Proceedings of I-KNOW '03
Graz, Austria, July 2-4, 2003

Finding Skills through Ranked Semantic Match of Descriptions

Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Marina Mongiello
(Politecnico di Bari, Bari, Italy,
s.colucci, t.dinoia, disciascio, mongiello@poliba.it)

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

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

Abstract: We propose a formal approach to Ontology-Based Semantic Matchmaking between Skills request and offer, devised as a virtual marketplace of knowledge. In such a knowledge market metaphor, 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; 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 for match ranking and categorization. The implementation of the approach in a prototype facilitator, which embeds a NeoCLassic reasoner, is also described.

Key Words: Description Logics, Skill Management, Matchmaking
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. Information systems of Knowledge-intensive organizations strive to provide access to such resources. A skill management system 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 [8, 6, 7, 3].

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 [9, 8, 7], they do not take full advantage of the ontological structure and limit their search to simple subsumption matching. Using Description Logics (DL) [4], described in more detail in Section 2, we logically distinguish matches into categories and rank matches within a category. Categories are: *potential match i.e.*, 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, and *partial match*, *i.e.*, some skills in the request are in contrast with the given profile. 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. Notice that *total match* is just a special case of *potential* match. This coarse subdivision allows to immediately classify a given request/offer description in a category. The ranking within each category is also of extreme importance for a practical use of the approach, as the key question that has to be answered is how far is a given skills request(offer) from its counterpart? And which are the requirements that would eventually fulfill it?

In the next section we briefly present DL basics to ease paper reading, then we show how Knowledge Representation can help in the formalization of some common sense properties an automated skill matching should have. Then we present our facilitator and point out main features of it. Last section draws the conclusions.

2 Description Logics basics

DLs [1, 4] 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. 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.*, $\texttt{Graduate} \sqcap \exists \texttt{hasDegree.Engineering}$, which describes the set of graduates with an engineering degree, and *universal role quantification*, *e.g.*, $\texttt{person} \sqcap \forall \texttt{livingIn.Apulia}$, which describes persons living exclusively in Apulia. Other constructs may involve counting, as number restrictions: $\texttt{Person} \sqcap (\leq 1\ \texttt{hasDegree})$ expresses persons with at most one degree, and $\texttt{Person} \sqcap (\geq 3\ \texttt{hasSpecialization})$ describes persons endowed of at least three specializations. 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 faculties can be divided into scientific and art ones using the two inclusions $\texttt{Faculty} \sqsubseteq \texttt{Scientific} \sqcup \texttt{SocialSciences}$ and $\texttt{Scientific} \sqsubseteq \neg \texttt{SocialSciences}$. Or that graduates has at least one degree as $\texttt{Graduate} \sqsubseteq (\geq 1\ \texttt{hasDegree})$. 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. For instance, the concept $\texttt{AFaculty} \sqsubseteq \texttt{Scientific} \sqcap \texttt{SocialSciences}$ is clearly unsatisfiable w.r.t. the TBox containing the inclusion scientific and social sciences. 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). The CLASSIC system we use 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?).

3 KR formalization of matching principles

We highlight here common sense principles that a semantic based approach should yield. First of all, a matchmaking facilitator of practical use has not to pretend a proposer to fill in forms with (say) 30 or more different characteristics to be set. This implies that the absence of a characteristic in the description of a requested or offered profile should not be interpreted as a constraint of absence. Instead, it should be considered as a characteristic that could be either refined later, or left open if it is irrelevant for a user — what is called *open-world assumption* in KR. Secondly, a matchmaking system may give different evaluations depending on *who* is going to use this evaluation.

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 [5]. Hence, we assume that supplies and demands are expressed in a DL. 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 highlight here three relations between concepts that we consider meaningful in the semantic matchmaking of skills:

Implication. If $T \models (D \sqsubseteq S)$, then every constraint imposed by D is fulfilled (implied) by S, and vice versa if $T \models (S \sqsubseteq D)$. 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*, we state that $T \models (D \sqsubseteq S)$. If both $T \models (D \sqsubseteq S)$ and $T \models (S \sqsubseteq D)$, then D and S should be considered equivalent in T. This relation extends exact matching by ruling out irrelevant syntactic differences.

Consistency. If $D \sqcap S$ is satisfiable in T, then there is a *potential* match, in the sense that the constraints of neither proposal exclude the other. 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 [9]. 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.

Inconsistency. Otherwise, if $D \sqcap S$ is unsatisfiable in T, some constraints of one proposal are in contrast with the properties of the other one. 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. However, also (say) supplies which are inconsistent with D may be reconsidered, if the demander accepts to *revise* some of D's constraints. We call this situation a *near miss* or *partial match*. Also in this case a ranking — different from the one of potential matches — is fundamental.

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. Besides, a ranking for semantic matchmaking should be *monotonic over sub-*

sumption. That is, 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 S_1 and S_2 should be ranked the same. The two properties should hold also for every pair of demands D_1, D_2 with respect to a supply S. When turning to partial matches, 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).

4 A prototype 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 [2]. Our matching engine is based on Java servlets; it embeds the NeoClassic reasoner and communicates with the reasoner running as a background daemon. The system receives a Knowledge Representation System Specification (KRSS) string describing the demand/supply and the URI referencing the skills ontology. 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.

As an example suppose to have the demand *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 } and a set of possible profiles: *computer science engineer, living in Italy, required as programmer specialized in db management*(S_1); *engineer living in Italy, working in Europe as OOP programmer*(S_2); *graduate with a degree in law, working in Italy as computer programmer*(S_3); *engineer living and working in Japan as computer programmer*(S_4); *doctor working and living in Germany*(S_5); *molecular biologist living in Germany*(S_6).

Comparing the requested skills with the available set by simple text analysis 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 $R = \{S_2, S_1, S_3, S_5, S_4 - S_6\}$ in which S_2, S_1, S_3, S_5 potentially match the demand and S_4, S_6 partially.

The facilitator currently provides the following services: support to the user in the data insertion and query submission; automatic construction and verification of consistency w.r.t. the reference ontology of the profile; deduction of new knowledge on the basis of available data; ability to provide ranked conceptually approximate answers, *i.e.*, near miss or partial match, in the presence of unsatisfiable queries; 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; 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.

At the current stage of the project our ontology is still a small one, with approximately seventy concepts. 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. Without any claim of completeness, the experiments show that the system response is quite close to the users' ones, and considering average volunteers orderings the systems rankings is in agreement with the human judgement almost always.

5 Conclusions

We have proposed a formal approach to Ontology-Based Semantic Matchmaking between Skills request and offer. The approach allows to semantically distinguish and rank possible matches. The implementation of the approach in a prototype facilitator, which embeds a modified NeoCLassic reasoner, has been described. The system is fully functional and we are studying how to extend logic formalism in order to achieve better correspondence with human judgement.

References

1. A. Borgida. Description Logics in Data Management. *IEEE TKDE*, 7(5):671–682, 1995.
2. T. Di Noia, E. Di Sciascio, F. Donini, and M. Mongiello. Semantic matchmaking in a P-2-P electronic marketplace. In *Proc. of ACM SAC'03*, pages 582–586, Melbourne, Florida, USA, March 9–12 2003.
3. R. Dieng, O. Corby, A. Giboin, and M. Ribiere. Methods and tools for corporate knowledge management. *Intl. J. of Human-Computer Studies*, 51(3):567–598, 1999.
4. F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in Description Logics. In G. Brewka, editor, *Principles of Knowledge Representation*, Studies in Logic, Language and Information, pages 193–238. CSLI Publications, 1996.
5. D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. F. Patel-Schneider. OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.
6. D. OLeary. Using AI in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems*, 13(3):34–39, 1998.
7. S. Staab, H. Schnurr, R. Studer, and Y. Sure. Knowledge Processes and Ontologies. *IEEE Intelligent Systems*, 16(1), 2001.
8. Y. Sure, A. Maedche, and S. Staab. Leveraging corporate skill knowledge - from proper to ontoproper. In *Proc. of the Third Intl. Conf. on Practical Aspects of Knowledge Management*, 2000.
9. D. Trastour, C. Bartolini, and C. Priest. Semantic Web Support for the Business-to-Business E-Commerce Lifecycle. In *Proc. WWW '02*, pages 89–98. ACM, 2002.