IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 10, 2017, accepted November 17, 2017, date of publication December 7, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2778879

Ontology-Based Education/Industry Collaboration System

HOUSSEM GASMI [1,2] AND ABDELAZIZ BOURAS[3], (Member, IEEE)
[1]Engineering Research and Graduate Studies, College of Engineering, Qatar University, Doha, Qatar
[2]DISP Laboratory, Université Lumière of Lyon, 69007 Lyon, France
[3]Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar

Corresponding author: Houssem Gasmi (houssem.gasmi@qu.edu.qa)

This work was supported by the Qatar National Research Fund (a member of Qatar Foundation) NPRP under Grant NPRP 7-1883-5-289.

ABSTRACT As the main supplier of the workforce to the industry, higher education is increasingly criticized for not being abreast with the digital revolution and being disconnected from the industry. Competency-based education was developed to address this issue and bridge the gap between what the university is producing and the requirements of the industry. Hence, tools need to be developed that assists in the analysis process. This paper focuses on proposing a system that models the competencies required by occupations in the industry and higher education curricula and assists in matching profiles from the two domains. The different concepts in the domain are modeled as a semantic web ontology, and an inference engine performs the profile matching. In addition to the profile matching, the system calculates a score for the matching degree using the analytic hierarchy process (AHP) method.

INDEX TERMS Competency model, matching system, ontology, profile matching.

I. INTRODUCTION

The ever-increasing advances in the industry necessitate a close collaboration between the industry and universities. Higher education programs need to be kept up to date to cope with the industry's highly qualified labour needs to ensure students are better prepared for the needs of industry and increase their marketability [1]. Hence, the traditional goals assigned to universities and academics have evolved to not only produce and transfer scientific knowledge, but also to prepare students for the workplace and more broadly for their careers. Over the years, several countries, universities and other academic institutions have incorporated competency frameworks in their curricula design activities. Competency frameworks are one of the used educational tools to 'vocationalise' academic courses. Their main objective is to describe in detail the typical activities, knowledge, attitudes and skills required by the type of job for which the students are prepared. Competency-based education was initially developed in the United States during the seventies, and despite some objections, it spread to other Western countries. This approach was the result of the growing criticisms towards traditional education which was becoming more and more disconnected from the social evolutions of that time, especially changes within workplaces [2]. It has been used to reform upper-secondary vocational curricula and more recently it appears to be more and more used in higher education to update and reform academic courses [3], [4]. Several local initiatives have been proposed to formalise the definition of competencies such as O*NET in the United States [5] and 'ROME' in France [6].

This paper presents a profile matching application that models the domain knowledge as a semantic web ontology [7]. Ontologies have been used to solve issues faced in interoperability between different domains [8]. Because different parties usually have different concepts and needs, ontologies make reuse much easier by avoiding the wasted effort to translate terms, where the shared understanding of a term across a set of domains prevents from mismatched concepts and different definitions.

The objective of the proposed application is to establish an ontological relationship between the competency requirements of the occupations in the market and the higher education learners profile to ensure a continuous alignment between student profiles and the industry [9]. The educational and industry profiles are represented as O*NET competency framework profiles. The system consists of two major parts. The first is an ontology that models the domain concepts and performs the matching between profiles. The second is the service that measures the degree of matching between industry profiles and educational profiles using the Analytic

Hierarchy Process (AHP) method [10]. This paper describes the system architecture, the ontological model, and main design decisions of the proposed system.

The rest of the paper is organised as follows. In section II, we review related works. Section III provides an overview of the application architecture. The next section introduces the different types of stakeholders and their interactions with the application. Section V provides an overview of the ontology design methodology we followed. Section VI describes the ontology and its implementation. The next section explains the logic behind profile matching. Section VIII evaluates the ontology and discusses the different challenges we encountered to model the domain and implement the profile matching logic. The last section shows how the system calculates the matching score for a student profile and a university curriculum using the AHP method.

II. STATE OF THE ART

This section reviews the related work in profile matching and ontology mapping by selecting the papers that are more relevant to our domain of research. Many recruitment tools for measuring the suitability of candidates for a job have appeared in recent years. However, there has been little research investigating the semantic matching systems between the concrete needs of the industry and educational outcomes of universities. Thus, most of the developed tools suffer from the inadequate matching of candidates with job requirements. To the best of our knowledge, no previous published work has applied a profile matching system integrating semantic information related to the industry occupations and competencies generated by education curricula in order to evaluate the relevance between the profiles.

Several attempts were made to automate the recruitment process [11] using recommendation systems that match the job with the suitable candidate. The goal of these recruitment systems is to speed-up and increase the efficiency of the recruitment process. Several recommendation techniques have been applied such as content-based filtering, rule-based filtering, collaboration filtering and hybrid filtering [12]. These recommendation systems combine the data collected from the user with related data collected from other sources and generate a recommended list of items for the user. These systems find the match between people skills and job offer descriptions taking into consideration the preferences of the recruiters and the interests of the candidates. Some recommendation systems such as CASPER [13] are based on user interactivity and employ collaborative filtering techniques to offer recommendations for users based on the preferences of other users with similar profiles. The PROSPECT system is proposed in [14] to select candidates for recruitment. It extracts selection criteria from resumes such as education, competencies and the relevant experience in each. [15] suggests a standardised format for resumes to improve the effectiveness of candidates filtering and selection. A mobile-based recommendation system was proposed by [16] targeting career services in universities. It helps at matching recruiting companies with graduated students at low cost and focuses on two-sided profile matching based on preference lists for further recommendations. Several approaches are used by the recruitment systems like relevance feedback, natural language processing [17], semantic matching, and machine learning [18].

Domain ontologies have become a mainstream knowledge representation tool in many recent applications. Previous related decision-support works in the educational domain have used a context ontology–based approach [12], [19]. Recommendation systems have also used ontologies as a key part of their efficient filtering techniques [20]. In order to improve vocational education, this paper aims to present a semantic matching application based on ontological models that allow us to better capture, analyse and use relevant semantic information for the study and analysis of the gap between education and the industry [21]. Consequently, the proposed application helps universities to satisfy both companies and students.

III. SYSTEM ARCHITECTURE

The proposed application is a desktop application and has two main responsibilities. The first responsibility is matching between an education profile and an industry profile and stating whether the two profiles match. The education profile can be either a student transcript or a university curriculum. The structure of the two profiles is similar;, the only difference is that the student profile contains his/her grades which are used as an additional criterion in the matching score calculation. The industry profile represents an occupation we want to match against a student transcript or check whether a university curriculum fulfils the occupation requirements. The profiles are represented as instances of classes in the ontology which will be covered in more detail in a later section. The second responsibility of the application is calculating the matching score for matched profiles using the AHP method taking into account the matching criteria that is defined in the system. Fig. 1 depicts the profile matching application architecture. The application is comprised of four main components: the ontology, the inference engine, the Profile Matching Service (PMS), and the Matching Score Calculation Service (MSCS).

The application ontology defines the concepts in the domain and the relationship between them in addition to the matching rules. The ontology is designed to require defining the minimal information possible and infer the rest using an inference engine which is a piece of software able to infer logical consequences from a set of asserted facts [22]. The ontology is persisted in the file system although more sophisticated ontology containers can be used if the application needs scaling. The PMS component is the heart of the application;, it loads the unclassified profiles from the ontology and invokes an external inference engine called Pellet [23]. The Semantic Query-Enhanced Web Rule Language (SQWRL) is the language used to read profiles from the ontology; it is a query language designed specifically to read information

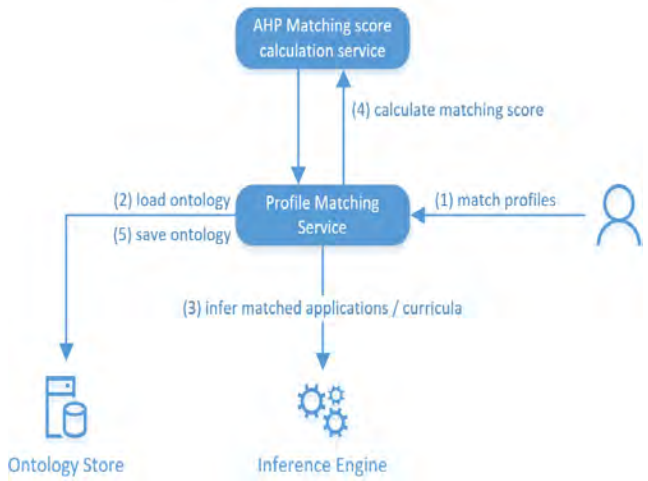

FIGURE 1. Application architecture.

from ontologies [24]. An example to load student transcripts from the ontology is shown below:

Student(?s) ^ hasTranscript(?t)
^consistsOfEnrolment(?t,?e)
^is-An-EnrolmentForCompetency(?e,?c)
^hasGrade(?e,?g)
→ sqwrl:select(?s,?t,?c,?g)

This query will return the list of students along with their full transcript which contains the gained competencies and the grades they obtained for each competency. SQWRL is accessed through the SQWRL API (Application Programming Interface) which is a Java-based API and is the interface between the PMS component and the ontology. Once the ontology is loaded, the inference engine executes a set of Semantic Web Rule Language (SWRL) rules [25] which define the rules that populate the ontology with more inferred knowledge and perform the profile matching which results eventually in the classification of all the education profiles to either fit or not fit to their associated industry profiles. The PMS component then updates the profiles in the ontology by marking all the processed ones to prevent them from being loaded in subsequent processing. The matched profiles are then converted into an object model and passed to the MSCS component. This component reads all the matched profiles and calculates the matching score for each profile using the AHP method. The profiles are then updated with the calculated score, and the ontology is updated.

The application has a conventional graphic user interface and a set of RESTful APIs [26] hosted in a web server to provide the direct communication with the ontology and issue queries about the different aspects of the domain like university curricula and occupations available. The graphical user interface displays the list of occupations defined in the ontology and the list of university curricula. It allows the user to drill down to list the competencies required by an

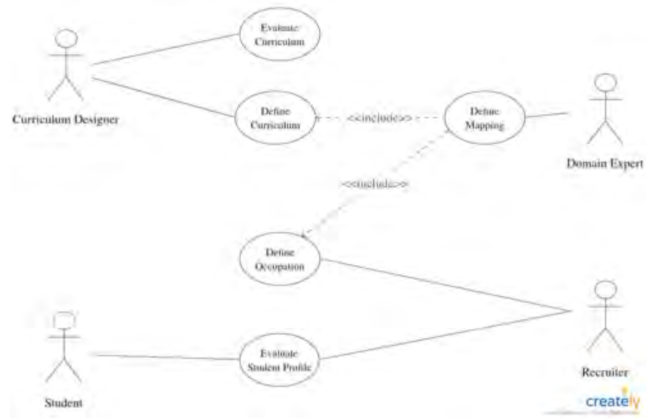

FIGURE 2. Application use case diagram.

occupation and also to list the courses of a university curriculum and their learning outcomes. Each learning outcome has a competency type (skill, ability, and knowledge) associated with it. Once an occupation and a curriculum are selected, the user can click on the 'calculate matching score' button to start the calculation process and display the results.

IV. SCENARIOS

The application is mainly used as an analysis tool by collaboration committees from the education and industry sectors to analyse the discrepancies between a curriculum and an occupation. The system can equally be used as a recruitment tool by industry companies to select candidates for recruitment. As depicted in Fig. 2, the main actors of the system are as follows:

- *Curriculum Designer*: this person defines the courses which constitute a university curriculum and defines the learning outcomes for each course. He is also responsible for evaluating the curriculum against related occupations in the industry based on the matching results of the application. The curriculum designer is part of an internal committee that is responsible for designing and updating the university curricula.
- *Student*: a student can evaluate his profile against an occupation. Based on the university program he selects and the grades he obtained, he gets a matching score describing how fit he is for the selected occupation.
- *Recruiter*: this person defines an occupation in the system. The definition must include a detailed list of competencies required and their requirements. He can calculate the matching score for student applicants against occupation openings.
- *Domain Expert*: this person is responsible for mapping between a university curriculum and an occupation. He must have industrial experience and a strong academic background. The domain expert is also part of a committee that consists of people from academia and industry responsible for evaluating university curricula.

The main use cases of the application are the following:

- *Define Curriculum*: defines the courses that constitute a university curriculum and their learning outcomes.
- *Evaluate Curriculum*: calculates a matching score for the curriculum against occupations in the industry.
- *Define Occupation*: defines an occupation and its required competencies.
- *Evaluate Student Profile*: calculates a matching score for a student profile against an occupation.
- *Define Mapping*: maps learning outcomes of a curriculum to the competencies required by an occupation. This mapping is the basis of the profile matching.

V. THE ONTOLOGY DESIGN

Several existing methodologies for constructing ontologies are available. We have followed the "Ontology Development 101" developed by Natalya Noy and Deborah McGuinness [27]. The language chosen to write the ontology is the OWL 2 Web Ontology Language [28], [29], and the tool used to build the model is Protégé (Version 4.3) [30]. To develop the ontology, we considered the following steps:

A. DEFINE THE DOMAIN AND THE SCOPE OF THE ONTOLOGY

To determine the domain and the scope of the ontology, we asked the experts in education and industry and studied competency formalisation methods like O*Net in the United States and 'ROME' in France.

B. CONSIDERING REUSING EXISTING ONTOLOGIES

In education and industry, several ontologies were found that model aspects of the domain. However, no ontology could be found that can be re-used to serve our intended purpose. Despite this, the current ontologies have been used as an inspiration to model the common concepts in the new ontology.

C. ENUMERATING THE DOMAIN TERMS

The ontology is be modelled as a taxonomy that helps describe the different aspects of the domain like student, competency, course, occupation, etc. Some concepts are further divided into subclasses that would improve the classification of the instances of these classes.

D. DEFINING THE CLASSES AND THE CLASS HIERARCHY

Classes are groups of individuals or instances that represent a class where all members share the same membership requirements. Classes are ordered in hierarchies called a taxonomy. Hierarchies are used by inference engines to infer inheritance relationships. Classes are defined by following the combination development process, which is a combination of both the bottom-to-top and the top-to-bottom approaches. Following this approach, the important terms are first defined, and generalisation and specialisation follow. This allows us to create the class hierarchy described in the next section.

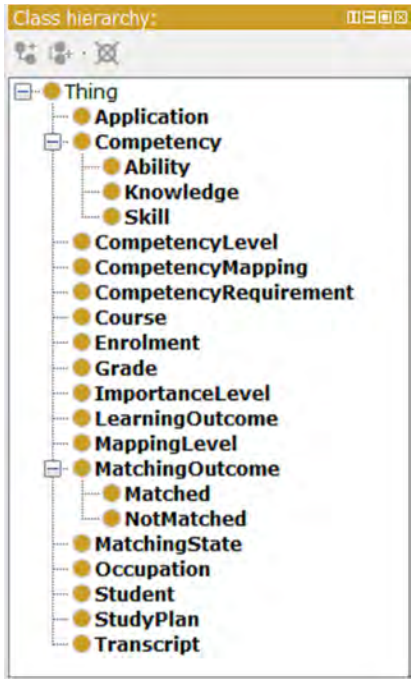

FIGURE 3. Model classes.

E. DEFINING THE RELATIONSHIPS BETWEEN CLASSES

The relationship between class members in an ontology are called properties. The two types of properties are object and data properties. Object properties represent binary relations between members of classes such as the relation between a *Student* and a *Transcript* where we define a property called *hasTranscript* to represent this relationship. Data properties link an individual to a data literal such as the *id* of a *Student.*

VI. THE ONTOLOGY MODEL

Fig. 3 shows the classes representing the domain and the hierarchy of the model. The purpose of each of these classes is explained in the following section.

A. CLASSES

The ontology model can be logically divided into the following sub-models:

1) *Common model:*
    - *Competency*: represents the central class in the model and is shared among all the sub-models. It has three sub-classes: Ability, Skill, and Knowledge. A competency can be an outcome of a course and gained by a student taking the course or is a requirement of an occupation.
2) *Education model:* the education profile consists of the following classes:
    - *Study Plan*: represents a curriculum and consists of a list of courses taught by a department in the university.
    - *Course*: a course of study that is normally recognised for credit towards the granting of an approved

degree. Each course should result in a list of learning outcomes gained by the student.

- *Learning Outcome*: The learning outcomes of a course. A learning outcome is mapped to one or more competencies.

3) *Student model:* the student profile consists of classes:
   - *Student*: represents a student in a university.
   - *Transcript*: represents the list of courses taken by the student and the grades he obtained.
   - *Enrolment*: is part of the transcript and is a combination of a course taken by a student and the grade he obtained in that course.
   - *Grade*: the grades for the courses are represented with the letters A, B, C, D, and F.

4) *Application model:* the application sub-ontology consists of the following classes:
   - *Application*: represents a student application for an occupation.
   - *Matching Outcome*: represents the outcome of the student application for the occupation.

5) *Occupation model:* the occupation sub-ontology consists of the following classes:
   - *Occupation*: represents an occupation in the industry.
   - *Competency Requirement*: each competency required by an occupation has a name, an importance level, and a required competency level.
   - *Importance Level*: represents how important the competency is for the recruiter and it can take the following values: Required, Preferred, Desired.
   - *Competency Level*: specifies the required level of a competency in an occupation and can take the following values: Knowing, Capable, Competent.
   - *Mapping Level* (Relevance): captures the degree of relevance between a curriculum competency and an occupation competency and take the following values: Weak, Related, Strong.

B. CLASSES STRUCTURE

Each of the classes described in the model has a definition that defines its object properties, data properties and its relation to other classes. For instance, the definition of the class *Student* is as follows:

```
Class: Student
  SubClassOf:
    hasTranscript some Transcript,
    selects some StudyPlan,
    hasApplication some Application,
    hasTranscript max 1 Transcript,
    selects max 1 StudyPlan
  HasKey: id
  DisjointClasses: Student, Competency,
  Transcript,...
```

This defines a class named *Student* as a subclass of an anonymous class which is defined by a set of properties relating it to other classes. The existential restrictions represented by the keyword *some* "describe classes of individuals that participate in at least one relationship along a specified property to individuals that are members of a specified class" [29]. For example, the axiom *Student hasApplication some Application* means that individuals of class *Student* have at least one *hasApplication* relationship with members of the class *Application*. Conversely, the universal restrictions represented by the keyword *only* "describe classes of individuals that for a given property only have relationships along this property to individuals that are members of a specified class" [29]. For instance, the class of individuals that only have *hasTranscript* relationships to members of the *Transcript* class would be described as *hasTranscript only Transcript*.

To ensure that members of a class cannot be members of another class, the *DisjointClasses* keyword is used. This is because OWL classes are assumed to overlap by default. Therefore, the ontology designer should make sure to explicitly make the classes disjoint where applicable [29].

Cardinality restrictions specify the number of relations between members of a class and members of other classes. Members of a class can have at least, at most, or exactly a specific number of relations with other members. The *minimum cardinality* restriction specifies the minimum number of relationships that a member of class can participate in. The *maximum cardinality* restriction specifies the maximum number of relationships that a member of a class can participate in. An *exact cardinality* restriction specifies the exact number of relationships that a member must participate in. In the *Student* class defined above, it is specified that a *Student* can have exactly one study plan using the assertion *selects max 1 StudyPlan*. With this assertion, an association of more than one *StudyPlan* with a *Student* will cause the ontology to become inconsistent.

The *HasKey* keyword says that each member of a class is uniquely identified by a data or object property or a set of properties. So, if two members of the class have the same values for each of the key properties, we can conclude that these two members are the same [31]. Fig. 4 shows the relationships between the remaining classes of the ontology.

C. ONTOLOGY INFERENCING USING SWRL

The Semantic Web Rule Language (SWRL) [25] has been added to OWL to extend its expressivity by adding rules to an ontology. These rules can be used to infer more knowledge from the ontology. For example, the relation between an *Occupation* and a *Competency* represented by the object property *requiresCompetency* is not directly asserted by the

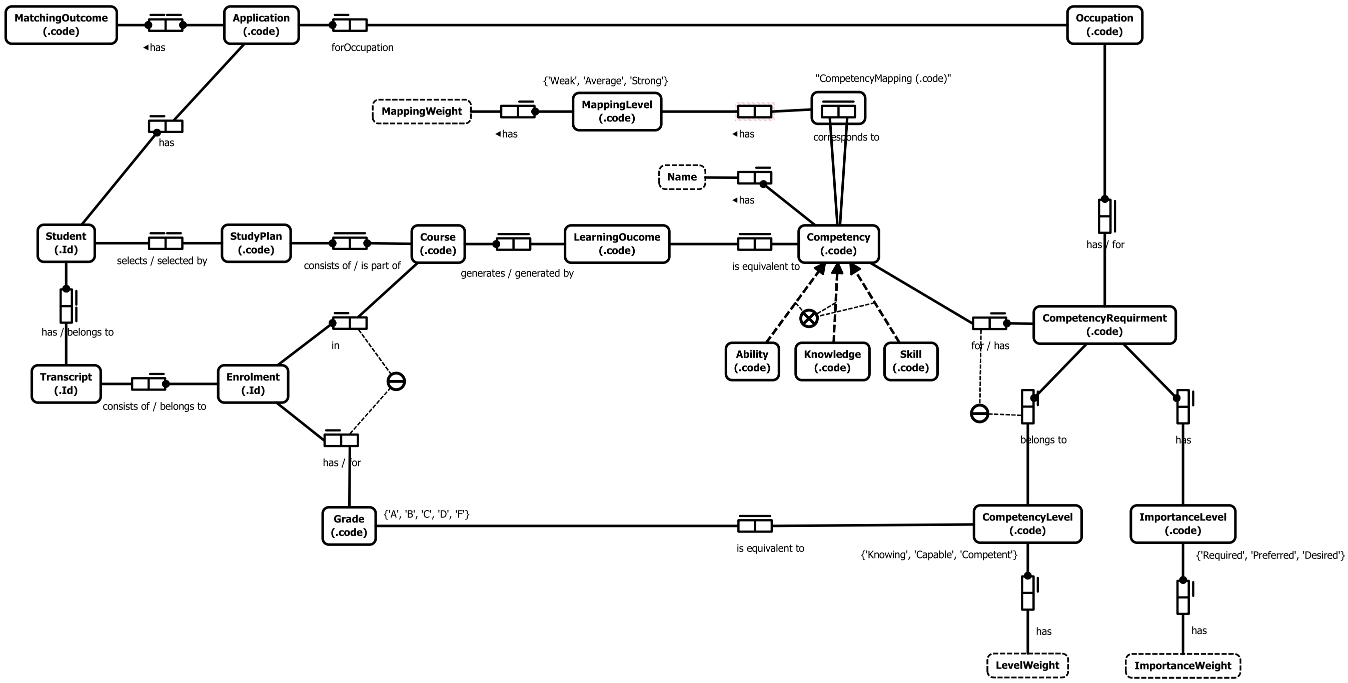

FIGURE 4. Ontology class relationships.

ontology designer in the ontology definition but inferred using the following rule:

Occupation(?o) ^hasCompetencyRequirement(?o,?r)
^forCompetency(?r,?c)
->requiresCompetency(?o,?c)

The rule is saying, if a member *o* from the *Occupation* class, has a *CompetencyRequirement r* which in turn describes a requirement for a *Competency c*, this implies that the *Occupation o requiresCompetency c*.

More complicated rules can be built. For example, to state that a *Student* has gained a *Competency* by enrolling in a *Course*, the following rule is applied:

Student(?s), hasTranscript(?s,?t)
^consistsOfEnrolment(?t,?e) ^inCourse(?e,?course)
^generatesLearningOutcome(?course,?lo)
^equivalentToCompetency(?lo,?competency)
-> hasGainedCompetency(?s,?competency)

VII. PROFILE MATCHING LOGIC

The profile matching logic in the ontology is expressed as a set of SWRL rules. A matching between an occupation and a student profile is performed by matching the competencies on each side. The matching between competencies is either based on having two competencies with the same name or based on a mapping defined between two competencies that have different identities. For a student profile to be classified as fit for an occupation, it needs to contain all the competencies required by an occupation. The matching logic steps are described in detail in the following sections.

A. MATCH BY NAME

The first step in the matching process is to find the set of student competencies that match the occupation competencies based on the competency name, i.e. the two competencies represent the same ontology member. This can be achieved using the following rule:

Student(?s), hasApplication(?s,?a), forOccupation(?a,?o)
^hasMatchingOutcome(?a,?matchingResult)
^hasGainedCompetency(?s,?c1),
requiresCompetency(?o,?c2)^SameAs (?c1,?c2)
-> hasMatchedCompetency(?matchingResult,?c1)

The *SameAs* atom [25] is used to check that *c1* and *c2* competencies represent the same ontology member. If so, the competency is added to the set of matched competencies.

B. MATCH BY DEFINED MAPPING

Some competencies could be defined differently but refer to the same competency. The following rule determines these competencies and adds them to the set of competencies of the student:

Student(?s), hasApplication(?s,?a), forOccupation(?a,?o)
^hasMatchingOutcome(?a,?matchingResult)
^hasGainedCompetency(?s,?c1)

^CompetencyMapping(?cm)
^mapsOccupationCompetency(?cm,?c2)
^mapsStudentCompetency(?cm,?c1)
-> hasMatchedCompetency(?matchingResult,?c2)

This rule adds the occupation competencies that are defined to be relevant to the student competencies (by using the *CompetencyMapping* class) to the list of competencies that a student has. For example, if a student has gained competencies *o1, o2, o3* and a *CompetencyMapping* individual exist that maps *o1* to *o4* where *o4* is a competency required by an occupation. Then *o4* is added to the set of competencies that the student has.

C. SUCCESSFUL APPLICATIONS MATCHING

Given the list of matched competencies from the two previous steps, the final list is compared with the list of competencies required by the occupation as follows:

Application(?a), forOccupation(?a,?o1)
^hasMatchingOutcome(?a,?mo)
^((is-AnOccupationForApplication some Application)
and (requiresCompetency only
(belongsToMatchingOutcome some
MatchingOutcome)))(?o2) ^SameAs (?o1,?o2)
->hasMatchingResult(?mo, Fit)

This rule matches the occupation *o1* associated with the student application *a* against the occupations that have competencies matching with a student profile and bind each of occupations to the variable *o2*, then using the *SameAs* axiom we check that the occupation *o1* is among that list of occupations. If there is a match, we can conclude that the application of the student has a match and mark the application as *Fit*.

VIII. ONTOLOGY EVALUATION

To assess the proposed ontology, the occupation of *Information Security Analyst* has been used to evaluate the Computer Science and Engineering (Qatar University) curricula. The data used to feed the job occupation and its competencies is derived from the data published on O*Net. The profile describing the job in O*Net is quite extensive, so we have opted to take only a subset of the highly important competencies based on the importance ranking of the O*Net. Therefore, any competency with 50% or more importance has been selected. The data used to feed the courses and study plans were derived from the Computer Science and Engineering curriculum. Two study plans were defined: one study plan containing courses on Computer Networks and Computer Security and we called it sp1 while the other study plan sp2 is missing these courses. The data has been defined in Protégé by creating instances of the model classes. As part of the data definition, we defined the mapping between the competencies required by the job and the outcome of the study plans. With two student profiles, each taking a different study plan and upon running the reasoner, the first student's profile taking the sp1 study plan was classified a fit as it contains all the needed competencies and the second student was classified as not fit. The model yielded the expected results. However, to make working with the ontology practical and to be able to involve all the stakeholders, we realised that we need to automate the model instantiation process as much as possible because we have observed that profiles and mapping definitions are labour intensive and time-consuming.

A. ONTOLOGY CHALLENGES

Despite the benefits that ontologies provide such as interoperability and provision of a shared understanding across domains, there are challenges in applying them to the domain of profile matching. One of the main difficulties we faced was to prove that an education profile is not fit for an occupation, while it was easy to prove it fits. The reason for this is due to the Open World Assumption (OWA) which OWL and SWRL assume. As per the OWA, if a proposition is not true with the present knowledge, the proposition cannot be declared as false because it might become true in the future when more knowledge is attained [32]. OWL works under OWA, because the semantic web assumes an unlimited amount of knowledge (Internet) [33]. Closed world based systems, on the other hand, assume that the absence of knowledge is false and the present knowledge in the system is complete. As the information sources for commercial information systems are finite, they are considered as closed worlds and treated under the Closed World Assumption (CWA) [34]. As for the problem at hand, if a required competency is missing from a student profile, then according to the OWA, it cannot be proved that the student does not have it. The student might acquire it in the future. There is a workaround for this limitation that involves closing the world manually by asserting that the competencies that the student has gained are the only ones he would ever gain as follows:

studentX hasGainedCompetency only ({c1, c2, c3})

With this assertion, we have limited the competencies of this particular student, and we can evaluate his profile against an occupation. This solution fixes the problem; however, it is impractical as it involves manually modifying the ontology for each profile. Works are ongoing on reasoners that reason under the CWA, like the work of Wang that proposes a new reasoner called BCAR [35].

The other issue we faced is that OWL is monotonic which says that adding new knowledge never falsifies a previous conclusion [32]. In our context, we found it difficult to apply state transitions for job applications. For example, implementing a simple state transition mechanism by initialising all applications to *unprocessed* then change their state to either *Matched* or *NotMatched* after processing is not possible as we are falsifying an older knowledge which states that the state of the application is *unprocessed*.

TABLE 1. Matching criteria alternatives.

Criterion Name	Values
Requirement Level	Knowing, Capable, Competent
Competency Relevance	Weak, Related, Strong
Competency Importance	Required, Preferred, Desired

IX. PROFILE MATCHING SCORE CALCULATION USING AHP

Profile matching score calculation involves calculating the matching score between each competency in the education profile and its matching industry profile competency. Then the total matching score between the two profiles is found by taking the average of the competency matching scores. The AHP method is used as an analytical tool to evaluate the level of matching against a set of three criteria. The criteria are: competency relevance, competency importance, and the competency required level. This evaluation criteria are explained below:

A. MATCHING CRITERIA

- *Competency Relevance:* There is no universal convention on competency names. So, there is a need to match competencies that are relevant but have different names, and this is achieved using the *CompetencyMapping* class in the ontology. The relevance degree between two mapped competencies is captured through this criterion.
- *Competency Importance:* Another factor that affects the matching score calculation is the importance of the competency for the job. S,ome competencies are mandatory whereas others are nice to have for instance. A student profile having more mandatory competency matchings than another student will potentially get a higher matching score.
- *Competency required level*: In addition to the competency importance, some occupations require that the competencies of a student be at a certain level of mastery. So, student profiles having the required level will score higher than students with a lower level. This matching criterion is only relevant in the context of comparing job profiles with student profiles and not with university programs. In the model, there is a matching between the required competency levels and the student grades. For example, the grades of A and B could be the equivalent of the required level of *Competent*, C equivalent to *Capable*, and so on.

TABLE 1 lists the values that each matching criterion can take.

B. A WORKED EXAMPLE

In the following example, we will calculate the matching score for a profile of a student against an occupation and calculate the matching score between a curriculum and an occupation. To keep the example simple, we assume that the curriculum contains only two courses. Following the AHP method, the steps required to calculate the matching score are as follows:

TABLE 2. Criteria comparison matrix.

Criteria	Level	Relevance	Importance
Level	1	1/2	1/6
Relevance	2	1	3
Importance	6	1/3	1

TABLE 3. Normalised comparison matrix.

Criteria	Level	Relevance	Importance
Level	0.11	0.27	0.04
Relevance	0.22	0.55	0.72
Importance	0.67	0.18	0.24

TABLE 4. Weight vector.

Criteria	Weight
Level	0.141
Relevance	0.496
Importance	0.363

1) DEFINE ANALYSIS GOAL

The starting point is to select the analysis goal. In this case, it is the calculation of a matching score between two competencies.

2) CONSTRUCT PAIRWISE COMPARISON MATRIX

The pairwise comparison matrix describes the relative importance of the criteria. We use a 1-9 ratio scale to obtain the data in constructing the comparison matrix. According to the 1-9 ratio scale, a surveyed expert suggests a comparison matrix which describes the relative importance of the three criteria as shown in TABLE 2.

3) NORMALISE THE COMPARISON MATRIX

The comparison matrix is normalised by applying the following equation:

$$\overline{a}_{ij} = \frac{\overline{a}_{ij}}{\sum\limits_{k=1}^{n}}, \quad i, j, = 1, 2, \ldots, n$$

By applying the equation above we get the following normalised matrix shown in TABLE 3.

4) CALCULATE THE WEIGHT VECTOR

The weight of each criterion is calculated using the following equations:

$$\overline{W}_i = \sum_{j=1}^{n} \overline{a}_{ij}, \quad j = 1, 2, \ldots, n$$

$$W = \frac{\overline{W}_i}{\sum\limits_{j=1}^{n} \overline{W}_j}, \quad i = 1, 2, \ldots, n$$

Applying the equations, we get the weight vector for the criteria as shown in TABLE 4.

TABLE 5. Industry / student profiles.

Id	Industry Competency	Student Competency
1	Programming in Java	Programming 101
2	Network Security	Computer Networks

TABLE 6. Industry profile requirements / student grades.

Id	Level	Relevance	Importance	Student Grade
1	Capable	Related	Required	B
2	Competent	Strongly related	Desired	A

TABLE 7. Industry / curriculum profiles.

Id	Required Competency	Course Competency
1	Programming in Java	Programming 101
2	Network Security	Computer Networks

TABLE 8. Industry profile requirements.

Id	Relevance	Importance
1	Related	Required
2	Strongly related	Desired

5) STUDENT PROFILE MATCHING

Using the weight vector obtained, let's compute the matching score for a student profile against an industry profile as shown in TABLE 5.

The requirements of the industry profile and the student grades are shown in TABLE 6.

To compute the matching score, the criteria alternatives are assigned the following values:

- Level: Competent = 90, Capable = 70, Knowing = 50
- Relevance: Strong = 90, Average = 70, Weak = 50
- Importance: Required = 90, Preferred = 70, Desired = 50
- Student grades: A = 90, B = 80, C = 70, D = 60.

These values are agreed upon by a committee consisting of people from education and industry. The mapping between the student grades and the competency required level is decided by the recruiter. In this example, we are assuming the following mappings: Competent maps to grade A, Capable to Grades B /- C, and Knowing to D.

Assuming the competency vector for a competency is V then the matching score is: W o V. Where W is the weight vector. From TABLE 6, the matching vector for Competency 2 (Network Security) is: V = (Level: A (90), Relevance (90), Importance: Desired (50)) = (90, 90, 50). As the weight vector = (0.141, 0.496, 0.363), then the matching score for competency 2 = $90 \times 0.141 + 90 \times 0.496 + 50 \times 0.363 = 75.48\%$

From the score obtained, notice that although the student score was 90, the resulting matching score dropped to 75% because the competency is not highly important for the job (Importance: *Desired*). Likewise, the matching score for competency 1 = $80 \times 0.141 + 70 \times 0.496 + 90 \times 0.363 = 78.67\%$

Comparing competencies 1 and 2, we notice that even though the student has obtained a lower grade for competency 1, the resulting matching score obtained was higher. This is because the competency is important for the recruiter and designated as *Required*. The final matching score is the average of the matching scores for the student competencies, and in this example it is equal to: $(78.67 + 75.48) / 2 = 77.0\%$. So, there is a 77% matching between the student profile and the occupation.

6) PROGRAM MATCHING

The profile matching between a university program and an occupation can be equally calculated by considering only two criteria which are the *Relevance* and the *Importance*. Following the previous example, the profiles to be matched and the occupation requirements are shown in TABLE 7 and TABLE 8.

The weight vector is recalculated by considering only the two criteria, so we get: W = (0.65, 0.36). The matching score for competency 1 in TABLE 7 = $70 \times 0.65 + 90 \times 0.35 = 77\%$. For the second competency, the matching score = $90 \times 0.65 + 50 \times 0.35 = 76\%$. Therefore, the final matching score = (77 + 76) / 2 = 76.5%. This means that there is a 76.5% matching between the curriculum and the occupation.

X. CONCLUSION

Studying the gap between higher education outcomes and the industry needs is an important endeavour that is worth investigating. However, developing efficient and accurate solutions for measuring that gap proved to be a challenging task to all stakeholders in education and industry. Despite some technological limitations, the proposed system presented in this paper showed how the ontologies, semantic web technologies, and the AHP method could be used to adequately capture the domain concepts and perform the required gap analysis. The ontology is considered a means to facilitate the communication between the different stakeholders and serves as a foundation for future collaboration that leads to reaching a solid competency model. As a future work, we would like to make the application more automated in terms of profile definition and competency mapping which are activities performed by a domain expert. These activities are labour intensive and time-consuming.

REFERENCES

[1] A. Bouras, L. Veillard, S. Tralongo, and M. Lenir, "Cooperative education development: Towards ICT reference models," in *Proc. Int. Conf. Interactive Collaborative Learn. (ICL)*, Dec. 2014, pp. 855–861.

[2] S. Monchatre, "En quoi la compétence devient-elle une technologie sociale? Réflexions á partir de l'expérience québécoise," in *Formation Emploi. Revue Française de Sciences Sociales*. Strasbourg, France, 2007, pp. 29–45.

[3] S. Stavrou, "Negotiating curriculum change in the French University: The case of 'regionalising' social scientific knowledge," *Int. Stud. Sociol. Edu.*, vol. 19, no. 1, pp. 19–36, 2009.

[4] T. Deibinger and S. Hellwig, "Structures and functions of competency-based education and training (CBET): A comparative perspective," InWEnt, Mannheim, Germany, Tech. Rep., 2005.

[5] N. G. Peterson, M. D. Mumford, W. C. Borman, P. R. Jeanneret, and E. A. Fleishman, *An Occupational Information System for the 21st Century: The Development of O∗NET*. Washington, DC, USA: American Psychological Association, 1999.

[6] J. Markowitsch and C. Plaimauer, "Descriptors for competence: Towards an international standard classification for skills and competences," *J. Eur. Ind. Training*, vol. 33, nos. 8–9, pp. 817–837, 2009.

[7] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic Web," *Sci. Amer.*, vol. 284, no. 5, pp. 28–37, 2001.

[8] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *Knowl. Eng. Rev.*, vol. 11, no. 2, pp. 93–136, 1996.

[9] A. Bouras, A. A. Zainal, and M. S. Abdulwahad, "Evolution of ICT industry landscape and its impact on higher education competencies," in *Proc. 15th Int. Conf. e-Learn. e-Bus., Enterprise Inf. Syst., e-Government (EEE)*, 2016, pp. 1–8.

[10] T. L. Saaty, "What is the analytic hierarchy process?" in *Mathematical Models for Decision Support*. Springer, 1988, pp. 109–121.

[11] I. Lee, "An architecture for a next-generation holistic e-recruiting system," *Commun. ACM*, vol. 50, no. 7, pp. 81–85, 2007.

[12] Q. Gao, J. Yan, and M. Liu, "A semantic approach to recommendation system based on user ontology and spreading activation model," in *Proc. IFIP Int. Conf. Netw. Parallel Comput.*, Oct. 2008, pp. 488–492.

[13] R. Rafter, K. Bradley, and B. Smyth, "Automated collaborative filtering applications for online recruitment services," in *Proc. Int. Conf. Adapt. Hypermedia Adapt. Web-Based Syst.*, Trento, Italy, 2000.

[14] A. Singh, C. Rose, K. Visweswariah, V. Chenthamarakshan, and N. Kambhatla, "PROSPECT: A system for screening candidates for recruitment," in *Proc. 19th ACM Int. Conf. Inf. Knowledge Manage.*, 2010, pp. 659–668.

[15] E. W. Wright, T. A. Domagalski, and R. Collins, "Improving employee selection with a revised resume format," *Bus. Commun. Quart.*, vol. 74, no. 3, pp. 272–286, 2011.

[16] X. Hu, L. Wu, C. Li, and M. Huang, "SMS-based mobile recommendation system for campus recruitment in China," in *Proc. 10th Int. Conf. Mobile Bus. (ICMB)*, 2011, pp. 152–157.

[17] S. Amdouni and W. B. A. Karaa, "Web-based recruiting," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. (AICCSA)*, May 2010, pp. 1–7.

[18] E. Faliagka, K. Ramantas, A. Tsakalidis, and G. Tzimas, "Application of machine learning algorithms to an online recruitment system," in *Proc. Int. Conf. Internet Web Appl. Services*, 2012, pp. 215–220.

[19] J. Pan, B. Zhang, S. Wang, G. Wu, and D. Wei, "Ontology based user profiling in personalized information service agent," in *Proc. 7th IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, 2007, pp. 1089–1093.

[20] A. M'Baya, J. Laval, N. Moalla, Y. Ouzrout, and A. Bouras, "Ontology based system to guide internship assignment process," in *Proc. 12th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Nov./Dec. 2016, pp. 589–596.

[21] H. Gasmi and A. Bouras, "PROMA: An ontology based application for education/industry collaboration," in *Proc. EDULEARN*, Barcelona, Spain, 2017, pp. 6919–6927.

[22] P. Wang, "Grounded on experience: Semantics for intelligence," Center Res. Concepts Cognition, Indiana Univ. Bloomington, Bloomington, IN, USA, Tech. Rep. 96, 1995.

[23] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Web Semantics, Sci., Services Agents World Wide Web*, vol. 5, no. 2, pp. 51–53, 2007.

[24] M. O'Connor and A. Das, "SQWRL: A query language for OWL," in *Proc. 6th Int. Conf. OWL, Exper. Directions*, vol. 529. 2009, pp. 208–215.

[25] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A semantic Web rule language combining OWL and RuleML," *W3C Member Submission*, vol. 21, p. 79, May 2004.

[26] L. Richardson and S. Ruby, *RESTful Web Services*. Newton, MA, USA: O'Reilly Media, Inc., 2008.

[27] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," Stanford Med. Informat., Stanford, CA, USA, Tech. Rep. SMI-2001-0880, 2001.

[28] D. L. McGuinness and F. Van Harmelen, "OWL Web ontology language overview," *W3C Recommendation*, vol. 10, p. 2004, Feb. 2004.

[29] M. Horridge, S. Jupp, G. Moulton, A. Rector, R. Stevens, and C. Wroe, "A practical guide to building owl ontologies using protégé 4 and co-ode tools edition 1.2," University of Manchester, U.K., 2009. [Online]. Available: http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_2.pdf

[30] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen, "The Protégé OWL plugin: An open development environment for semantic Web applications," in *Proc. Int. Semantic Web Conf.*, 2004, pp. 1–15.

[31] B. Motik *et al.*, "OWL 2 Web ontology language: Structural specification and functional-style syntax," *W3C Recommendation*, vol. 27, p. 159, Sep. 2009.

[32] N. Drummond, R. Shearer, "The open world assumption," Univ. Manchester, U.K., Tech. Rep., 2006. [Online]. Available: http://www.cs.man.ac.uk/drummond/presentations/OWA.pdf

[33] L. Baresi, E. D. Nitto, and C. Ghezzi, "Toward open-world software: Issues and challenges," *Computer*, vol. 39, no. 10, pp. 36–43, Oct. 2006.

[34] Q. Wang and X. Yu, "Reasoning over OWL/SWRL ontologies under CWA and UNA for industrial applications," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 7106. 2011, pp. 789–798.

[35] V. Fortineau, T. Paviot, L. Louis-Sidney, and S. Lamouri, "SWRL as a rule language for ontology-based models in power plant design," in *Proc. 9th Int. Conf. Product Lifecycle Manage. (PLM)*, 2012, pp. 588–597.

HOUSSEM GASMI received the B.Sc. degree in computer science from Qatar University in 2001 and the M.Sc. degree in software engineering from the University of Oxford, U.K., in 2013. He is currently pursuing the Ph.D. degree in data integration for deep learning with the DISP Laboratory, Université Lumière of Lyon, France. He is currently a Research Assistant with Qatar University. His research interest includes ontologies, machine learning and its applications in the domain of smart cities.

ABDELAZIZ BOURAS received the Ph.D. degree (Hons.) in science from Chiang Mai University, Thailand, from Her Royal Highness Princess MahaChakri Sirindhorn of Thailand, in 2011. He was a Full Professor (Exceptional Class) with Université Lumière of Lyon, France, a Deputy Director of the DISP Research Laboratory and the Director of the CERRAL innovation and knowledge transfer center. He is currently a Professor in computer science with the Computer Science and Engineering Department, Faculty of Engineering, Qatar University.