Programming*

Received date: 15 December 2016
Revised date: 20 October 2017
Accepted date: 10 January 2018

Please cite this article in press as: M.A. Paredes-Valverde et al., An ontology-based approach with which to assign human resources to Software projects, *Sci. Comput. Program.* (2018), https://doi.org/10.1016/j.scico.2018.01.003

Highlights

- We present an ontology-based system to personnel assignment to software projects.
- We propose an ontology to describe personnel competences.
- We present an ontology to describe the software development technologies domain.
- We have conducted an evaluation in a software development organization.

An ontology-based approach with which to assign human resources to Software projects

Mario Andrés Paredes-Valverde[1], María del Pilar Salas-Zárate[1], Ricardo Colomo-Palacios[2], Juan Miguel Gómez-Berbís[3], Rafael Valencia-García[1]
[1]*Departamento de Informática y Sistemas, Universidad de Murcia, Campus de Espinardo 30100, Murcia, Spain*
[2]*Computer Science Department, Østfold University College, Norway*
[3]*Computer ScienceDepartment, Universidad Carlos III de Madrid, Madrid, Spain*
marioandres.paredes@um.es, mariapilar.salas@um.es, ricardo.colomo-palacios@hiof.no, juanmiguel.gomez@uc3m.es, valencia@um.es

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 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 their personnel. The term competence refers to the state or quality of being adequately or well qualified [1]. Human resources play a critical role in the success of software projects [2], [3] and ensuring their correct assignment to a specific software project is, therefore, an immediate requirement to which software development organizations must pay special attention. Software development teams are currently formed on the basis of human resources managers' experience of 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 large organizations and SMEs (Small and Medium-sized Enterprises) owing to the number of employees that are available.

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 the customer's requirements [5]. The 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 concerning what the software product will, and, if necessary, will not do [6]. The SRS document enlists sufficient and necessary requirements that are required for the project to be developed [7]. This document can be used as a basis on which to determine the technological knowledge and skills that a person must have in order to be integrated into the development team in charge of a project.

The objective of the Semantic Web is to provide Web information with a well-defined meaning and make it understandable to not only humans but also computers[8]. The ontologies in the Semantic Web 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 representation of knowledge, with the advantage of it being reusable and shareable. This knowledge is mainly formalized through the use of five kinds of components, namely classes, relations, attributes, axioms, and instances. 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 that is capable of automatically suggesting the human resources who are best suited to participating in a new software development project. On the one hand, this system uses ontologies to model the domain in which it will be implemented, thus allowing both the software requirements specification and the personnel's competencies to be formalized. More specifically, ontologies are used to formalize SRS (Software Requirements Specification) documents and personnel competencies as regards markets, technologies and language skills. On the other hand, the proposed system recommends those personnel that best fit with a new software development project on the basis of the semantic similarity among its SRS document and the SRS documents of previous projects. This similarity is computed by means of a semantic indexing approach.

The remainder of this paper is organized as follows. Section 2 outlines the state of the art of the technologies involved in this work. The components involved 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, our conclusions are shown in Section 5.

2. Related work

Several prominent efforts by which to provide systems that will assist in the process of assigning human resources to software projects currently exist. For instance, [4] presents a formal model with which to assign human resources to software projects that is based on the Delphi method, an expert consultation method whose essence is that of organizing an anonymous dialogue among experts consulted individually by means of questionnaires. This method is applied in order to determine the criteria that will be employed to select experts and to draft a proposal of the fixed roles and competences required to tackle software projects. The objective of the Skillrank approach [14] is, meanwhile, to measure users' expertise by analysing their profiles and activities in social networks. The system then uses this information as a basis to detect, verify, and rank experts using an appropriate trust metric. The work presented in [15] proposes a methodology that can be used to assign resources to tasks when optimum skill sets are not available. This methodology considers the candidates' existing capabilities, the levels of expertise required, and the priorities of the skills required for the task. In [16], the authors present a release planning method with which to develop software in an incremental manner. This method assigns human resources on the basis of their productivity as regards executing different kinds of tasks. However, the method is only applicable in

mature software organizations that have well-defined and measured processes. Another work [17] proposes a flexible and effective model for software project planning that is based on an event-based scheduler (EBS) and an ant colony optimization (ACO) algorithm. The proposed model represents a plan by means of a plan list and a planned employee allocation matrix in order to consider task scheduling and employee allocation. In [18], the authors present I-Competere, a tool whose purpose is to forecast competence gaps in key management personnel by predicting planning and scheduling competence levels. This tool makes it possible to forecast and anticipate a competence need, thus articulating personnel development and techniques. The work presented in [19] proposes a methodology based on dynamic programming with which to assign human resources to software development projects. This methodology considers the complexity of each project, the staff's existing capabilities and the skills required for the project. In [20], the authors present an optimisation model that can be used 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 [21], the authors develop a method whose purpose is to select the proper set of human resources with which to form a software development team in order to complete the project at a 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 (which were estimated as the average numbers of software lines of code per day) and the candidates' salaries. Furthermore, there are works that focus on the topic of roles. For example, in [22] the authors state the importance or roles in the assignment of human resources. They propose a role-based software engineering process using primitive roles. They additionally develop a prototype tool that can be employed to dynamically add, specify and modify roles. In the same context, the author of [23] proposes a human resources management process for software development projects based on the reuse of organizational knowledge regarding competences and previous role assignments. According to this process, the project leaders assign people to each project task by considering the profile defined.

The works above propose a variety of solutions to the human resources assignment problem. The common aspect of all these works is that they in some respect consider the employees' competencies in the assignment of human resources. However, none of them consider the employees' previous participation in similar software projects. The research effort presented in this work provides a decision support system that aims to assist human resources managers to assign those workers who are best suited to participating in a new software development project. This system has been designed to take advantage of well-known XML vocabularies so as to describe both the competences of an organization's personnel and their experience on previous software projects. Furthermore, our approach integrates semantic technologies, and more specifically ontologies, in order to relate the information resources used by the system (personnel profile and SRS documents) and can then suggest personnel on the basis of their previous participation in similar projects. A detailed description of the architecture design and component functionality of the present approach is provided in the following sections.

3. System architecture

The main goal of the decision support system proposed herein is to assist in the assignment of human resources to new software development projects. The architecture of this system relies on four main elements: information resources (SRS documents and employees' profiles), an ontology repository, a semantic indexing module, and the decision support module. Figure 1 shows the

complete architecture of the system. The work flow of the system is summarised in the nine phases shown below.

1. SRS documents list the requirements of a software development project and help to determine the technological knowledge and skills that employees must have to be involved in a software development project. In this respect, all the SRS documents from previous software development projects are semantically annotated in accordance with ontologies stored in an ontology repository. More specifically, this process uses an ontology that models the software development technology domain (see Section 3.2.1), including all those tools and technologies that support all the phases of the software development lifecycle. The objective of the semantic annotation process is to discover semantic similarities between software projects and the personnel's profile.
2. The employees' profiles are formalized by means of an ontology that helps to describe the employees' competences, emphasizing their knowledge and skills as regards software development technologies. This task uses the same ontology used in Step 1. The employees' profiles also include information about the employees' experience on previous software projects within the organization. It must be emphasized that the assignment of employees to previous projects was performed by the organization's human resources managers and projects leaders. The system proposed in this work takes advantages of that knowledge and formalizes it in such a way that it can be reused and shared by the entire organization. Ontologies were, therefore, chosen for this purpose since they help to provide a formal and explicit specification of a shared conceptualization.
3. Once the SRS documents have been semantically annotated, the semantic indexing module assigns a weight to each annotation to reflect how relevant the ontological entity is as regards the meaning of the SRS document. These weights are computed by using an adaptation of the TF-IDF algorithm based on the occurrences of the ontological entity in each SRS document. The semantic meaning of the SRS documents is additionally enriched by considering ontological entities that have a taxonomic relation with the entities that appear explicitly. At the end of this process, each SRS document is represented by means of a vector space model.
4. When a new software development project is established, the first step consists of generating the SRS document of the project according to requirements specification standards. The semantic indexing module then generates a vector representation of the SRS document.
5. The project similarity module computes the similarity level among the new project and previous ones. This similarity level is calculated by comparing their vectors in the space through the use of the cosine metric. At the end of this phase, the project similarity module generates a matrix containing the similarity rates among all projects, including the new project.
6. The HR (Human Resources) suggesting module obtains the most similar projects from the matrix based on a threshold, and more specifically, the statistical percentile method. The percentile indicates the value of a variable (similarity rate) below which a given percentage of observations in a group of observations fall.
7. Once the set of similar projects has been obtained, the HR suggesting module obtains all the people involved in those projects. Please recall that employees' profiles, which include the employees' participation in previous projects, have been described by means of an ontology and stored in the ontology repository (Step 2). This information is, therefore, obtained from this repository by means of SPARQL queries.

8. The HR suggesting module assigns a score to each employee by adding up the similarity rates of the most similar projects in which he/she participated. The most similar projects were obtained in Step 6.
9. Finally, the HR suggesting module sorts all the employees by their score. The sorted list of employees is then sent to the human resources managers or project leaders, who must select which employees will be involved in the new project.

A detailed description of each component of the architecture, along with its principal contribution, is provided in the following sections.

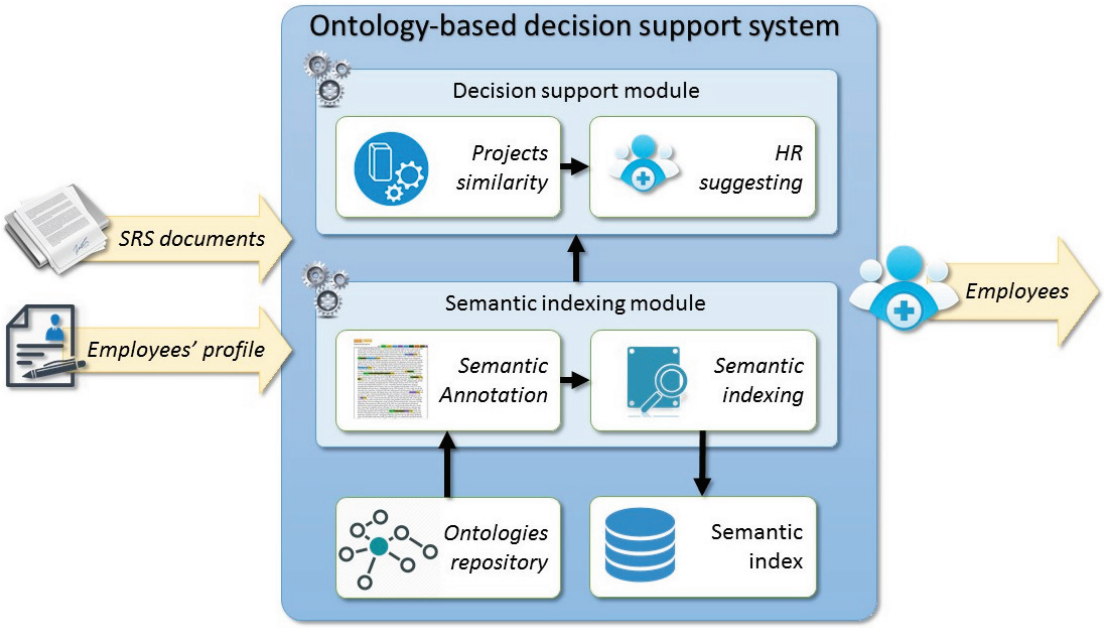


Figure 1.System architecture.

3.1 Information resources

The information resources used by our system are the SRS documents and the semantic profile of each person that could have the responsibility of executing work units throughout the software development process. The SRS documents used in this work are based on requirements specification standards, specifically IEEE [24], [25]. With regard to the employees' profile, we have established an ontology with which to describe it, stressing the employees' knowledge and skills as regards software development technologies, along with 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 by means of documents organized according to the IEEE Recommended Practice for Software Requirements Specifications [24] and the IEEE Guide for Developing System Requirements Specifications [25]. The SRS document used in this work consists of several sections. However, a general description of the five main sections is presented below:

1. Introduction. This section should provide an overview of the entire SRS. It must also identify the software product by name, explain what the software product will, and, if necessary, will not do, and 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 requirements to a level of detail sufficient to enable designers to design a system that will satisfy those requirements, and a 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 the 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 in such a way that all the participants can understand it. Furthermore, it is the basis for all other development activities and its quality is fundamental for the success of the project [26]. Based on this understanding, we believe that this document is crucial as regards determining the human resources who are the best suited to participating in a specific software development project, since it describes all requirements through the use of concepts that refer to technologies and markets in which the organization's employees must be competent.

3.1.2. Employees' profile

The present approach also needs to know the personnel's competences. In this respect, we propose an ontology with which to describe the employees' competences, emphasizing their knowledge and skills regarding software development technologies in addition to their experience on previous software projects. This ontology considers not only the employee's personal information but also her/his technological skills, business markets skills, language skills and experience on previous software projects within a specific organization. The main objective of this ontology is to improve the semantic description of employees' competences within the context of software development. Figure 3 shows an excerpt of the employees' competences ontology proposed in this study. In summary, this ontology helps to describe personal information such as name, surname and marital status, among others, along with her/his language, technology and markets skills, education, courses, the projects in which he/she has been involved and the roles he/she has played.

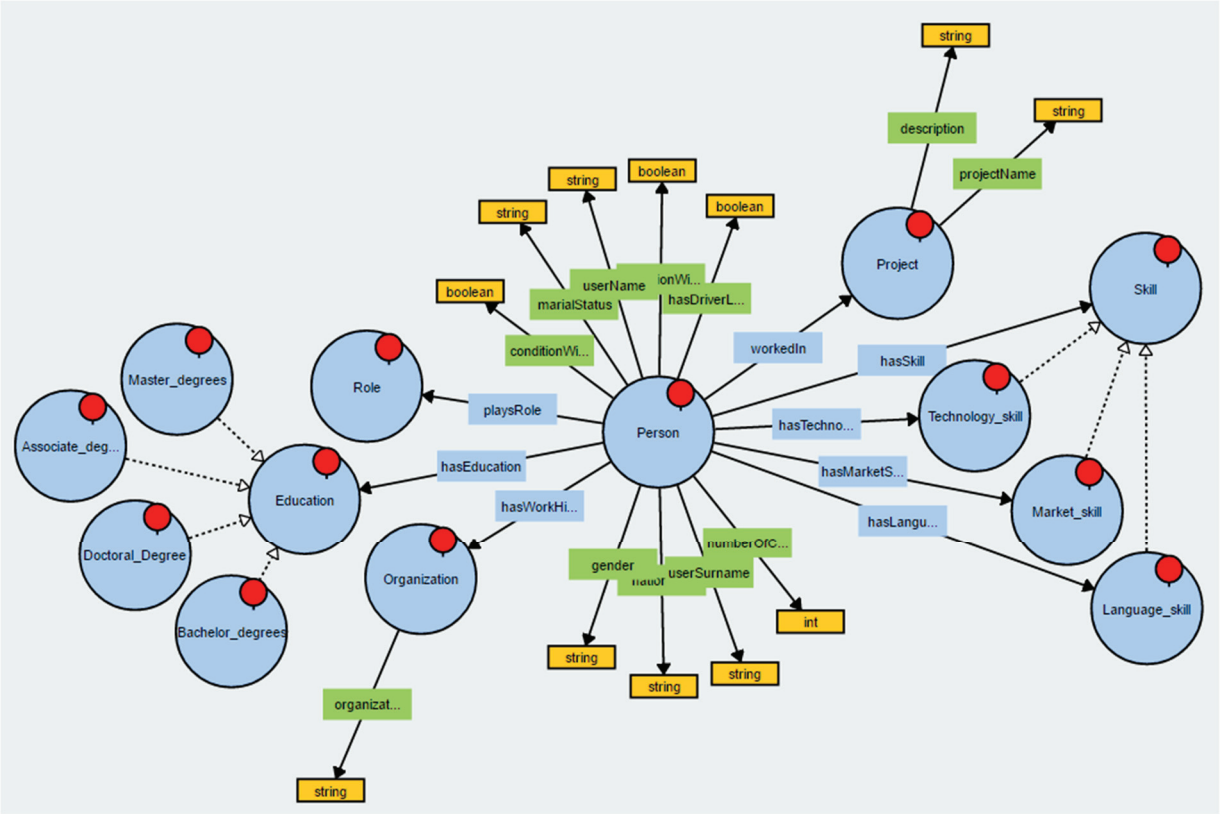


Figure 2. Excerpt of the employees' competence ontology.

3.2. Ontology repository

The decision support system proposed in this work requires a formal representation of the domain in which it will be implemented. In this respect, the ontology repository stores all the ontological models that will be used by the semantic annotation module to perform its corresponding tasks. In order to achieve the main goal of this work, an ontology describing software development technologies domain has been designed and implemented using the OWL 2 Web Ontology Language [27] and the Protégé software [28]. This ontology is described below.

3.2.1. Software development technology ontology

This ontology provides a common vocabulary for the software development technology domain. This domain includes all the tools and technologies that provide all phases of the software development life cycle with support, such as DBMS (Database Management Systems), programming languages, web design tools, modelling languages, Cloud platforms, web development frameworks, mobile frameworks, among others. 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 edited content, i.e. it is written collaboratively by the people who use it. We have performed a concept extraction process by taking advantage of the categorization function in Wikipedia. This process resulted in an ontology consisting of 2134 concepts that form a hierarchy that can be used by different kinds of applications or tasks. The terms contained in this ontology enable it to describe software development industry-needed skills throughout different areas such as software engineering, web programming, databases, mobile development, to name but a few. Figure 3 shows an excerpt of the software development technology ontology, along with some of the generic terms contained within it.

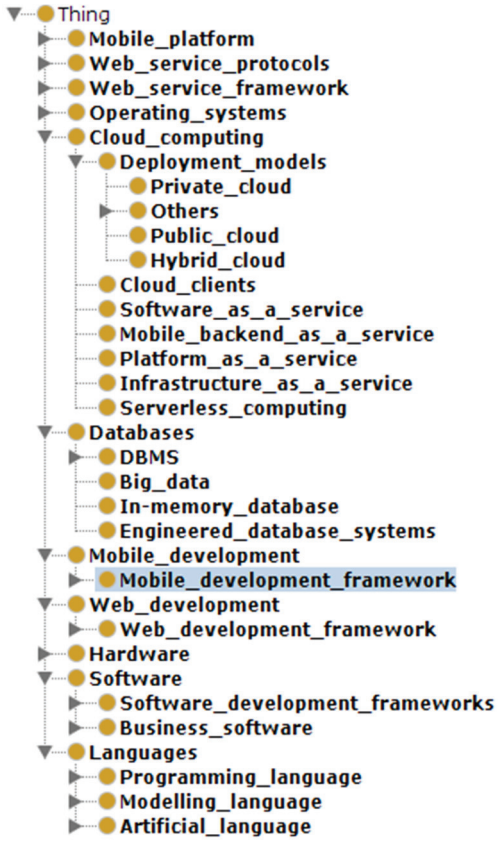


Figure 3. Excerpt of the software development technology ontology.

3.3. Semantic indexing module

3.3.1. Semantic annotation

The information resources on which the semantic indexing module relies are the SRS documents and the personnel profiles. Firstly, the SRS documents are processed by means of natural language processing (NLP) techniques in order to obtain a set of semantic annotations in accordance with the ontologies stored in the ontology 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 regard to a formal conceptual model [30], in this case, the software development technology ontology. This alignment will allow the decision support module to discover semantic similarities between the software projects, and also between these projects and the personnel's profiles. The semantic annotation process is performed using the GATE framework [31], an infrastructure whose purpose is to develop and deploy software components that process human language.

3.3.2. Semantic indexing

Once the information resources have been semantically annotated, this module assigns a weight to the annotations with the objective of reflecting how relevant the ontological entity is as regards the meaning of the SRS document. The weight assignment is fundamental as regards computing semantic relatedness among the information resources involved in this work, thus allowing the decision support module to suggest personnel who 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] and on the basis of 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.

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

where $n_{i,p}$ is the number of occurrences of 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 annotation weighting process used until now considers only those ontological entities that explicitly appear in the SRS documents. In order 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. This extension has been implemented using Dijkstra's algorithm [34] which makes it possible to find the shortest paths between the nodes in a graph. In this case, Dijkstra's algorithm is used to determine the distance between two classes of the ontology on the basis of their taxonomic relations. This module, therefore, also assigns a weight to those ontological concepts whose distance with regard to the annotated concept is less than 2. More specifically, the weight of this kind of annotations is computed using equation 2.

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

where *dist(i,j)* is the semantic distance between concept *i* and concept *j* in the domain ontology. As mentioned earlier, the distance is calculated by considering the taxonomic relations between concepts, i.e. the subclass of relations. The distance between a concept and itself is, therefore 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 that should be selected for a new software project by considering the semantic relatedness among them. When a new software project is created it is, therefore, necessary to establish its requirements by means of the SRS document. Once this resource has been created, the semantic indexing module processes the SRS document in order to obtain a vector representation of it by means of equation 2. This vector then enables this module to suggest which human resources should participate in that project. To this end, this module needs to rank all the employees existing in the knowledge base in order to discover 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. These tasks are explained in detail as follows.

3.4.1. Project similarity

It is possible that a great number of software projects within an organization might be focused on the same topics and may, therefore have a certain degree of similarity. The aim of this module is to employ the aforementioned fact as a basis on which to compute the level of similarity between the new project and 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. The semantic similarity of software projects is, therefore, calculated by comparing their vectors in the space through the use of 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. Please recall that these vectors are calculated by using equation 2, although semantic similarity is computed using equation 3.

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

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 symbol θ is the angle that separates both vectors and 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 greater angle between them.

As will be noted, equation 3 helps to determine the similarity between two software projects. However, it is necessary to obtain the similarity level between all the projects stored in the system. In order to achieve this objective, the aforementioned module obtains all these similarity rates and stores them in a matrix denominated as the Project2Project matrix. Figure 4 provides a representation of this matrix, in which 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 project similarity matrix.

3.4.2. HR suggesting

The second task performed by the decision support module consists of suggesting people who could be involved in the new software project in accordance with their experience on previous projects. This module therefore obtains, from the Project2Project matrix, the most similar projects to the new one based on a threshold. This threshold is calculated using the statistical percentile method (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.

The selection of the most similar projects is still hard work, since there are sometimes a lot of projects that share many technologies and markets, and there are many employees with the knowledge and experience required to work on that project. In this respect, the system proposed herein implements a configurable mechanism that allows human resources managers and project leaders to select the threshold that best fits with their needs on the basis of a set of experiments performed before its implementation within the organization. For instance, the HR suggesting module can estimate the threshold value as the similarity rate that separates 80% of the levels of similarity, represented by P80, i.e. the similarity rate that separates 80% of the software projects, leaving the other 20%, which represents meaningful similar projects. The percentile is calculated by using equation 4, which was presented in [36].

$$k = \frac{p\ x\ n}{100} \qquad (4)$$

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 percentile p, p represents the percentile we are seeking and *n* represents the sample size.

Once the set of similar software projects has been obtained, it is necessary to obtain all the participants on those projects, and the employees' profiles are used for this purpose. As mentioned previously, these profiles describe the employees' knowledge and skills as regards software development technologies, in addition to their experience on previous software projects. However, the HR suggesting module considers only the employees' experience on similar projects. Even though the current version of the system proposed in this work does not use the semantic description of employees as regards their knowledge and skills to assign human resources to projects, the main objective of this representation is to attain a semantic perspective of all the organization's knowledge concerning software projects that can be reused and shared in future implementations. Finally, since the employees' profiles are stored in an ontology repository, this task is performed using SPARQL queries.

Having obtained all the people involved in all similar projects, the system now needs to associate a score with each person with the objective of determining the personnel that best fit with the new software project. Each person's score is, therefore, obtained by adding up 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.

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

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, along with the roles that he/she has played in those projects. In order to prove the effectiveness of the approach presented in this work, we conducted an evaluation, which will be explained in the following section.

4. Case study

4.1. Method

In this work, we have evaluated our proposal with the objective of measuring its effectiveness regarding the software project similarity measurement and the assignment of human resources to Software projects. To this end, our system was implemented in a software development SME which provides solutions and services for different markets, such as public administration, healthcare, finance, among others. This organization is characterized by its 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.

1. Domain modelling. In order to obtain the semantic meaning of all the information resources to be used by the system, it is necessary to have a formal representation of the domain in which it will be implemented. Considering that a great number of projects developed by the SME under consideration rely on the software development technology domain, we have used the ontology presented in section 3.2.1.
2. SRS information. The software development processes followed by the organization are attached to international standards which allow the organization to provide its customers with high-quality software products and services. Of the standards used by this organization are the IEEE recommendations for requirements specification. The SRS of each project, therefore, 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 the information related to different kinds of requirements, such as functional requirements, software system attributes and external interface requirements, to name but a few. Around 50 SRS documents within the software development technology domain were processed by using the semantic indexing module to represent each of them through vectors, in addition to the creation of the Project2Project matrix, which contains the similarity rates among all the projects.
3. Personnel profile. Information related to the personnel's experience on previous software projects within the organization was represented by means of the ontology presented in section 3.1.2. Moreover, 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 personnel profiles were introduced into the system.
4. Experiment. Once all information resources had been introduced, the experiment took place. The following activities were, therefore, performed.
    a. Ten new software projects within the contexts of finance (3), e-learning (3), e-health (2) and social networks (2) were proposed and described by means of the SRS document template.
    b. The project leaders manually selected, for each new project, the most closely related projects. It is important to mention that project similarity depends not only on the main context (in this case, finance, e-learning, e-health and social networks), but also on the technologies and tools established in the SRS document. There was no a limit to the number of projects that the project leaders could select.
    c. The organization's human resources managers were asked to select, for each new project, the personnel that best fitted with the project. The number of people assigned to each project depended on its size. According to the experts the number of people in their organization who are assigned to a small-sized project does not exceed 7, while for medium-sized projects the number varies between 8 and 15, and for large-sized projects the number of employees involved in the project exceeds 15.

d. Finally, the set of selected projects and the personnel suggested by the organization's experts were compared to the results obtained automatically by our system. It is worth noting that the experiments were performed with different threshold configurations, more concretely P60, P80 and P90, respectively. With regard to project similarity, the metric precision, recall and F-measure were calculated. With regard to the assignment of human resources, only the precision was computed because the system was configured to obtain the same number of employees as those selected by the experts.

As mentioned previously, in this evaluation, we used the primary metric precision, recall and its harmonic mean, known as the F-measure. These alternative inter-rater agreement metrics have been applied by researchers in the context of information retrieval [38].

$$precision = \frac{correct\ suggestions}{total\ suggestions} \quad (6)$$

$$recall = \frac{correct\ suggestions}{number\ of\ relevant\ items} \quad (7)$$

$$F - measure = 2 * \frac{precision * recall}{precision + recall} \quad (8)$$

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 organization's experts. Finally, the F-measure score is the weighted average of the precision and recall. The F-measure score attains its best value at 1 and its worst score at 0.

4.2. Discussion

The global experiment results for software project similarity and human resources assignment are reported in the following tables. With regards to project similarity, it should be mentioned that the experiments described in section 4.1 were performed with three different threshold configurations (P60, P80 and P90). More concretely, Table 1, Table 2 and Table 3 show the results obtained for software project similarity with the thresholds P60, P80 and P90, respectively.

As will be noted, the threshold with which the best evaluation results were obtained was P80 with an F-measure of 0.7588, since the F-measures obtained for P60 and P90 are 0.7131 and 0.6971, respectively. Although the precision results for P60 are better with an average precision of 0.7594, the recall decreases significantly to 0.6754. These results signify that with a threshold of 60, fewer projects are selected, and although the most important projects are correctly suggested, there are some important projects that are not suggested by the system. On the other hand, the results obtained for P90 are better as regards the recall score 0.8437, but the precision decreases drastically to 0.5950. This is because more projects are selected and almost none of the last projects selected are good candidates.

Although the best results are obtained with threshold P80 in the case of this experiment (see Table 2), it must be emphasized that the threshold may vary according to both the complexity of the

software projects and the employees' existing capabilities, along with the expertise and skills required by the project. Here, it can be observed that the best result was obtained for project 9, with an F-measure score of 0.8235. The worst result was, meanwhile, obtained for the project 7, again with an F-measure score of 0.6316.

Table 1. Evaluation results for project similarity with P60.

Project	Projects selected by experts	Projects selected by the system	Projects correctly suggested	Precision	Recall	F-measure
1	15	13	10	0.7692	0.6667	0.7143
2	17	18	13	0.7222	0.7647	0.7429
3	14	11	10	0.9091	0.7143	0.8000
4	14	13	10	0.7692	0.7143	0.7407
5	16	15	12	0.8000	0.7500	0.7742
6	13	12	9	0.7500	0.6923	0.7200
7	8	8	5	0.6250	0.6250	0.6250
8	9	8	5	0.6250	0.5556	0.5882
9	16	13	11	0.8462	0.6875	0.7586
10	12	9	7	0.7778	0.5833	0.6667
Average				0.7594	0.6754	0.7131

Table 2. Evaluation results for project similarity with P80.

Project	Projects selected by experts	Projects selected by the system	Projects correctly suggested	Precision	Recall	F-measure
1	15	17	12	0.7059	0.8000	0.7500
2	17	21	15	0.7143	0.8824	0.7895
3	14	16	12	0.7500	0.8571	0.8000
4	14	17	12	0.7059	0.8571	0.7742
5	16	18	13	0.7222	0.8125	0.7647
6	13	15	11	0.7333	0.8462	0.7857
7	8	11	6	0.5455	0.7500	0.6316
8	9	11	7	0.6364	0.7778	0.7000
9	16	18	14	0.7778	0.8750	0.8235
10	12	14	10	0.7143	0.8333	0.7692
Average				0.7005	0.8291	0.7588

Table 3. Evaluation results for project similarity with P90.

Project	Projects selected by experts	Projects selected by the system	Projects correctly suggested	Precision	Recall	F-measure
1	15	20	12	0.6000	0.8000	0.6857
2	17	23	15	0.6522	0.8824	0.7500
3	14	20	12	0.6000	0.8571	0.7059
4	14	18	12	0.6667	0.8571	0.7500
5	16	22	13	0.5909	0.8125	0.6842
6	13	20	11	0.5500	0.8462	0.6667
7	8	13	6	0.4615	0.7500	0.5714

8	9	14	7	0.5000	0.7778	0.6087
9	16	22	15	0.6818	0.9375	0.7895
10	12	17	11	0.6471	0.9167	0.7586
Average				0.5950	0.8437	0.6971

The global experiment results for the assignment of human resources are shown in Table 4, Table 5 and Table 6 for P60, P80 and P90, respectively. As can be observed, the best results are obtained for thresholds P80 and P90 with a precision of 0.7382. The results of the precision obtained for P60 (0.6127) are worse owing to the fact that the system obtains fewer projects in order to calculate the *peopleScore* function (see Equation 5) and fewer candidate employees are, therefore, assigned by the system. However, with a threshold of P90, more projects are suggested by the system and they are used in order to calculate the *peopleScore* function, but there are no significant differences with regard to P80.

When using P80 and P90, the approach obtained an average rate of 0.7382 for the precision metric. The best result was, again, obtained for projects 2 and 9, with a precision score of 0.8333. The worst result was, meanwhile, obtained for project 7, with a precision score of 0.6429.

Table 4. Evaluation results for human resources assignment with P60.

Project	Topic	Size	Employees assigned by experts	Employees correctly assigned by the system	Precision
1	Finance	Medium	11	6	0.5455
2	Finance	Medium	12	7	0.5833
3	Finance	Big	17	10	0.5882
4	E-learning	Small	7	4	0.5714
5	E-learning	Medium	12	7	0.5833
6	E-learning	Medium	10	6	0.6000
7	E-Health	Medium	14	8	0.5714
8	E-Health	Medium	12	8	0.6667
9	Social networks	Small	6	4	0.6667
10	Social networks	Medium	12	9	0.7500
Average					0.6127

Table 5. Evaluation results for human resources assignment with P80.

Project	Topic	Size	Employees assigned by experts	Employees correctly assigned by the system	Precision
1	Finance	Medium	11	8	0.7273
2	Finance	Medium	12	10	0.8333
3	Finance	Big	17	13	0.7647
4	E-learning	Small	7	5	0.7143
5	E-learning	Medium	12	9	0.7500
6	E-learning	Medium	10	7	0.7000
7	E-Health	Medium	14	9	0.6429
8	E-Health	Medium	12	8	0.6667
9	Social networks	Small	6	5	0.8333
10	Social networks	Medium	12	9	0.7500

Average	0.7382

Table 6. Evaluation results for human resources assignment with P90.

Project	Topic	Size	Employees assigned by experts	Employees correctly assigned by the system	Precision
1	Finance	Medium	11	8	0.7273
2	Finance	Medium	12	10	0.8333
3	Finance	Big	17	13	0.7647
4	E-learning	Small	7	5	0.7143
5	E-learning	Medium	12	9	0.7500
6	E-learning	Medium	10	7	0.7000
7	E-Health	Medium	14	9	0.6429
8	E-Health	Medium	12	8	0.6667
9	Social networks	Small	6	5	0.8333
10	Social networks	Medium	12	9	0.7500
Average					0.7382

The results obtained by our approach seem promising. Furthermore, and as can be observed in Table 2 and Table 5, the best results are obtained with the threshold P80, and there is no great difference between the evaluation scores obtained for all new software projects as regards both aspects (projects similarity and personnel suggestion). As was described in previous sections, the personnel suggestion is based on the set of most similar projects, which are obtained by the system beforehand. This is confirmed by the results obtained, since we have observed a correlation between the scores obtained for project similarity and those obtained for personnel suggestion. For instance, we noted that projects 9 and 2, which have a higher number of similar projects, obtained the best precision concerning personnel suggestion. Meanwhile, projects 7 and 8, which have a lower number of similar projects, got the worst results as regards personnel suggestion. On the basis of this understanding, we can conclude that as the number of similar projects increases, the personnel suggestion is more precise. Furthermore, we have used a detailed analysis of all the software projects (both old and new) and the profile of each employee to ascribe the score variations to software projects focused on out-of-context topics.

With regard to out-of-context topics, even though our approach has an ontology that models the software development technology domain, a domain into which most organization projects fall, the SRS documents contain several concepts that are not described by this ontology. This is mainly because 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 8 obtained a low precision for project similarity suggestion because these works are focused on the development of solutions for e-health, a domain that was not described in depth by the ontology. This signified that their corresponding SRS documents were not correctly described, i.e. the number of annotations generated for each one was slightly low. Moreover, there are not a great number of similar projects in the system because this domain has recently been adopted by the organization. In this respect, 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 regard 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 that presented in [29].

Furthermore, with regard to personnel assignment, human resources managers provided suggestions about the global functionality of the system. One of the most outstanding observations concerns the inclusion of employees that have little experience in non-critical software projects. In this respect, 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 may emerge throughout the organization's software development process.

5. Conclusions

In this piece of research, we have presented a semantic-based system that is able to suggest which personnel could be incorporated into a new software project by considering their experience on previous projects. Our proposal obtained encouraging results with F-measure scores of 0.7131 (P60), 0.7588 (P80), and 0.6971 (P90) for similarity projects and precision scores of 0.6127 (P60) and 0.7382 (P80 and P90) for personnel suggestion. The main contribution of our research effort is twofold. First, we propose an ontology focused on describing the competences and experience of the personnel within an organization. Second, the integration of semantic techniques to the personnel assignment process within a software development organization can help the human resources manager to speed up this task, and even make it completely automated. And all of this in addition to a high level of confidence that the personnel selected by the system are the most appropriate for the project. Moreover, the method proposed herein can be applied to the organization's current software development process without any changes having to be made 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 that the system be provided with an ontology that models the corresponding context. It is important to stress that the incorporation of semantic technologies into 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 the semantic web, and more specifically of ontologies, is for them to serve as a reference for communication not only among humans but also among computers.

Since the similarity of projects is based on a semantic annotation, we plan to extend the ontology used in this work (the software development technology ontology) by taking advantage of current and proven methods, such as those presented in [39] or [40], to include more terms that will help to describe the projects in a better way, thus increasing the possibility of discovering similarities among software projects. We also plan to perform further experiments by considering more ontological entities that do not explicitly appear in the SRS documents, i.e. by increasing the semantic distance between the ontological entity that explicitly appears and their subclasses. With regard to out-of-context projects, we plan to integrate new ontologies that will help the system to generate the corresponding semantic annotations. Considering these facts, it is important to pay particular attention to the complexity of the software projects that have been developed, in addition to new software projects, since this feature could influence the results obtained by the evaluation process.

Furthermore, one limitation of our approach is that the current version does not use the semantic description of employees regarding their knowledge, skills and other properties of the employees' competence ontology to assign human resources to projects. As future research, we have, therefore, considered developing a new method with which to calculate the *peopleScore* function by taking

into account this type of information in order to determine the personnel that best fit with a new software project.

Finally, as future work we also plan to prove the effectiveness of our approach by means of its implementation in other software development SMEs whose business markets do not fall into the software development technology domain. With regard to this plan, it will be necessary for the new organization to implement recommended practices for software requirements specifications, because we believe that a well-described SRS document is the most important element as regards determining the personnel that should be involved in a new project since it describes all the requirements by using concepts and relationships that are formalized through domain ontologies such as that used in this work.

Acknowledgements

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] E. H. Mifflin, *The American Heritage Dictionary*, 4th edition. Turtleback, 2001.

[2] N. Gorla and Y. W. Lam, "Who Should Work with Whom? Building Effective Software Project Teams," *Commun ACM*, vol. 47, no. 6, pp. 79–82, Jun. 2004.

[3] S. T. Acuña, M. Gómez, and N. Juristo, "How do personality, team processes and task characteristics relate to job satisfaction and software quality?" *Inf. Softw. Technol.*, vol. 51, no. 3, pp. 627–639, Mar. 2009.

[4] M. André, M. G. Baldoquín, and S. T. Acuña, "Formal model for assigning human resources to teams in software projects," *Inf. Softw. Technol.*, vol. 53, no. 3, pp. 259–275, Mar. 2011.

[5] R. Zeineddine and N. Mansour, "Software Quality Improvement Model for Small Organizations," in *Computer and Information Sciences - ISCIS 2003*, A. Yazıcı and C. Şener, Eds. Springer Berlin Heidelberg, 2003, pp. 1027–1034.

[6] I. C. S. S. E. S. Committee and I.-S. S. Board, "IEEE Recommended Practice for Software Requirements Specifications," 1998.

[7] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, Edición: 8. New York, NY: McGraw-Hill Education, 2014.

[8] T. Berners-Lee, J. Hendler, O. Lassila, and others, "The semantic web," *Sci. Am.*, vol. 284, no. 5, pp. 28–37, 2001.

[9] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data Knowl. Eng.*, vol. 25, no. 1, pp. 161–197, Mar. 1998.

[10] M. del P. Salas-Zárate, R. Valencia-García, A. Ruiz-Martínez, and R. Colomo-Palacios, "Feature-based opinion mining in financial news: An ontology-driven approach," *J. Inf. Sci.*, p. 165551516645528, May 2016.

[11] M. A. Paredes-Valverde, R. Valencia-García, M. Á. Rodríguez-García, R. Colomo-Palacios, and G. Alor-Hernández, "A semantic-based approach for querying linked data using natural language," *J. Inf. Sci.*, p. 165551515616311, Nov. 2015.

[12] L. Prieto-González, V. Stantchev, and R. Colomo-Palacios, "Applications of Ontologies in Knowledge Representation of Human Perception," *Int J Metadata Semant Ontol.*, vol. 9, no. 1, pp. 74–80, Feb. 2014.

[13] M. Á. Rodríguez-García, R. Valencia-García, F. García-Sánchez, and J. J. Samper-Zapater, "Ontology-based annotation and retrieval of services in the cloud," *Knowl.-Based Syst.*, vol. 56, pp. 15–25, enero 2014.
[14] J. M. Álvarez-Rodríguez, R. Colomo-Palacios, and V. Stantchev, "Skillrank: towards a hybrid method to assess quality and confidence of professional skills in social networks," *Sci. Program.*, vol. 2015, p. 3, 2015.
[15] L. D. Otero, G. Centeno, A. J. Ruiz-Torres, and C. E. Otero, "A systematic approach for resource allocation in software projects," *Comput. Ind. Eng.*, vol. 56, no. 4, pp. 1333–1339, 2009.
[16] A. Ngo-The and G. Ruhe, "A systematic approach for solving the wicked problem of software release planning," *Soft Comput.*, vol. 12, no. 1, pp. 95–108, Jan. 2008.
[17] W. N. Chen and J. Zhang, "Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler," *IEEE Trans. Softw. Eng.*, vol. 39, no. 1, pp. 1–17, Jan. 2013.
[18] R. Colomo-Palacios, I. González-Carrasco, J. L. López-Cuadrado, A. Trigo, and J. E. Varajao, "I-Competere: Using applied intelligence in search of competency gaps in software project managers," *Inf. Syst. Front.*, vol. 16, no. 4, pp. 607–625, Sep. 2014.
[19] L. C. e Silva and A. P. C. S. Costa, "Decision model for allocating human resources in information system projects," *Int. J. Proj. Manag.*, vol. 31, no. 1, pp. 100–108, Jan. 2013.
[20] C. Heimerl and R. Kolisch, "Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets," *Int. J. Prod. Res.*, vol. 48, no. 13, pp. 3759–3781, Jul. 2010.
[21] H.-T. Tsai, H. Moskowitz, and L.-H. Lee, "Human resource selection for software development projects using Taguchi's parameter design," *Eur. J. Oper. Res.*, vol. 151, no. 1, pp. 167–180, Nov. 2003.
[22] H. Zhu, M. Zhou, and P. Seguin, "Supporting Software Development With Roles," *IEEE Trans. Syst. Man Cybern. - Part Syst. Hum.*, vol. 36, no. 6, pp. 1110–1123, Nov. 2006.
[23] L. R. Carvalho Schnaider, "Planejamento da Alocação de Recursos Humanos em Ambientes de Desenvolvimento de Software Orientados à Organização," PhD Thesis, Universidade Federal do Rio de Janeiro, Brazil, 2003.
[24] "IEEE Recommended Practice for Software Requirements Specifications," *IEEE Std 830-1998*, pp. 1–40, Oct. 1998.
[25] "IEEE Guide for Developing System Requirements Specifications," *1998 Ed. IEEE Std 1233*, pp. 1–36, Dec. 1998.
[26] H. A. Soares and R. S. Moura, "A methodology to guide writing Software Requirements Specification document," in *2015 Latin American Computing Conference (CLEI)*, 2015, pp. 1–11.
[27] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, "OWL 2: The next step for OWL," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 6, no. 4, pp. 309–322, Nov. 2008.
[28] M. A. Musen, "The Protégé Project: A Look Back and a Look Forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, Jun. 2015.
[29] M. Á. Rodríguez-García, R. Valencia-García, and F. García-Sánchez, "An Ontology Evolution-Based Framework for Semantic Information Retrieval," in *On the Move to Meaningful Internet Systems: OTM 2012 Workshops*, P. Herrero, H. Panetto, R. Meersman, and T. Dillon, Eds. Springer Berlin Heidelberg, 2012, pp. 163–172.
[30] P. Andrews, I. Zaihrayeu, and J. Pane, "A classification of semantic annotation systems," *Semantic Web*, vol. 3, no. 3, pp. 223–248, 2012.

[31] H. Cunningham, V. Tablan, A. Roberts, and K. Bontcheva, "Getting more out of biomedical documents with GATE's full lifecycle open source text analytics," *PLoS Comput. Biol.*, vol. 9, no. 2, p. e1002854, 2013.

[32] P. Castells, M. Fernandez, and D. Vallet, "An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 2, pp. 261–272, Feb. 2007.

[33] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[34] S. Skiena, "Dijkstra's algorithm," *Implement. Discrete Math. Comb. Graph Theory Math. Read. MA Addison-Wesley*, pp. 225–227, 1990.

[35] J. Zobel and A. Moffat, "Exploring the Similarity Space," *SIGIR Forum*, vol. 32, no. 1, pp. 18–34, Apr. 1998.

[36] W. M. Mendenhall, T. L. Sincich, and N. S. Boudreau, *Statistics for Engineering and the Sciences*. CRC Press, 2016.

[37] O. Virtuoso, *Universal server platform for the real-time enterprise*. 2009.

[38] G. Hripcsak and A. S. Rothschild, "Agreement, the F-Measure, and Reliability in Information Retrieval," *J. Am. Med. Inform. Assoc.*, vol. 12, no. 3, pp. 296–298, May 2005.

[39] D. Zhang, Z. Yang, N. Wang, B. Wang, andH. Zhao, "Ontology extension based on axiomatic cognitive model for Ontology learning," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, pp. 825–829.

[40] I. Novalija, D. Mladenić, and L. Bradeško, "OntoPlus: Text-driven ontology extension using ontology content, structure and co-occurrence information," *Knowledge-Based Syst.*, vol. 24, no. 8, pp. 1261–1276, Dec. 2011.