Journal of Universal Computer Science, vol. 17, no. 7 (2011), 1060-1077
submitted: 30/10/10, accepted: 15/3/11, appeared: 1/4/11

A Ranking Tool Exploiting Semantic Descriptions for the Comparison of EQF-based Qualifications

Valentina Gatteschi
(Politecnico di Torino, Dip. di Automatica e Informatica, Torino, Italy
valentina.gatteschi@polito.it)

Fabrizio Lamberti
(Politecnico di Torino, Dip. di Automatica e Informatica, Torino, Italy
fabrizio.lamberti@polito.it)

Andrea Sanna
(Politecnico di Torino, Dip. di Automatica e Informatica, Torino, Italy
andrea.sanna@polito.it)

Claudio Demartini
(Politecnico di Torino, Dip. di Automatica e Informatica, Torino, Italy
claudio.demartini@polito.it)

Abstract: Nowadays, one of the main issues discussed at the Community level is represented by the mobility of students and workers across Europe. During the last years, in order to deal with the above picture, several initiatives have been carried out: one of them is the definition of the European Qualification Framework (EQF), a common architecture for the description of qualifications. At the same time, several research activities were established with the aim of finding how semantic technologies could be exploited for qualifications comparison in the field of human resources acquisition. In this paper, the EQF specifications are taken into account and they are applied in a practical scenario to develop a ranking algorithm for the comparison of qualifications expressed in terms of knowledge, skill and competence concepts, potentially aimed at supporting European employers during the recruiting phase.
Key Words: Semantic Web, Ontologies, EQF, Matchmaking
Category: H.4.2, I.2.4

1 Introduction

During the last years, the mobility of students and workers across Europe has become more and more a relevant topic in the Community scenario. A context in which the professional experience could be analyzed from a transnational point of view requires a methodology capable of guaranteeing transparency, comparability, transferability and recognition of qualifications and associated learning outcomes (i.e. knowledge, skills and competences) across different countries. For this reason, several initiatives have been carried out in order to overcome the gaps among training systems with the final aim to develop “a knowledge-based

Europe" capable of ensuring a "European labor market open to all", as it is expected from the Bruges-Copenhagen process and the Copenhagen Declaration [The Copenhagen Declaration, 2002].

Viable strategies for achieving the above goals should rely on the exploitation of common rules for the description of qualifications acquired after a training path, or during everyday work.

A first step toward the creation of a shared base in the lifelong learning domain has been done by the European Parliament Council which, in 2008, defined the European Qualification Framework (EQF) [EQF, 2008], a common reference system devised to support the linking of different countries' national qualifications systems and frameworks together. In the vision of the EQF, the above objective is expected to be achieved by exploiting a rigorous classification of lifelong learning qualifications based on eight reference levels and by identifying precisely the semantics of associated learning outcomes (referred to as level descriptors) in terms of knowledge (*the body of facts, principles, theories and practices that is related to a field of work or study*), skill (*the ability to apply knowledge and use know-how to complete tasks and solve problems*), and competence (*the demonstrated ability to use knowledge, skills and personal, social and/or methodological abilities, in work or study situations and in professional and personal development in order to achieve objective results according to a specific level of autonomy and context complexity*) concepts, thus opening the way for the creation of a shared understanding in the lifelong learning domain.

However, the definition of a European-wide framework is only the beginning of a more complex process: in fact, although the EQF defines several guidelines for the description of qualifications, in order to guarantee mobility, other tools - such as instruments for supporting students and workers who want to continue their training or working career abroad, or companies who are looking for workers with specific abilities - have to be developed. Hence, it is clearly visible that in order to be able to work at the European level, these instruments should depend on descriptions of qualifications (and associated learning outcomes) achieved by a given student or worker based on a standard and syntax-independent formalism, i.e. by making reference to strategies and tools developed in the framework of the Semantic Web related initiatives.

Several works presenting interesting applications of semantic technologies to the learning domain already exist in the literature; one of them is represented by the CUBER-project [Pöyry and Puustjärvi, 2002], where a tool to support learners in looking for European higher education courses that match their needs was developed. The work presented in [Nemirovskij et al. 1999] goes beyond the above solution: in particular, the authors presented a semantic search strategy based on the analysis of the relations between concepts belonging to user queries and concepts used in learning documents, and developed a collection of Web services for the search and the comparison of study programmes.

While the authors of the above projects exploited Semantic Web instruments in the learning domain, other works focused on the occupational field. In particular [Lau and Sure, 2002] proposed an ontology-based skill management system for the classification of employee's skills, providing a search feature within the intranet while [Garro and Palopoli, 2003] developed an XML multi-agent system providing support to the management during the search of the most suitable employee for a specific job. Another interesting work is reported by [Colucci et al. 2007]: here the authors presented a strategy exploiting Description Logics (DLs) [Baader et al. 2002] for annotating curricula based on a given ontology, so as to avoid ambiguities in the description; moreover, they also proposed an ontology-based search engine built upon the above description.

Another promising field of research is characterized by the development of algorithms for matchmaking (a process that queries a knowledge base and returns all the elements that potentially match the requirements expressed by the user). In one of the works related to this topic ([Lei and Horrocks, 2004]), the authors investigated how semantic technologies could be exploited to support service advertisement and discovery in e-commerce, and developed a Description Logic reasoner to compare ontology-based service descriptions. A similar issue has been investigated in [Di Noia et al. 2004]: in this work the authors presented two algorithms for the comparison of demands and supplies in the electronic marketplace, based on a modification of the CLASSIC structural subsumption algorithm [Borgida et al. 1989] and analyzed whether an efficient matchmaking algorithm could provide results similar to ranking made by human users.

While a matching between a demand and an offer could be done by exploiting subsumption relationships, a step forward could be to investigate which elements could be added to a supply that only partially matches a demand, in order to make it fully satisfy the requirements expressed by the end user. This subject has been considered in [Di Noia et al. 2003]: in this work, the authors presented an algorithm capable of supporting users who are looking for apartments to rent by solving a Concept Abduction Problem.

While the authors of the above works highlighted the importance of ontological descriptions for overcoming multi-cultural barriers and developed some interesting reasoning rules, the authors of [Pernici et al. 2006] significantly contributed to the domain pertaining the analysis of qualification semantics, by describing learning outcomes as a combination of knowledge, action verbs and context concepts. In particular, according to [Pernici et al. 2006], a knowledge can be defined as a set of knowledge objects (KO); a skill can be represented as a KO "put into action", i.e. as one or more pairs KO – Action Verb (AV); finally, a competence can be identified by means of a triple KO – AV – CX, that describes the ability of putting into action a KO in a specific context (CX). Thanks to this

EQF-aware representation, higher-level elements of the transnational framework could be expressed, from a practical point of view, as a sum of lower-level elements. Moreover, lower-level elements could be decomposed and analyzed at a even lower level of details. Finally, KO, AV and CX elements could be organized into an ontology, and suitable inference rules could be created upon them.

In this paper we present an ontology-based matchmaking algorithm for the comparison of qualifications expressed according to the European guidelines [EQF, 2008]. Starting from the *rankPotential* algorithm in [Di Noia et al. 2004], we propose a ranking method that integrates a subsumption technique taking into account the EQF formalism and the definition of knowledge, skill and competence elements given in [Pernici et al. 2006]. Moreover, we propose an adaptation of the *findIrred* algorithm [Di Noia et al. 2003] that could be used in the EQF perspective.

The remaining of the paper is organized as follows: Section 2 provides some basics of Description Logics, whereas Section 3 illustrates the reference algorithms and discusses their applicability in the considered scenario. Section 4 presents the modified algorithms, while Section 5 reports on experimental results. Finally, conclusions are drawn in Section 6.

2 Description Logics

Description Logics (DLs) are formalisms for the representation of the knowledge of a given application domain in a structured and formally well-understood way. The basic elements of a DL are *concept* names (classes such as `movie`, `person`, etc.) that are used for the description of real-world objects, and *role* names (like `hasDirector`, `hasKnowledge`, etc.) that identify binary relations between objects. Moreover, the *domain of interpretation* of concepts is represented by $\triangle$, and roles denote relations in the subset of $\triangle \times \triangle$.

According to the DL formalism, concepts could be combined by using constructors like *disjunction* ($\sqcup$), *conjunction* ($\sqcap$) and *complement* ($\neg$) whereas, by exploiting *existential quantification* ($\exists$) and *universal quantification* ($\forall$) constructors, *role restrictions* could be identified. Other constructors are $\top$ and $\bot$, that identify all the objects in the domain and the empty set, respectively.

By exploiting DLs, it is possible to describe complex concepts such as $\texttt{Student} \sqcap \neg\ \texttt{Female}$, that denotes all the students that are not female. Other examples concerning also roles are $\texttt{Student} \sqcap \exists\ \texttt{hasParent.Professor}$, which identifies the students with at least one parent who is a professor, or $\texttt{Person} \sqcap \forall\ \texttt{hasChild.Male}$ that denotes persons who have only male children.

A core element of DLs is the *Terminological Box* (TBox), that represents a set of axioms built by combining concepts through the *inclusion* ($\equiv$) or *definition* ($\sqsubseteq$) operators (in this case new concepts could then be defined by exploiting other concepts which were defined previously). Consequently, in order to express the fact that, as a matter of example, a given actor could be either the main character or a supporting actor of a particular movie, the following two definitions could be used: $\texttt{actor} \sqsubseteq \texttt{maincharacter} \sqcup \texttt{supportingactor}$ and $\texttt{maincharacter} \sqsubseteq \neg \texttt{supportingactor}$. It is worth remarking that, since a TBox expresses a set of relations between concepts, it could be used to represent an ontology. As a result, inference rules could be created in order to identify implicit relations among concepts.

Some reasoning services usually provided by DLs-based systems are:

1. **Satisfiability**: a concept C is satisfiable with respect to a TBox $\mathcal{T}$ if there exists an interpretation $\mathcal{I}$ in which C is mapped into a nonempty space.

2. **Subsumption**: given a TBox $\mathcal{T}$ and two concepts C and D, D subsumes C if it is more general than D in any model of $\mathcal{T}$; the subsumption between C and D is expressed as $C \sqsubseteq_{\mathcal{T}} D$ or $\mathcal{T} \models C \sqsubseteq D$.

With the aim of favouring comparability between the proposed ranking strategy and the approach in [Di Noia et al. 2004] and [Di Noia et al. 2003], in this work we made reference to the CLASSIC system [Borgida et al. 1989], a data model for the description of the general nature and structure of generic objects. According to this notation, each concept C can be represented through its normal form, that makes reference to a combination of three components, i.e. $D_{names} \sqcap D_{\sharp} \sqcap D_{all}$, where D_{names} denotes the conjunction of all the concept names belonging to the TBox, $D\sharp$ is the conjunction of number restrictions related to roles, and D_{all} links all concepts of the form, one for each role $\forall\ R.D$, where D is in the normal form, hence guaranteeing syntax independence.

3 Reference Algorithms: Context and Constraints

The application domain of the comparison strategy presented in this paper could be better understood by considering the example of a company that is looking for new human resources to recruit. When the employer decides to hire new workers, he has clearly in mind the "qualities" (expressed as a set of knowledge, skills and competences) the future employees should have, but he encounters several difficulties during the analysis of the received curricula. In fact, usually, in a profile description the above learning outcomes are presented in a heterogeneous way, without any shared lexicon; thus, for the employer, the inspection of a large amount of profiles in a short time with the aim of identifying the best candidates for a particular working position is a task hard to accomplish.

In this context, a tool computing the match based only on textual comparison could fasten the curricula selection phase, but it probably would not be able to identify the potential candidates that did not express, in an explicit way, the required qualities.

On the contrary, the exploitation of ontological descriptions for the representation of concepts organized in a hierarchical structure could provide a better result, and could ease the work of the employer, by allowing him to identify which are the connections between the elements of a curriculum and the requested qualities (e.g. *more specific*, *more general* concept, etc.) and by helping him to discover if, given a request, there are compatible offers and, in conclusion, which are the best available candidates (and why). As a matter of example, it could be interesting to focus on a company that is looking for a *programmer of dynamic Web pages*: let us assume that the human resources office receives a curriculum, built according to the EQF specifications, expressing the knowledge of *ASP* and *PHP*. In this case, a semantic search engine could easily identify the applicant as a possible candidate, because *ASP* and *PHP* concepts are subsumed by the *dynamic Web pages* concept (i.e. *ASP* and *PHP* are more detailed concepts, and the knowledge of *ASP* and *PHP* implies the knowledge of *dynamic web pages*).

Even though the above example could hide an easy matchmaking problem, when skills and competences have to be compared in a comprehensive EQF perspective, the problem becomes definitely more complex: in fact, as previously said, these two elements can be considered as a combination of lower-level concepts, namely KO − AV for skills, and KO − AV − CX for competences. Thus, if the source profile and the target profile differ even for only one lower-level element, they cannot be classified as equal. In order to explain this statement, it could be useful to examine a target skill *to program operating systems*, and a source skill *to use operating systems*: it can be easily seen that the two skills are different, and therefore denote different levels of ability. Moreover, like for the knowledge, also AV (and CX) elements could be organised according to an ontology; hence, in a possible scenario, a source skill *to develop Linux* could be more specific than the given target skill.

Thus, it is clear that during the development of a matchmaking algorithm designed for the EQF dimension, the above constraints should be taken into account. The solution proposed in this paper moves from the approach presented in [Di Noia et al. 2004] and [Di Noia et al. 2003] where, starting from two $\mathcal{ALN}$ concepts C and D both satisfiable in $\mathcal{T}$, the *findIrred* and the *rankPotential* algorithms are designed to solve the Concept Abduction Problem and to compute the semantic distance between the above concepts.

However, even though the *rankPotential* and the *findIrred* algorithms provide interesting results in many application scenarios [Ragone et al. 2005], they are not designed to cope with roles hierarchies and they do not allow to manage relations defined in [Pernici et al. 2006]. For this, we decided to describe AV and CX as concepts (rather than roles), linked by relations such as *hasActionVerb* and *actsOnKnowledge*. Thus, as a matter of example, the skill *to develop Linux* previously considered is no more represented as a pair role-concept `develop.Linux`, but rather as a relation `develop` $\sqcap$ `actsOnKnowlege.Linux`.

Even though, based on the above discussion, it seems that skills and competences can be represented in a suitable way, in the above configuration the *rankPotential* and the *findIrred* algorithm show some problems in correctly identifying relations among KO, AV and CX elements.

This could be explained again with an example: let us consider, among the skill elements, a demand D defined as *to compile Java and to debug C++ and C♯* and two supplies, namely $S1$ and $S2$, defined as *to compile Java* and *to debug Java and to compile C++ and C♯*, respectively (Figures 1(b) and 1(c)). When $rankPotential(S1, D)$ is computed, the algorithm detects the absence of *debug*, *C++* and *C♯* concepts. In a similar way, when $rankPotential(S2, D)$ is computed, the algorithm identifies the presence of *compile* and *debug* concepts, but it does not see the *Java*, *C++* and *C♯* concepts. Thus, both supplies are assigned the same ranking ($n = 3$). Despite this result, it is evident that $S1$ should obtain a better ranking, as it satisfies – at least in a partial way – the requirements of D. Similarly, $S2$ should be classified as worst, since it does not match with D.

In order to better understand the behaviour of the *rankPotential* algorithm, it could be useful to consider the abduction algorithm. Consequently, when the *findIrred* algorithm is applied to the abduction problem $\mathcal{P}1$=$\langle\mathcal{L}$, $S1$, D, $\mathcal{T}\rangle$, it provides, as a result $H1$ = `(all hasActionVerb (and Debug (all actsOnKnowledge (and (C++ C♯))))))`. On the contrary, the result of *findIrred(P2)*, being $\mathcal{P}2$=$\langle\mathcal{L}$, $S2$, D, $\mathcal{T}\rangle$ is $H2$ = `(all actsOnKnowledge (and (Java C++ C♯)))`.

Another example showing that, in some cases, the *rankPotential* algorithm may not produce optimal results is given by the skills *to compile Java and to debug C♯* and *to debug C++ and C♯*, that will be referred to as $S3$ and $S4$ (Figure 2(b) and 2(c), respectively). On the one hand, the distance computed by $rankPotential(S3, D)$ is $n = 1$, as the algorithm identifies the presence of *compile*, *Java*, *debug* and *C♯* concepts, but it does not find *C++*. On the other hand, when $rankPotential(S4, D)$ is computed, a distance $n = 2$ is obtained, since only the *debug*, *C++* and *C♯* concepts are found (whereas *compile* and *Java* are missed). Despite this ranking, a more detailed analysis could show that $S3$ actually lacks the pair *debug* – *C++*, whereas in $S4$ it is the pair *compile* – *Java* that is missing. This means that the two profiles should be considered as equivalent.

An in-depth analysis of the results of the *rankPotential* algorithm applied to $S3$ and $S4$ is provided by the *findIrred* algorithm. Here, the two concept abduction problems could be defined as $\mathcal{P}3$=$\langle\mathcal{L}$, $S3$, D, $\mathcal{T}\rangle$ and $\mathcal{P}4$=$\langle\mathcal{L}$, $S4$,

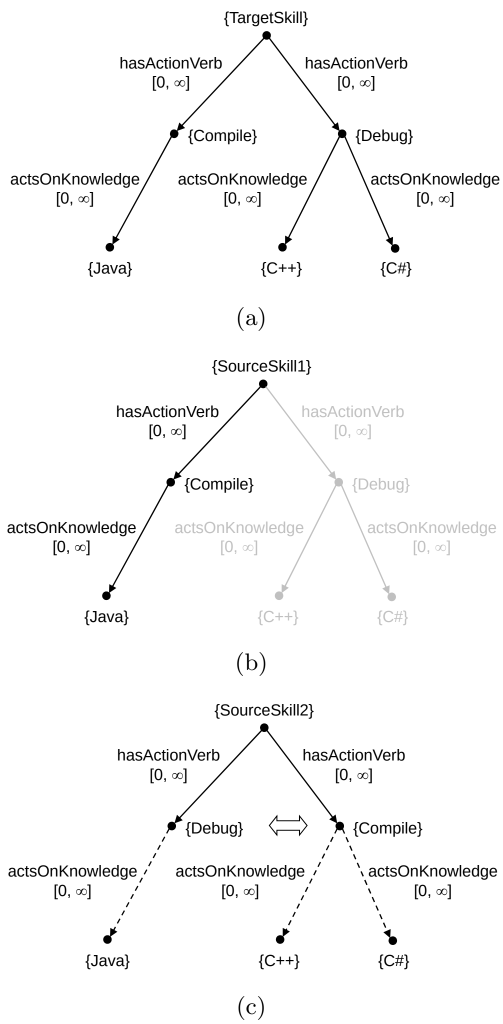

Figure 1: Demand *to compile Java and to debug C++ and C*$\natural$ (a) and skills *to compile Java* (b) and *to debug Java and to compile C++ and C*$\natural$ (c)

D, $\mathcal{T}\rangle$. In this case, by computing *findIrred(P3)* and *findIrred(P4)*, $H3 =$ `(all actsOnKnowledge (and (C++)))` and $H4 =$ `(all hasActionVerb (and Compile (all actsOnKnowledge (and (Java)))))` are obtained.

It is worth remarking that, even though for sake of simplicity selected examples focused only on skills, all the above considerations also apply to competences.

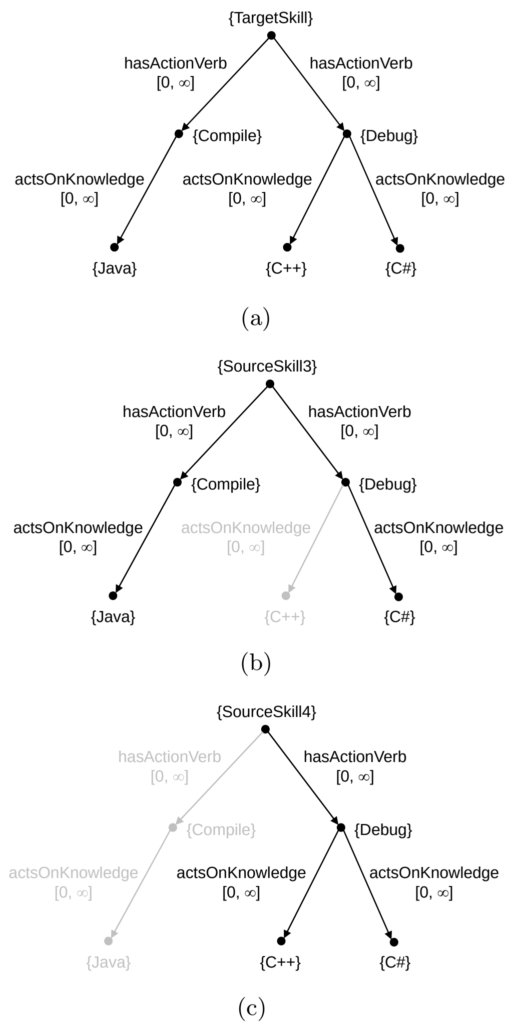

Figure 2: Demand *to compile Java and to debug C++ and C♯* (a) and skills *to compile Java and to debug C♯* (b) and *to debug C++ and C♯* (c)

4 Proposed Approach

The proposed approach aims at computing the semantic distance between a demand and a set of supplies, described as a set of words. We suppose that a) both demand and supplies are described in terms of a set of words (we consider this description as a particular case of a DL $\mathcal{L}$), and b) for all the concepts, a common ontology is defined, as Tbox $\mathcal{T}$ in $\mathcal{L}$.

Let us consider a demand $D = \{$*to program C++*$\}$ and a supply $S = \{$*to* *program C++ and C♯*}. Although the demand and the supply are not identical, all the requirements of D are satisfied by S, since it contains all the fundamental knowledge elements expressed by the employer. The fact that S contains additional details should not be considered as misleading; on the contrary, it should be regarded as a possibility, for the employer, to better refine his request, by specifying additional constraints which had not been considered in a previous step. Moreover, it is worth remarking that the result of the comparison between a demand D and a supply S, in the case of non-identical elements, should be different from the result of the comparison between the supply S and the demand D.

The above considerations point out that a matchmaking system should have the following properties:

> **Property 1: (Open-world semantics)**: *The lack of a characteristic in the description of a supply or a demand should not be considered as a constraint of absence, but it should be interpreted as a characteristic that could be either refined later or left open if it is irrelevant for the user.*

In other words, the absence of information in a source qualification does not have to be regarded with a negative approach. It should be instead interpreted as a under-specified information, possibly neglected in the definition of the qualification because considered irrelevant, or implicit, or simply forgotten. According to the above definition, comparison results into the identification of hypotheses on what it is not explicitly described thus improving the degree of comparability among qualifications.

> **Property 2: (Non-symmetric evaluation)**: *The result of the computation of the match between a demand D and a supply S may differ from the result of the comparison between S and D, i.e. it depends on the objective of the comparison.*

In addition to the above properties, a matchmaking system should also be syntax independent in ranking and should guarantee monotonicity of ranking over subsumption. This leads to the following two definitions:

> **Definition 1: (Syntax independence in ranking)**: *A ranking of concepts is* syntax independent *if, given a demand D, two supplies S1 and S2 and an ontology $\mathcal{T}$, if S1 is equivalent to S2, both supplies have the same ranking for the match with D (and the same holds for each pair of demands D1 and D2 when compared to a supply S).*

In other words, logically equivalent learning outcomes should get the same ranking in each matching process despite the fact that a different syntax is used to describe them.

Definition 2: (Monotonicity of ranking over subsumption): *A ranking of potential matches is* monotonic over subsumption *if, given a demand D and two supplies S1 and S2 with* $\mathcal{T} \models (S2 \sqsubseteq S1)$*, S1 has the same or a better ranking than S2 (and the same holds for each pair of demands D1 and D2 when compared to a supply S).*

The meaning of the above definition is that more specific concepts should have a better ranking than less specif ones, since they imply the knowledge of the more generic concept.

The devised matchmaking system considers all the above properties. In particular, the proposed *rankPotentialKSC* algorithm presented below takes as input two concepts C and D (expressed in normal form), computes the semantic distance between them and provides a rank value equal to zero if a concept C is subsumed by a concept D.

Algorithm *rankPotentialKSC*(C, D)
input: CLASSIC concepts C, D, in normal form, such that $C \sqcap D$ is satisfiable
output: rank $n \geq 0$ of C w.r.t. D, where 0 means $C \sqsubseteq D$ (best ranking)
begin algorithm
let $n := 0$, $t := 0$, $d := 0$, $a := 0$ **in**
/* add to d the number of concept names of D which are not among the concept names of C */
$d := d + |D_{names+} - C_{names+}|$;
/* if a = 1 add the result of the previous call */
if $a = 1$ **then**
  if $d \neq 0$ **then** $d := |D_{names+}| + p$; $q := 0$;
for each concept $R.E$ in D
/* for each CX or AV, store the result of the current call */
  if $R = actsOnKnowledge$ **and** $a = 0$
    then $a := 1$; $p := |D_{names+}| + q$;
    else if $R = hasActionVerb$
      then $a := 0$; $q := |D_{names+}|$;
      else $a := 0$; $n := n + d$;
        if $d = 0$ **then** $t := t + 1$;
        $d := 0$;
/*for each universal role quantification in C add the result of a recursive call*/
for each concept $\forall R.E \in D_{all}$
  if there exist $\forall R.F \in C_{all}$
    then $n := n + rankPotential(F, E)$;
    else $n := n + rankPotential(\top, E)$;
return n; **return** t;
end algorithm

Based on the discussion in Section 3 and considering the *rankPotential* algorithm selected as a reference, when the EQF scenario has to be tackled the ranking approach must take into account two main constraints:

- when a skill or competence belonging to a supply S and a demand D are compared, if they differ even only by one element, the rank value should not be only influenced by the differing element, but it should be affected by all the concepts composing it;
- if an AV (or a pair CX – AV) is linked (through a role R) to more than one KO elements, the ranking should be influenced by the number of KO elements the AV is linked to.

Hence, in order to satisfy the first constraint, two integer variables d and t have been introduced: d expresses the semantic distance between two learning outcomes, whereas t counts the total number of learning outcomes that show a semantic distance equal to zero (i.e., their composing elements are either identical or subsumed by the ones belonging to D). Thus, when $d = 0$, t is increased. This variable allows the users to go into more depth when the result provided by the algorithm shows equality between two or more concepts (in this case, the highest ranking will be obtained by the concept showing the highest t). Additionally, in order to better characterize the rank value, a percentage value could be calculated, by dividing t by the total number of combinations.

In order to deal with the second constraint, the variable a has been introduced: the value of this variable is normally equal to zero, except in the case of an AV linked to a KO element through a role of type *actsOnKnowledge*, where its value is set equal to one. This means that, when the KO element will be analyzed in order to identify the number of concepts of the demand that are not satisfied by the supply, also AV and CX elements should have to be considered.

Finally, p and q variables store the semantic distance of the couple CX – AV and the AV element respectively. These results are exploited during the analysis of KO elements, since KO, AV and CX must be considered together.

It is worth observing that the proposed methodology allows the system to attach a specific weight to any learning outcome, by adding to n the semantic distance d multiplied by the corresponding weight. In this way, when different requirements are characterized by different levels of importance, a better rank could be provided. The devised algorithm allows the user to rank a set of supplies with respect to a target demand. Should the end user want to go into more depth and examine which elements must be added to the description of an offer in order to let it satisfy the demand, abduction should be considered.

Definition 3: (Concept Abduction): *Let C and D be two concepts in a DL $\mathcal{L}$, and $\mathcal{T}$ a set of axioms in $\mathcal{L}$. A* Concept Abduction Problem *(CAP)* *denoted as* $\langle \mathcal{L}, C, D, \mathcal{T} \rangle$ *is finding a concept* $H \in \mathcal{L}$ *such that* $\mathcal{T} \not\models C \sqcap H \equiv \bot$ *and* $\mathcal{T} \models C \sqcap H \sqsubseteq D$.

Consequently, the solution to a CAP can be interpreted as what has to be added to C in order to make it more specific than D, which would make subsumption result true. Hence, if a demand D is not completely satisfied by a supply S, it may be very interesting to know which parts of the demand are not covered by the supply. In the following, we denote a CAP as $\mathcal{P}$ and we indicate with $SOL(\mathcal{P})$ the set of all the solutions to a CAP $\mathcal{P}$.

Moreover, since according to DLs a concept could be written in a normal form as conjunction of concepts, the result of the CAP should be analyzed in order to check whether the solution is irreducible.

Definition 4: (Irreducible solution): *Let* $\mathcal{P} = \langle \mathcal{L}, C, D, \mathcal{T} \rangle$ *be a CAP in which a normal form is admitted. Then, the set* $SOL_{\sqcap}(\mathcal{P})$ *is the subset of* $SOL(\mathcal{P})$ *in which no sub-conjunction of* C *is included. These concepts are called* irreducible solution *of* $\mathcal{P}$.

The *findIrredKSC* algorithm is presented below.

Algorithm *findIrredKSC*($\mathcal{P}$)
input: a CAP $\mathcal{P}=\langle \mathcal{L}, C, D, \mathcal{T} \rangle$, with $\mathcal{L} = \mathcal{ALN}$, acyclic $\mathcal{T}$ and $\mathcal{T} \not\models C \sqcap D \equiv \bot$
output: concept $H \in SOL_{\sqcap}(\mathcal{P})$ ($H = \top$ means that C is already subsumed by D)
begin algorithm
let $H := \top$, $A := \top$, $X := \top$ $a := 0$ **in**
for each concept name y in D
    if y is not in C **then** $X := X \sqcap y$;
    $A := A \sqcap y$;
/* if a = 1 add the result of the previous call */
if $a = 1$ **then**
    if $X \neq \top$ **then** $X := P \sqcap A$;
    $Q := \top$;
for each concept $R.E$ in D
/* for each CX or AV, store the result of the current call */
    if $R = actsOnKnowledge$ **and** $a = 0$
        then $a := 1$; $P := A \sqcap Q$;
        else if $R = hasActionVerb$
            then $a := 0$; $Q := \top \sqcap A$;
            else $a := 0$; $H := H \sqcap X$; $X := \top$;
    $A := \top$;
/*for each universal role quantification in C add the result of a recursive call*/
for each concept $\forall R.E \in D_{all}$
    if there exist $\forall R.F \in C_{all}$

then $H := H \sqcap \mathit{findIrredKSC}(\langle \mathcal{L}, F, E, \mathcal{T} \rangle)$;
else $H := H \sqcap \mathit{findIrredKSC}(\langle \mathcal{L}, \top, E, \mathcal{T} \rangle)$;
/* now $H \in SOL(\mathcal{P})$, but it might be reducible */
for each concept H_i in H
if H without $H_i \in SOL(\mathcal{P})$
then delete H_i from H; **return** H;
end algorithm

First, for each concept belonging to the description of the demand, a search within the representation of the supply is carried out: if a requirement of the demand is missing in the supply, the respective concept is added to X. Subsequently, if the role linking the previous concept to the following one is of type *actsOnKnowledge* or *hasActionVerb*, the elements of the demand are stored into the variables P and Q, respectively. These values will then be used to extend the set X, in the case C is not subsumed by D.

Once the set $SOL(\mathcal{P})$ has been identified, it has to be inspected to verify the minimality of the solution. For this reason, the algorithm checks if the concept H belongs to the irreducible solution: if not, it is deleted from the set of $SOL(\mathcal{P})$.

It is worth remarking that in this phase the relationships among concepts should be considered by the algorithm: in fact, it is possible that in the $SOL(\mathcal{P})$ an AV (or CX) is repeated several times, each time linked to a different KO. In this case, in order to guarantee the correctness of the solution, the AV (CX) should not be deleted from the set.

5 Experimental Results

In order to analyse the effectiveness of the proposed solution, a sample demand D and several supplies $S1$, $S2$, $S3$, $S4$, $S5$, $S6$ and $S7$ are considered (Figure 3). Results obtained using the *rankPotential* and *rankPotentialKSC* algorithms are reported in Table 1.

It can be easily seen that both the algorithms assign to $S4$ and $S7$ the worst ranking, as most of their composing concepts are not subsumed by the requirements expressed in D.

On the contrary, $S2$ and $S5$ appear to be promising supplies, since they show a lower semantic distance. However, while according to the *rankPotential* algorithm $S2$ is assigned the highest ranking (since it satisfies half of the requirements of D, i.e. a knowledge, a skill and a competence), for the *rankPotentialKSC* both $S2$ and $S5$ get the same value of n. Only a subsequent analysis of the number of learning outcomes matched by the selected supplies (i.e., 3 for $S2$ and 4 for $S5$) could identify as best option $S5$ and then assign to it the highest ranking, since only the competence does not fully match the requirements of D, whereas both skills and knowledge satisfy them.

DEMAND: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)))) CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)) STATICWEBPAGES DYNAMICWEBPAGES)))

SUPPLY1: (and (all requires (and SOMEAUTONOMY (all hasActionVerb (and PROGRAM (all actsOnKnowledge PHP))) PROGRAM (all actsOnKnowledge PHP) PHP)))

SUPPLY2: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and PROGRAM (all actsOnKnowledge HTML))) PROGRAM (all actsOnKnowledge HTML) HTML)))

SUPPLY3: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and DEBUG (all actsOnKnowledge (and HTML PHP)))) DEBUG (all actsOnKnowledge (and HTML PHP)) HTML PHP)))

SUPPLY4: (and (all requires (and SOMEAUTONOMY (all hasActionVerb (and CREATE (all actsOnKnowledge WEBPAGES))) CREATE (all actsOnKnowledge WEBPAGES) WEB-PAGES)))

SUPPLY5: (and (all requires (and SOMEAUTONOMY (all hasActionVerb (and PROGRAM (all actsOnKnowledge (and HTML ASP)))) PROGRAM (all actsOnKnowledge (and HTML ASP)) HTML ASP)))

SUPPLY6: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and PROGRAM (all actsOnKnowledge WEBPAGES))) PROGRAM (all actsOnKnowledge WEBPAGES) WEB-PAGES)))

SUPPLY7: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and DEBUG (all actsOnKnowledge WEBPAGES))) DEBUG (all actsOnKnowledge WEBPAGES) WEBPAGES)))

Figure 3: Normal form for demand and supplies consider in the experimental evaluation

Supply	**rankPotentialKSC**	**rankPotential**	**t**
S1	15	6	2
S2	10	3	3
S3	16	6	2
S4	20	8	0
S5	10	4	4
S6	20	6	0
S7	20	8	0

Table 1: Experimental results on demand and supplies considered in the experimental evaluation

A further case, which shows how considering a whole learning outcome instead of analyzing each single element separately could provide better results, is revealed by $S6$, for which the two algorithms provide different rankings: according to the *rankPotential* algorithm, $S6$ should be ranked in third position whereas, based on the *rankPotentialKSC* algorithm it should get the lowest ranking (as it does not satisfy any requirement). This result is coherent with the considerations in Section 3, as it demonstrates that KO, AV and CX elements should not be considered as separate items, since any of them influences the others.

Moreover, it is worth remarking that the algorithm allows the user to make a distinction between the lack of a detailed concept and of a more generic one. In fact, the more a concept is specific, the heavier his absence will affect the final result. This behavior is consistent with the monotonicity requirement, since more specific concepts are implicitly assigned highest weights.

While the results obtained with the *rankPotential* algorithm give a first idea of the distance between a curriculum and the demand, the *findIrred* algorithm highlights which elements should be added to a curriculum, in order to make the applicant a possible candidate for the job; in other terms, it gives information about which knowledge, skill and competence elements the applicant should deepen in order to get employed.

Figure 4 reports the results obtained when the *findIrred* algorithm is executed on the demand and supplies presented in Figure 3. It is worth remarking that, for readability reasons, the normal form has been adopted for the presentation of the results. It is easy to see that the rank provided by the *rankPotentialKSC* algorithm is substantially confirmed, since $S5$ remains the best option (in facts it is only lacking with respect to competences, while it satisfies the requirements on skills and knowledge), followed by $S2$ (which lacks the learning outcomes linked to the *dynamic web pages* concept), $S1$ (for which two competences, one skill and one knowledge should be deepened by the applicant who wants to be engaged), and $S3$ (in which only the two knowledge elements related to *static* and *dynamic web pages* match the demand). Finally, $S4$, $S6$ and $S7$ obtain the worst rank, since they do not match any requisite expressed by the demand.

6 Conclusion

In this paper, an adaptation of the *rankPotential* algorithm that has been introduced in [Di Noia et al. 2004] and of the *findIrred* method that has been discussed in [Di Noia et al. 2003] is presented. The reference algorithms have been modified to take into account the EQF specifications. The core EQF components, such as knowledge, skills and competences, have been identified by making reference to several sub-elements, like knowledge objects, action verbs and context. The algorithms have been integrated in a Web portal designed to support matchmaking in the European dimension, by focusing on both the occupational and educational perspectives.

The designed algorithms could be effectively used in any mobility scenario, as well as in any lifelong learning context relying on transparency and readability of qualifications.

SUPPLY1: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)))) CREATE (all actsOnKnowledge (and STATICWEBPAGES)) STATICWEBPAGES)))

SUPPLY2: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and CREATE (all actsOnKnowledge (and DYNAMICWEBPAGES)))) CREATE (all actsOnKnowledge (and DYNAMICWEBPAGES)) DYNAMICWEBPAGES)))

SUPPLY3: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)))) CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)))))

SUPPLY4: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)))) CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)) STATICWEBPAGES DYNAMICWEBPAGES)))

SUPPLY5: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)))))))

SUPPLY6: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)))) CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)) STATICWEBPAGES DYNAMICWEBPAGES)))

SUPPLY7: (and (all requires (and FULLAUTONOMY (all hasActionVerb (and CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)))) CREATE (all actsOnKnowledge (and STATICWEBPAGES DYNAMICWEBPAGES)) STATICWEBPAGES DYNAMICWEBPAGES)))

Figure 4: Result of the *findIrredKSC* algorithm on the demand and supplies considered in the experimental evaluation

Acknowledgements

This work was partially supported by the TIPTOE "Testing and Implementing EQF and ECVET Principles in Trade Organizations and Education" project funded under the Lifelong Learning Programme – Leonardo da Vinci (NL/08/LLP-LdV/TOI/123011).

References

[Baader et al. 2002] Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D. and Patel-Schneider, P. (eds.): "The Description Logic Handbook"; Cambridge University Press. 2002.

[Borgida et al. 1989] Borgida, A., Brachman, R., McGuinness, D., Alperin Resnick, L.: "CLASSIC: a structural data model for objects"; Proc. of the 1989 ACM SIGMOD international conference on Management of data, Oregon, (June 1989), 58-67.

[Borgida et al. 1989] Borgida, A. and Patel-Schneider, P. F.: "A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic"; Journal of Artificial Intelligence Research (1995), Volume 1, (1989), 277-308.

[Colucci et al. 2007] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., Ragone, A., Trizio, M.: "Semantic-based Search Engine for Professional Knowledge"; Proc. 7^{th} Int. Conf. on Knowledge Management and Knowledge Technologies (I-KNOW 2007), (Sep 2007), 472-475.

[Di Noia et al. 2004] Di Noia, T. ,Di Sciascio, E., Donini, F., Mongiello, M: "A System for Principled Matchmaking in an Electronic Marketplace"; International Journal of Electronic Commerce, Volume 8 , Issue 4, (2004), 9-37.

[Di Noia et al. 2003] Di Noia, T. ,Di Sciascio, E., Donini, F., Mongiello, M: "Abductive matchmaking using description logics"; Proc. 18^{th} Int. Joint Conf. on Artificial Intelligence (IJCAI), (August 2003), 337-342.

[EQF, 2008] The European Qualifications Framework for Lifelong Learning (EQF) `http://ec.europa.eu/education/lifelong-learning-policy/doc44_en.htm`.

[Garro and Palopoli, 2003] Garro, A. and Palopoli, L.: "An xml multi-agent system for e-learning and skill management"; Agent Technologies, Infrastructures, Tools, and Applications for E-Services, Volume 2592 of LCNS (2003), 283-294.

[Lei and Horrocks, 2004] Lei, L. and Horrocks, I.: "A Software Framework for Matchmaking Based on Semantic Web Technology"; International Journal of Electronic Commerce, Volume 8, Issue 4 (2004), 39-60.

[Lau and Sure, 2002] Lau, T. and Sure, Y.: "Introducing ontology-based skills management at a large insurance company"; Proc. of the Modellierung, (2002).

[Nemirovskij et al. 1999] Nemirovskij, G., Bugiel, P. and Heuel, E: "From Semantic Document Annotation to Global Search Facilities for Personalized Study Programmes"; World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, (2007) 1164-1171.

[Pernici et al. 2006] Pernici, B. Locatelli, P. Marinoni, C.: "The eCCO System: An eCompetence Management Tool Based on Semantic Networks"; Lect. Notes Comp. Sci. 2006, Springer, Berlin.

[Pöyry and Puustjärvi, 2002] Pöyry, P. and Puustjärvi, J.: "The role of metadata in the CUBER system"; 2002 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement Through Technology, (2002) 172-178.

[Ragone et al. 2005] Ragone, A., Coppi, S., Di Noia, T., Di Sciascio, E., Donini, F.: "Natural Language Processing for a Semantic Enabled Resource Retrieval Scenario"; 27^{th} Intl. Conf. on Information Technology Interfaces, (2005) 395-401.

[The Copenhagen Declaration, 2002] The Copenhagen Declaration - Declaration of the European Ministers of Vocational Education and Training, and the European Commission, 2002, `http://ec.europa.eu/education/pdf/doc125_en.pdf`.