Journal of Universal Computer Science, vol. 9, no. 12 (2003), 1437-1454
submitted: 29/8/03, accepted: 18/9/03, appeared: 28/12/03 © *J.UCS*

A Formal Approach to Ontology-Based Semantic Match of Skills Descriptions[1]

Simona Colucci
(Politecnico di Bari, Bari, Italy,
s.colucci@poliba.it)

Tommaso Di Noia
(Politecnico di Bari, Bari, Italy,
t.dinoia@poliba.it)

Eugenio Di Sciascio
(Politecnico di Bari, Bari, Italy,
disciascio@poliba.it)

Francesco M. Donini
(Università della Tuscia, Viterbo, Italy,
donini@unitus.it)

Marina Mongiello
(Politecnico di Bari, Bari, Italy,
mongiello@poliba.it)

Marco Mottola
(Accenture, Roma, Italy,
marco.mottola@accenture.com)

Abstract: Skills management has been recently acknowledged as one of the key factors to adequately face the increasing competitiveness between knowledge intensive companies.

In this paper we present a formal approach to Ontology-Based Semantic Matchmaking between Skills demand and supply, devised as a virtual marketplace of knowledge. In such a knowledge market metaphor, skills are a peculiar kind of good that has distinguishing characteristics with respect to traditional assets. Buyers are entities that need the skills of people, such as projects, departments and organizations; sellers are workers that offer their own skills.

The formal framework supports the semantic match of descriptions provided by demanders and sellers of skills. In particular our approach, based on Description Logics formalization and reasoning, overcomes simple subsumption matching and allows match ranking and categorization. The implementation of the approach in a prototype system, which embeds a NeoClassic reasoner, is also described.

[1] A short version of this article was presented at I-Know '03 (Graz, Austria, July 2-4, 2003)

Key Words: Knowledge Representation, Skill Management, Matchmaking, Ontologies
Category: I.2.4, K.6.1

1 Introduction

Knowledge Management provides methods and tools to increase the know-how of competitive companies. Its main focus is to strategically capture and make accessible knowledge and expertise of individuals within and across companies. The individual is then able to share such expertise and knowledge with the whole organization. As intellectual capital has become one of the most strategic assets of successful organizations in the last decade, the capability of managing the expertise, skills and experience of people represents a key factor to face the increasing competitiveness of the global market.

In the definition of the strategy of knowledge-intensive organizations, emphasis should be given to capitalization of the knowledge acquired during their business since this has shown to improve the return on investment.

Such capitalization calls for the assessment of the knowledge states of individuals [Stefanutti and Albert 2003].

Besides, the information systems of the organizations should provide access to knowledge, so emphasis should be given to skills mining and knowledge sharing through the network.

In this way the expert finding problem can be considered under an information retrieval perspective, where the goals to pursue are the information and expertise finding. The methods actually adopted in experts finding processes could be improved by tracing knowledge -both held by persons and explicitly documented- and by enhancing the visibility of non documented information.

Nevertheless, such approaches have some drawbacks due to the information overloading that occurs when too much information is available about a given topic and the user has to filter out the proper results satisfying his information needs. Problems deriving from the unstructured nature of web data are currently being faced in several fields through approaches supported by the Semantic Web initiative that has begun to revolutionize the way information is provided on the Internet. Also, languages based on the Semantic Web foundations have been developed that allow the representation of machine-understandable description of web content through the creation of domain ontologies useful in supporting interoperability in the web environment.

Obviously, skill management systems should be able to efficiently deal with cases in which profiles only approximatively match some given requirements. To this aim knowledge has to be modelled and structured and in recent years ontologies have been proposed as the best way to obtain this [Sure et al. 2000], [OLeary 1998], [Staab et al. 2001], [Dieng et al. 1999], [Guarino 1998].

Ontological frameworks have been proposed also as group memory systems for managing corporate internal competencies in Knowledge Intensive Organizations [de Vasconcelos et al. 2003].

An issue that arises is *using* ontologies once they have been built, *i.e.*, there is a need for reasoners and reasoning services able to take full advantage of the effort placed in structuring an ontology. Reverting to plain information retrieval techniques would make the whole conceptualization effort almost useless, while full match between skills requested and available is usually rare.

In this paper we present a semantic based approach to the problem of skills finding in an ontology supported framework. Our framework considers skill management as an electronic marketplace of knowledge in which skills are a peculiar kind of goods that have distinguishing characteristics with respect to traditional assets; buyers are entities that need the skills of people, such as projects, departments and organizations. On the other hand, knowledge sellers are individuals that offer their own skills. Obviously, descriptions of profiles share a common skills ontology.

Although semantic facilitators have been proposed in the literature for several scenarios [Trastour et al. 2002], [Sure et al. 2000], [Staab et al. 2001], they do not take full advantage of the ontological structure and limit their search to simple subsumption matching.

Our approach, based on Description Logics formalization and reasoning, is oriented to finding the best individual for a given task or project, based on profile descriptions sharing a common ontology. The approach is able to cope with cases in which no perfect matches exist, *i.e.*, finding those available profiles that, for a given skill request best match, also if not identical, and vice versa. In particular we logically distinguish cases in which some skills in a request profile are not specified in the offered one, yet there is no contradiction and *e.g.*, further inquiries can be done (what is called a *potential match*); cases in which some skills in the request are in contrast with the given profile (what is called a *partial match*); in this case the one who is carrying out the search may check for unsatisfiable requests and eventually retract them if no better choice is at hand. It is noteworthy that our approach allows not only a logical categorization, but also a ranking of matches within each category. Notice that a *total match* is hence just a special case of a *potential* match.

The remaining of the paper is organized as follows. In the next section we briefly revise, to make the paper self-contained, basic concepts of Description Logics, in strict correlation with the logic of the Classic system we adapted in our system. Then we formalize some issues typical of skill management, pointing out the particular scenario in which we place our approach and highlighting properties that should hold in a semantic approach to skill finding. We present algorithms used for matching skills in a logic based way in Section 4. We describe our system and its features in Section 5. Last section draws the conclusions and outlines directions for future research.

2 Description Logics

DLs [Borgida 1995], [Donini et al. 1996] are a family of logic formalisms whose basic syntax elements are *concept* names, *e.g.*, `person`, `degree`, `specialization`, and *role* names, such as `workingIn`, `requiredAS`. Intuitively, concepts stand for sets of objects, and roles link objects in different concepts.

Formally, concepts are interpreted as subsets of a domain of interpretation Δ, and roles as binary relations (subsets of $\Delta \times \Delta$). Basic elements can be combined using *constructors* to form concept and role *expressions*, and each DL has its distinguished set of constructors.

Every DL allows one to form a *conjunction* of concepts, usually denoted as $\sqcap$; some DL include also disjunction $\sqcup$ and complement $\neg$ to close concept expressions under boolean operations. Roles can be combined with concepts using *existential role quantification*, *e.g.*, $\mathtt{Graduate} \sqcap \exists \mathtt{hasDegree.Engineering}$, which describes the set of graduates with an engineering degree, and *universal role quantification*, *e.g.*, $\mathtt{person} \sqcap \forall \mathtt{livingIn.Apulia}$, which describes persons living exclusively in Apulia. Other constructs may involve counting, as number restrictions: $\mathtt{Person} \sqcap (\leq\ 1\ \mathtt{hasDegree})$ expresses persons with at most one degree, and $\mathtt{Person} \sqcap (\geq\ 3\ \mathtt{hasSpecialization})$ describes persons endowed of at least three specializations. Many other constructs can be defined, increasing the expressive power of the DL, up to n-ary relations [Calvanese et al. 1998].

Concept expressions can be used in *inclusion assertions*, and *definitions*, which impose restrictions on possible interpretations according to the knowledge elicited for a given domain. For example we could impose that working teams members may be divided into those belonging to internal personnel and consultants using the two inclusions $\mathtt{TeamMember} \sqsubseteq \mathtt{InternalPersonnel} \sqcup \mathtt{Consultant}$ and $\mathtt{InternalPersonnel} \sqsubseteq \neg \mathtt{Consultant}$. Or that working teams have at least two members as $\mathtt{Team} \sqsubseteq (\geq\ 2\ \mathtt{hasTeamMember})$. Historically, sets of such inclusions are called TBox (Terminological Box). The basic reasoning problems for concepts in a DL are satisfiability and subsumption relatively to a TBox, which accounts for the more general/more specific relation among concepts, that forms the basis of a taxonomy.

More formally, a concept C is satisfiable if there exists an interpretation in which C is mapped into a nonempty set, unsatisfiable otherwise.

If a TBox T is present, satisfiability is relative to the models of T, that is, the interpretation assigning C to a nonempty set must be a model of the inclusions in T. For instance, the concept $\mathtt{Member1} \sqsubseteq \mathtt{InternalPersonnel} \sqcap \mathtt{Consultant}$ is clearly unsatisfiable w.r.t. the TBox containing the inclusion

`InternalPersonnel` $\sqsubseteq$ ¬`Consultant`. Also a TBox can be said satisfiable if there exist at least one model (i.e., an interpretation fulfilling all its inclusions in a nontrivial way).

A concept C subsumes a concept D if every interpretation assigns to C a subset of the set assigned to D.

Also Subsumption is usually established relative to a TBox, a relation that we denote $T \models CD$. It is important to note that in the CLASSIC system we use, each C concept has an equivalent normal form as $C_{names} \sqcap C_{\sharp} \sqcap C_{all}$, in which C_{names} is a conjunction of names, $C_{\sharp}$ of number restrictions, and C_{all} of universal role quantifications. In the normal form, also all inclusions, definitions and disjoint groups have been made explicit [Borgida and Patel-Schneider 1994]. CLASSIC provides the two basic reasoning services of DL-based systems, namely *Concept Satisfiability* (given a TBox T and a concept C, does there exist at least one model of T assigning a non-empty extension to C?), and *Subsumption* (given a TBox T and two concepts C and D, is C more general than D in any model of T?). Being a complete KR system, CLASSIC provides also data types as numbers and strings, and other services which are useful in a deployed prototype.

3 Formalization of Skill Matching Issues

Skill matching is affected by a number of factors. The choice of how to manage these factors may determine different approaches and lead to the search for different solutions. Some of these factors are independent on the particular kind of good we analyze for the match and the choice of how to manage them affects all matchmaking scenarios. Other factors are typical of the matching of skills and can be neglected in other contexts.

We outline in the following some of the factors we believe characterize skill matching scenarios and then focus on our particular setting.

Negative Information treatment

This factor affects the choice of the language in which descriptions have to be expressed and is fundamental in the matching process of any kind of description. We may itemize possibilities as follows:

a) *absent*: all information allowed in profile descriptions are positive and all others are considered unknown.

b) *implicit*: lacking information in a description are implicitly managed as negative.

c) *explicit*: negative information can be elicited in descriptions together with positive ones, but all not elicited information are considered unknown.

Notice that considering negative information as absent or implicit in a profile description, as is usually done in databases, may result quite limiting. Instead the absence of a characteristic in the description of a profile should not be interpreted as a constraint of absence but as an item that can be either refined later, or left unknown if irrelevant for a user, what is usually called *open-world assumption* in Knowledge Representation.

Multiplicity of Relationship between Individuals and Tasks

This issue is typical of the skill matching process, because in the matching of other kinds of good the multiplicity is always *one to one*. We have for example a demand describing *one* particular good and we search for *one* supply fulfilling the demand. When turning to skill matching, instead, one offered profile may be assigned more than one task and viceversa.

Match relationship between Individuals and Tasks may be characterized by a multiplicity:

a) *one to one*: we have *one* job profile to match with *one* individual; offered and requested profile descriptions may be relative to more than one skill. The scenario is typical of temporary work agencies or counseling companies, in which one person is employed if s/he is able to attend one task.

b) *many to one*: we have *one* task to assign to *several* people. This happens for example in the selection of a working team for a project, representing in this case the task to assign. For this case, each person is assigned no more than one task.

c) *one to many*: we search for *one* individual attending to *many* simple tasks. The scenario is similar to time-sharing in Operating Systems, in which we have one resource to share between several users. In this context many tasks share the same human resource and several constraints may ensue.

d) *many to many*: we have *many* tasks to assign and *many* individuals available and we have to search for the best scheduling of human resources on the different tasks.

In this work we concentrate on one-to-one skill matching and highlight some intuitive properties that a semantic approach should take into account. First of all notice that we make the *open-world assumption*. The rationale of this approach can be highlighted with the help of an example. Let S be an offered profile, describing a *C Programmer*. If a recruiter is searching for an *Engineer able to program in C Language*, the candidate described by S should be considered for the assessment of the match. It is not said that the candidate is not an Engineer because nothing has been specified about his degree. Obviously, the algorithm employed for matchmaking should take this issue into account.

Secondly, a matchmaking system may give different evaluations depending on whether it is trying to match a request S with an offer D, or D with S — *i.e.*, depending on *who* is going to use this evaluation.

This requirement is already evident when characteristics are modeled as sets of words. For example, let a programmer skills Demand D be schematically represented as $D = \{C++, TCP/IP, SQL\}$ and let an available profile (a Supply in our framework) S be $S = \{Javascript, TCP/IP, SQL, VBScript\}$. Then $D - S$=C++ represents the missing skills in the match of D and S while $S - D = \{Javascript, VBScript\}$ represents additional skills not needed by D: in that case, underconstrained requirements of S from the point of view of D are expressed by $D - S$ (set difference) while underconstrained requirements of D from S's viewpoint are expressed as $S - D$. Of course, using sets of words to model supplies and demands would be too sensible to the choice of words employed — it misses meanings that relate words. It is now a common opinion that such fixed-terminology problems are overcome if terms have a logical meaning through an ontology, *i.e.*, a specific vocabulary used to describe a certain reality plus a set of explicit assumptions regarding the intended meaning of the vocabulary words [Fensel et al. 2001]. Hence, we assume that supplies and demands are expressed in a DL. Obviously this approach includes the sets-of-keywords one, since a set of keywords can be considered also as a conjunction of concept names. We assume also that the common ontology is established, as a TBox in DL. Now a match between a supply S and a demand D could be evaluated according to T. Let $T \models \ldots$ denote logical implication (truth in all models of T), and let $\sqsubseteq$ (subsumption) denote also implication between constraints of S and D. We introduced in Section 1 the distinction we make between potential and partial matches. In order to evaluate both kinds of match we have to use three relations between concepts, highlighted in the following.

For the assessment of *potential* match we have to check that no constraint imposed by D is excluded by S and vice versa. Formally we have to assess if $D \sqcap S$ is satisfiable in T, checking for **Consistency**. For example, the demand D asking for an *engineer, required as Javascript programmer* and the supply S describing a *programmer expert about SQL* represent a potential match. This relation has been highlighted also by other researchers [Trastour et al. 2002]. However, that proposal lacks a *ranking* between different potential matches, which we believe is fundamental in order to support *e.g.*, a project manager in the choice of the most interesting curricula, among all potential ones.

We are also interested in evaluating a *partial* match when some constraints of one proposal are in contrast with the properties of the other one. The relation between concepts describing this situation is **Inconsistency**, characterized by the fact that $D \sqcap S$ is unsatisfiable in T. For example, let D be yet the demand asking for an *engineer, required as Javascript programmer* and let S describe the profile of a *Javascript programmer with a degree in Economics*. Then S is inconsistent with D. This situation seems to be not interesting for a match evaluation, but it becomes important if the recruiter is allowed to free D by constraints in contrast with the available S. Of course D has to be not too far from the original request. Then it arises the need to rank also partial matches, even if their ranking function is different from the one used for potential matches.

Finally we model the *exact or total* match situation. In this case D and S should be considered equivalent in T. To this aim every constraint imposed by D has to be fulfilled (implied) by S and vice versa. This fulfillment is formalized by **Implication**. We state that S implies D by writing $T \models (D \sqsubseteq S)$ ($T \models (S \sqsubseteq D)$ if D implies S). For example, if D is a demand asking for a *C++, TCP/IP, SQL expert* and S is a supply describing a *TCP/IP, SQL expert*, this corresponds to $T \models (D \sqsubseteq S)$.

Both in potential and in partial match we underlined the importance of a ranking function. In the following we point out some properties we believe every ranking function should have.

A ranking for semantic matchmaking should be *syntax independent*. That is, for every pair of supplies S_1 and S_2, demand D, and ontology T, when S_1 is logically equivalent to S_2 then S_1 and S_2 should have the same ranking for D. — and the same should hold also for every pair of logically equivalent demands D_1, D_2 with respect to every supply S. For example, suppose we have S_1 describing a *Part-time worker* and S_2 describing a *Person working four hours a day* and suppose the TBox specifies that every person working between 1 and 6 hours a day is a part-time worker, then the two supplies have to be ranked the same w.r.t. every demand.

Besides, a ranking for semantic matchmaking should be *monotonic over subsumption*: for every demand D, for every pair of supplies S_1 and S_2, and ontology T, if S_1 and S_2 are both potential matches for D, and $T \models (S_2 \sqsubseteq S_1)$, then S_2 should be ranked either the same, or better than S_1.

S_2 should be ranked better than S_1 if the Demand D asks for the characteristics of S_2 not implied by S_1. Otherwise, if the characteristics of S_2 not implied by S_1 are not required by the demand, S_1 and S_2 should be ranked the same. We show this property with the help of an example: let D be the demand for a *Java programmer expert in SQL*. Suppose now to have the following supplies: S_1 describing a *TCP/IP and SQL expert*, S_2 proposing a *C++ programmer expert in TCP/IP and SQL* and S_3 offering a *Java programmer expert in TCP/IP and SQL*. Notice that both S_2 and S_3 imply S_1. But S_2 should be ranked the same as S_1, while S_3 should be ranked better than S_1.

The property should hold also for every pair of demands D_1, D_2 with respect to a supply S. Intuitively, monotonicity over subsumption could be read of as "A ranking of potential matches is monotonic over subsumption if the more specific, the better." Observe that we use the word better instead of using any symbol $\geq, \leq$. This is because some rankings may assume that better=increasing (toward infinity, or 1) while others may assume better=decreasing (toward 0).

When turning to partial matches, a property complementary to the last one described should yield. In fact, in the evaluation of partial match, ranking is a measure of the number of characteristics in the supply contrasting demand constraints. The situation deals with unsatisfactory proposals. So, adding another characteristic to an unsatisfactory proposal may either worsen its ranking (when another characteristic is violated) or keep it the same (when the new characteristic is not in contrast). This behavior is formalized by *antimonotonicity over implication*: for every demand D, for every pair of supplies S_1 and S_2, and ontology T, if S_1 and S_2 are both partial matches for D, and $T \models (S_2 \sqsubseteq S_1)$, then S_2 should be ranked either the same, or worse than S_1. Let D be, for example, the demand asking for a *Java programmer as internal team member of a Company placed in the South of Italy.* Consider now the following supplies: S_1 offering a *Consultant, TCP/IP expert*, S_2 describing a *Consultant, expert in C++ programming and TCP/IP* and S_3 proposing a *Consultant, working in the North of Italy, expert in TCP/IP.* Both S_2 and S_3 imply S_1. But S_2 should be ranked the same as S_1, while S_3 should be ranked worse than S_1, because it has one further characteristic in contrast with D.

Obviously, properties pointed out here are independent of the particular DL employed, or even the particular *logic* chosen.

4 Matching Algorithms

The algorithms for skill matching have been devised adapting the original Classic structural algorithm for subsumption [Borgida and Patel-Schneider 1994]. We have two different, though similar, algorithms for potential and partial matching.

The algorithm for potential match is outlined in Figure 1.

A Classic concept C can be put in normal form as $C_{names} \sqcap C_{\sharp} \sqcap C_{all}$. Without ambiguity, we use the three components also as sets of the conjoined concepts. Furthermore, as the TBox in Classic can be embedded into the concepts, we do not consider explicitly the TBox, although it is obviously present. The algorithm takes as inputs two descriptions to be matched *i.e.*, D –the requested profile– vs. C the available profile, in normal form, such that $C \sqcap D$ is satisfiable and returns a rank $n \geq 0$ of C w.r.t. D, where 0 corresponds to total match. The algorithm adds to n the number of concept names in D that are not among the concept names of C and number restrictions of D that are not implied by those of C and for each universal role quantification in D adds to n the result of a recursive call.

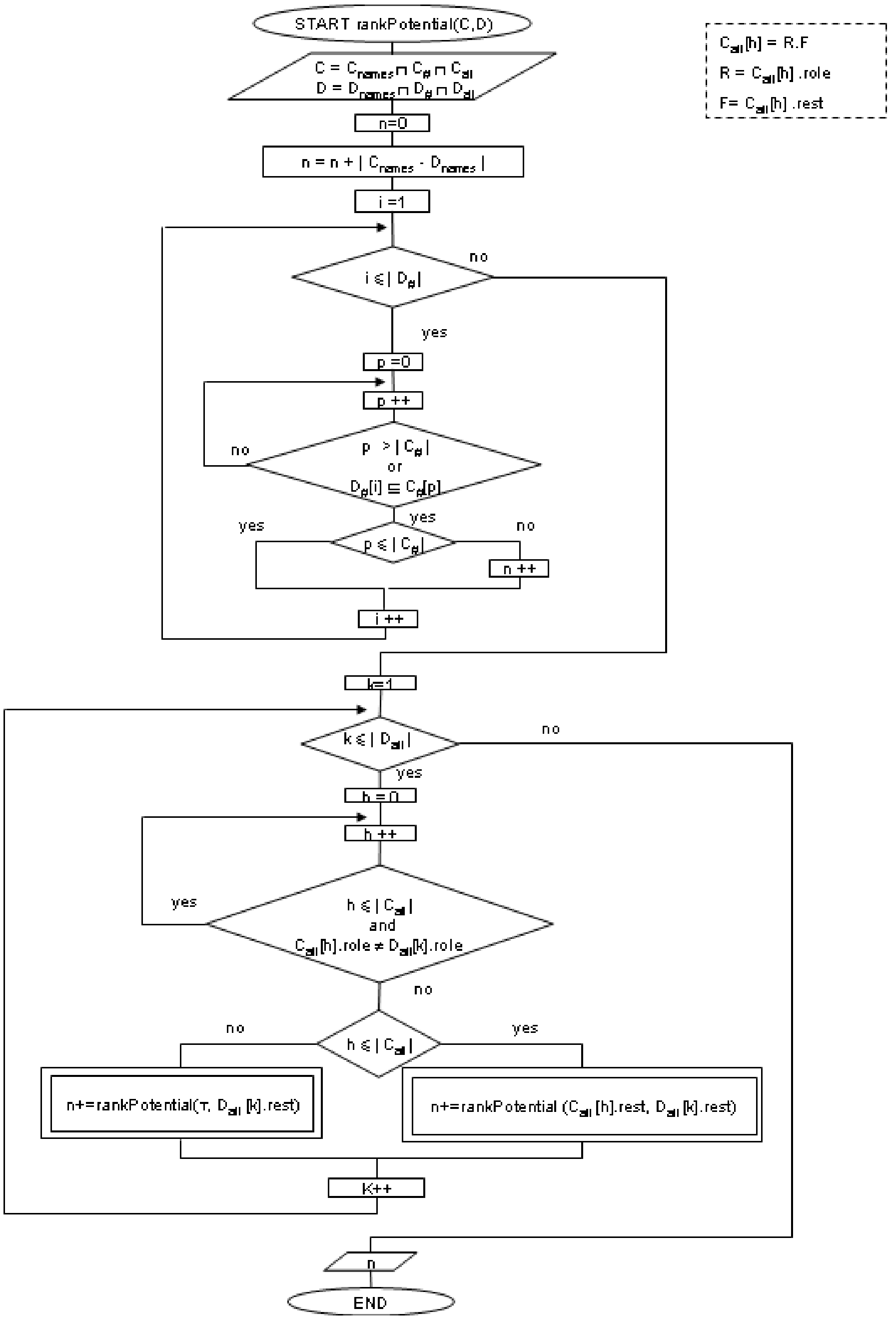

Figure 1: *FlowChart of the Potential Match Algorithm*

Looking at the algorithm it is simple noticing that a total match, *i.e.*, concept implication, yields a 0 ranking, and the ranking increases (worsens) as the two profile descriptions are, though still compatible, more different. Also notice the rationale of the approach, which penalizes generic profile descriptions, which in simple subsumption matching would be unfairly advantaged, as the algorithm ranks better more specific descriptions of C matching D.

With reference to the computational complexity, which is extremely important for a practical use of the system, the expansion of the TBox in the construction of the normal form can lead to an exponential blow-up, as demonstrated by Nebel [Nebel 1990]. Obviously a polynomial algorithm cannot be expected since subsumption in $\mathcal{AL}$ with an acyclic TBox $\mathcal{T}$ is co-NP-hard [Calvanese 1996]. However, Nebel also argues that the expansion is exponential in the depth of the hierarchy $\mathcal{T}$, yet if the depth of $\mathcal{T}$ is $O(\log |\mathcal{T}|)$, then the expansion is polynomial, and so is our algorithm [Di Noia et al. 2003a].

The algorithm can be modified so that weights on subconcepts of D are taken into account: instead of adding 1 to n for each D's concept missing in C, one just adds the corresponding weight. Then, a far rank would mean that either many minor characteristic, or a very important one, are left unspecified in C. We implemented also a version of the algorithm in which weights are *learned* by the system, upon repeated analysis of proposals. In this case, of course, the learned weights are *absolute* ones, and not relative to a particular actor.

The algorithm for ranking partial matches follows again the partition of Classic concepts into names, number restrictions, and universal role quantifications. However, this time we are looking for inconsistencies. Hence, when a universal role quantification is missing in either concept, the recursive call is unnecessary.

Also in this case weights can be added to subconcepts of D, where the greater the weight, the more that characteristic is important, making the rank of C far off when in contrast.

For both matching algorithms it can be proved they respect the properties highlighted in the previous section.

5 Skill Matching System

The matchmaking framework presented in the previous sections has been deployed in a prototype facilitator originally designed for a Peer to Peer electronic marketplace [Di Noia et al. 2003b]. Our matching engine is based on Java servlets; it embeds the adapted version of the NeoClassic reasoner and communicates with the reasoner running as a background daemon. The system receives a Knowledge Representation System Specification (KRSS) string containing the query profile description (either a Demand or a Supply profile) and the URI referencing the skills ontology.

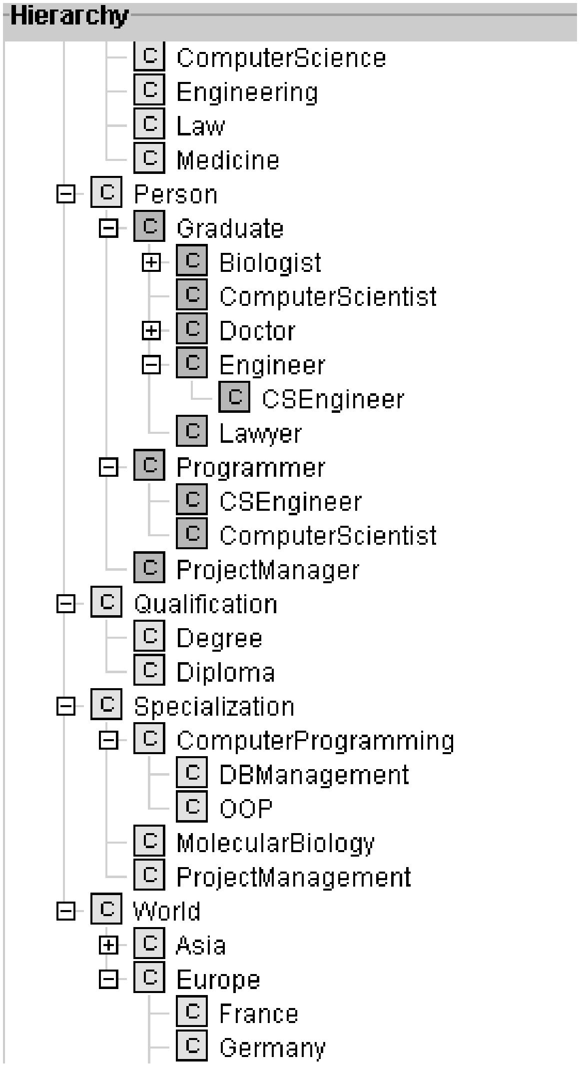

Figure 2: *Hierarchy of concepts in the skills ontology.*

Our ontology currently is endowed of approximately seventy concepts and is still being expanded.

In recent years several methodologies have been proposed for ontologies design, see [Jones et al. 1998] for a survey. The methodology we used to generate the ontology based on the one proposed in [Uschold and Gruninger 1996].

A portion of the ontology hierarchy is pictured in Figure 2. Figure 3 shows a snapshot of the Oiled interface used in the construction. The ontology has

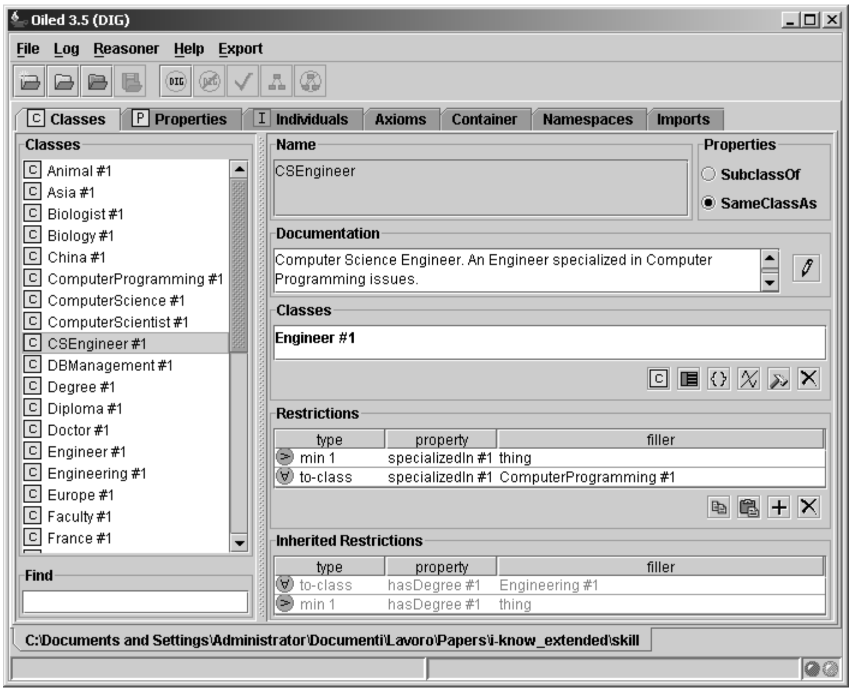

Figure 3: *Oiled user interface. The left side shows the list of classes; the right side shows inherited restrictions for class engineer.*

been obviously translated in Classic and Figure 4 pictures a portion of it using Classic syntax.

The operating mode is as follows: the Reasoner checks the description for consistency; if it fails, based on the Reasoner output, the system provides an error message stating the error occurred. Otherwise the proper matchmaking process takes place. Each match can return a 0, which means total match or a value > 0. Recall that returned values for partial matches and potential matches have logically different meaning and matching descriptions are sorted in different sets. Up to three disjoint result sets may then ensue.

As an example of the advantages of using our approach, let us consider in the following simple scenario the system behavior w.r.t. a simple text-based one. Suppose to have the Demand profile description *Looking for an engineer, living in Europe, required as OOP programmer for a work in Europe* and to set as search key words S= { engineer, living, required, OOP, programmer, work, Europe }. Also suppose that a set of possible profiles, shown in Figure 5 is to be analyzed with respect to the Demand.

Comparing the requested skills with the available set by simple text analysis

```
(createRole hasDegree) (createRole workingIn)
(createRole livingIn) (createRole specializedIn)
(createRole requiredAs)
(createConcept Qualification TOP true)
  (createConcept Degree Qualification true)
  (createConcept Diploma Qualification true)
(createConcept Faculty TOP true)
  (createConcept Engineering Faculty true)
  (createConcept Law Faculty true)
  (createConcept Biology Faculty true)
  (createConcept Medicine Faculty true)
  (createConcept ComputerScience Faculty true)
(createConcept Specialization TOP true)
  (createConcept ComputerProgramming Specialization spec)
    (createConcept OOP ComputerProgramming true)
    (createConcept DBmanagement ComputerProgramming true)
  (createConcept MolecularBiology Specialization spec)
  (createConcept ProjectManagement Specialization spec)
(createConcept World TOP true)
  (createConcept Europe World continent)
    (createConcept Italy Europe country)
    (createConcept France Europe country)
    (createConcept Germany Europe country)
    (createConcept Swiss Europe country)
    (createConcept Asia World continent)
    (createConcept China Asia country)
    (createConcept Japan Asia country)
(createConcept Person TOP true) (createConcept Animal TOP true)
(createConcept Graduate (and Person (at-least 1 hasDegree)))
  (createConcept Engineer (and Graduate (all hasDegree Engineering)))
    (createConcept CSEngineer (and Engineer (at-least 1 specializedIn)
                               (all specializedIn ComputerProgramming)))
(createConcept Biologist (and Graduate (all hasDegree Biology)))
  (createConcept MolecularBiologist (and Biologist
      at-least 1 specializedIn)
                                 (all specializedIn MolecularBiology)))
(createConcept Lawyer (and Graduate (all hasDegree Law)))
(createConcept Doctor (and Graduate (all hasDegree Medicine)))
  (createConcept Veterinarian (and Doctor (at-least 1 specializedIn)
                                         (all specializedIn Animal)))
(createConcept Programmer (and Person (at-least 1 specializedIn)
                        (all specializedIn ComputerProgramming)))
(createConcept ComputerScientist (and Graduate Programmer
                         (all hasDegree ComputerScience)))
(createConcept ProjectManager (and Person (at-least 1 specializedIn)
                             (all specializedIn ProjectManagement)))
```

Figure 4: *Part of Classic description of the test skill ontology*

we get the following ordered result set $R = \{S_2 - S_1, S_4, S_3 - S_5 - S_6\}$ in which S_2, S_1 and S_3, S_4, S_5 have the same rank.

Using our system w.r.t. our reference ontology we have the following ranked list $R = \{S_2, S_1, S_3, S_5, S_4 - S_6\}$ in which S_2, S_1, S_3, S_5 potentially match the demand and S_4, S_6 partially.

It obviously appears that our semantic approach provides results that are quite close to the choice a human user would do. Also notice that, with reference to supply5, which is generic, but not in contrast –hence a potential match– with the Demand description, the system ranks it lowest among potential matches. Obviously a *head-hunter* unable to find among higher ranking profiles an adequate candidate may then carry out further inquiries on the doctoral degree of the individual in supply5.

Our system currently provides the following services:

1. Support to the user in the data insertion and query submission. The user is incrementally guided in the definition of a profile.

2. Automatic construction and verification of consistency w.r.t. the reference ontology of the profile.

3. Deduction of new knowledge on the basis of available data.

4. Ability to provide ranked conceptually approximate answers, *i.e.*, near miss or partial match, in the presence of unsatisfiable queries.

5. Ability to provide ranked potential matches and possibility to ask for unforeseen (hence not immediately available) features to the supplier, with successive automatic update of description and communication of update.

6. Storage of satisfiable demands or supplies that were still unmatched, with automatic reexamination when new supplies are provided, and notification on successful match between supply and demand. The same service is available for unmatched supplies.

At the current stage of the project our ontology is still a small one. Nevertheless we have started evaluating, with the help of some volunteers, the degree of conformance of the system response to users' perception in terms of matches categorization and especially in terms of ranking.

The experiments carried out so far have been made selecting a set of job request and offer advertisements from local newspapers, which were subsequently translated into Classic syntax. The sets were fed to the system, which provided a ranking of them. The same sets of advertisements were proposed to two volunteers who carried out the same ranking of proposals according to their judgment. Without any claim of completeness, the experiments showed that the system response is quite close to the users' ones, and considering average volunteers

```
demand LOOKING FOR AN ENGINEER, LIVING IN EUROPE, REQUIRED AS OOP
PROGRAMMER TO WORK IN EUROPE. (createIndividual demand (and
Engineer (at-least 1 livingIn)(all livingIn Europe)(at-least 1
requiredAs) (all requiredAs Programmer)(at-least 1
specializedIn)(all specializedIn OOP)))

supply1 COMPUTER SCIENCE ENGINEER, LIVING IN ITALY, REQUIRED AS
PROGRAMMER SPECIALIZED IN DB MANAGEMENT (createIndividual supply1
(and  CSEngineer (at-least 1 livingIn)(all livingIn
Italy)(at-least 1 requiredAs) (all requiredAs Programmer)(at-least
1 specializedIn)(all specializedIn DBmanagement)))

supply2 GRADUATE WITH A DEGREE IN ENGINEERING, LIVING IN ITALY,
WORKING IN EUROPE AS OOP PROGRAMMER (createIndividual supply2 (and
Graduate (all hasDegree Engineering)(at-least 1 workingIn) (all
workingIn Italy)(at-least 1 livingIn)(all livingIn Italy)(at-least
1 requiredAs) (all requiredAs Programmer)(at-least 1
specializedIn)(all specializedIn OOP)))

supply3 GRADUATE WITH A DEGREE IN LAW, WORKING IN ITALY AS
COMPUTER PROGRAMMER (createIndividual supply3 (and  Graduate (all
hasDegree Law)(all workingIn Italy)(at-least 1 requiredAs) (all
requiredAs Programmer)(at-least 1 specializedIn)(all specializedIn
ComputerProgramming)))

supply4 ENGINEER LIVING AND WORKING IN JAPAN AS COMPUTER
PROGRAMMER (createIndividual supply4 (and  Engineer (at-least 1
workingIn)(all workingIn Japan)(at-least 1 livingIn) (all livingIn
Japan)(at-least 1 requiredAs)(all requiredAs Programmer)(at-least
1 specializedIn) (all specializedIn ComputerProgramming)))

supply5 DOCTOR WORKING AND LIVING IN GERMANY (createIndividual
supply5 (and  Doctor (all workingIn Germany)(all livingIn
Germany)))

supply6 MOLECULAR BIOLOGIST LIVING IN GERMANY (createIndividual
supply6 (and  MolecularBiologist (all livingIn Germany)))
```

Figure 5: *Sample demand and supplies together with their Classic description*

orderings the systems rankings is in agreement with human judgment almost always.

6 Conclusions

We have proposed a formal approach to Ontology-Based Semantic Matchmaking between skills demand and supply, modelled as a virtual marketplace of knowledge.

We have set up a formalization of some issues of skill matching and proposed properties that should hold in a semantic-based skill matching approach. Then we have proposed algorithms to rank matches between skills profile descriptions. Finally we have presented our ontology based system, which embeds a modified NeoCLassic reasoner, implementing the ranking algorithms.

Benefits of our logic based approach are the possibility to fully exploit an ontological structure for skill management; the possibility to semantically distinguish between potential, *i.e.*, no contradiction matches, and partial matches between skill profiles; a rational ranking of queried descriptions, which is close to human judgement.

The kind of hypothetical reasoning used in evaluating potential and partial matches gave us the basic motivation to study new non-standard reasoning services in Description Logics, namely concept abduction and contraction and relative computational complexity. A theoretical framework has been devised and results are forthcoming.

We are currently working on a more complete ontology and on further test with human users. Also we are extending our framework to cope with skills matching for the constitution of working team and optimization of human resources scheduling, adapting algorithms based on bipartite graphs and algorithms for optimal resource allocation.

Acknowledgments

This work has been supported by MURST project CLUSTER22, by POR project ‘*‘Tecnologie innovative per la valorizzazione e la fruizione dei Beni Culturali”*, by Italian CNR projects LAICO, and *“Metodi di Ragionamento Automatico nella modellazione ed analisi di dominio”*.

References

[Borgida 1995] Borgida, A.(1995): “Description Logics in Data Management”; IEEE Transactions on Knowledge and Data Engineering, 7(5): 671-682.

[Borgida and Patel-Schneider 1994] Borgida, A. and Patel-Schneider, P. F.(1994): “A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic”; Journal of Artificial Intelligence Research, 1: 277-308.

[Calvanese 1996] Calvanese, D. (1996): “Reasoning with inclusion axioms in description logics: Algorithms and complexity”; In Proceedings of the Twelfth European Conference on Artificial Intelligence (ECAI’96), pages 303-307. John Wiley & Sons.

[Calvanese et al. 1998] Calvanese, D., De Giacomo, G., and Lenzerini, M. (1998): “On the Decidability of Query Containment under Constraints”; In Proceedings of the Seventeenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS’98), pages 149-158.

[de Vasconcelos et al. 2003] de Vasconcelos, J., Kimble, C., and Rocha, A. (2003): “Ontology and the Dynamics of Organizational Environments: An Example of a Group Memory System for the Management of Group Competencies”; In Proc. I-KNOW’03, pages 161-167. J.UCS.

[Di Noia et al. 2003a] Di Noia, T., Di Sciascio, E., Donini, F., and Mongiello, M. (2003): “Abductive matchmaking using description logics”; In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI 2003),pages 337-342, Acapulco, Messico. Morgan Kaufmann, Los Altos.

[Di Noia et al. 2003b] Di Noia, T., Di Sciascio, E., Donini, F., and Mongiello, M.(2003): “Semantic matchmaking in a P-2-P electronic marketplace”; In SAC-03, pages 582-586, Melbourne, Florida, USA. ACM, New York. Special track on E-Commerce technologies.

[Dieng et al. 1999] Dieng, R., Corby, O., Giboin, A., and Ribiere, M. (1999): “Methods and tools for corporate knowledge management”; Intl. J. of Human-Computer Studies, 51(3): 567-598.

[Donini et al. 1996] Donini, F. M., Lenzerini, M., Nardi, D., and Schaerf, A. (1996): “Reasoning in Description Logics”; In Brewka, G., editor, Principles of Knowledge Representation, Studies in Logic, Language and Information, pages 193-238. CSLI Publications.

[Fensel et al. 2001] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D., and Patel-Schneider, P. F. (2001): “OIL: An Ontology Infrastructure for the Semantic Web”. IEEE Intelligent Systems, 16(2): 38-45.

[Guarino 1998] Guarino, N. (1998): “Formal ontology and information systems”; In Proceedings of FOIS ’98 Formal Ontologies in Information Systems, pages 3-15, Trento. IOS Press.

[Jones et al. 1998] Jones, D., Bench-Capon, T., and Visser, P. (1998): “Methodologies for ontology development”; In J. Cuena, editor, Proc. ITi and KNOWS Conference of the 15th IFIP World Computer Congress, pages 62-75, London, UK. Chapman and Hall Ltd.

[Nebel 1990] Nebel, B. (1990): “Terminological Reasoning is Inherently Intractable”; Artificial Intelligence, 43: 235-249.

[OLeary 1998] O’Leary, D. (1998): “Using AI in knowledge management: Knowledge bases and ontologies”; IEEE Intelligent Systems, 13(3): 34-39.

[Staab et al. 2001] Staab, S., Schnurr, H., Studer, R., and Sure, Y. (2001): “Knowledge Processes and Ontologies”; IEEE Intelligent Systems, 16(1).

[Stefanutti and Albert 2003] Stefanutti, L. and Albert, D. (2003): “Skill Assessment in Problem Solving and Task Simulation”; In Proc. I-KNOW ’03, pages 174-180. J.UCS.

[Sure et al. 2000] Sure, Y., Maedche, A., and Staab, S. (2000): “Leveraging corporate skill knowledge - from proper to ontoproper”; In Proc. of the Third Intl. Conf. on Practical Aspects of Knowledge Management.

[Trastour et al. 2002] Trastour, D., Bartolini, C., and Priest, C. (2002): “Semantic Web Support for the Business-to-Business E-Commerce Lifecycle”; In Proc. WWW ’02, pages 89-98. ACM.

[Uschold and Gruninger 1996] Uschold, M. and Gruninger, M. (1996): “Ontologies: Principles, methods and applications”; Knowledge Engineering Review, 11(2): 93-113.