Journal of Universal Computer Science, vol. 10, no. 6 (2004), 723-731
submitted: 26/1/04, accepted: 22/3/04, appeared: 28/6/04 © J.UCS

Semantic-based Approach to Task Assignment of Individual Profiles

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)

Giacomo Piscitelli
(Politecnico di Bari, Bari, Italy,
piscitel@poliba.it)

Abstract: This paper is focused on the problem of skill matching in an organizational context. We endow the classical weighted bipartite graph approach with a semantic based assignment of arcs weight and we describe a skill matching system implementing the approach. The system takes curricula and project specifications as inputs and extracts from them individual profiles respectively offered and requested, according to an ontology modeling skill management context. The suitability of each available individual to each task to assign is evaluated based on an algorithm whose returned scores are used as arc weights. As a result the semantics of profile descriptions is taken into account in the assignment process.

Key Words: skill matching, knowledge elicitation, Description Logics

Category: H.3.3, I.2.4, K.6.1

1 Introduction

Matching personal profiles is an activity required in an arising number of scenarios, ranging from recruitment agencies and human resource organizational units to dating services. Those contexts, even in their deep dissimilarity, share the need to satisfy a list of individual requests, by matching them with available individual profile offers. The match between individual profiles we are interested in is, obviously, not an exact one, which is both quite simple and rare. Given a task description and individual profile descriptions, the matchmaking process has to return one or more best possible matches among the available ones. In this paper we focus on the problem of skill matching in an organizational context, in which management has to assign different tasks to employees. Finding compatible profiles for given tasks is an issue that has been faced under several perspectives. Similarity between weighted vectors of stemmed terms, typical of text-based Information Retrieval, has been used to evaluate possible matches [14], as well as simple bipartite graph matching[12]. Obviously, forcing profiles to be expressed by vectors of terms does not allow to deal with incomplete information, always present in matchmaking context in the form of either unavailable or irrelevant information. Skill matching has been also modeled as a bipartite graph in which the first set of vertices includes assignees and the second one includes tasks to be performed. Edges belonging to this graph link people to task. By determining a cost function that associates each edge with a real value, a weighted bipartite graph ensues, which results in a well known problem in Operational Research area, the Assignment Problem [9, 6, 8]. Among previous work on the subject, in [13] two skill matching systems, *ProPer* and *OntoProper*, were presented, both storing in a database skill profiles represented as vectors and using approaches from decision theory to allow for approximate match, not obtainable with plain database queries. *OntoProper* embeds also an ontology, reducing skill database maintenance effort by enriching the database with ground and inferred facts from secondary information, such as project documents. Both systems lack of an ontology as skill repository, allowing to infer on previously introduced profiles. Other proposals provide support to the search for the right expert: in [11] an ontology based skill management system is proposed, allowing employees to elicit their skills and providing an advanced expert search within the intranet; in [7] an XML multi-agent system is proposed to support management in searching the most suitable employee for a specific job.

Here we present an approach to endow with semantics the process of searching solutions to task assignment. In particular we endow weighted bipartite graph approach with a semantic based assignement of arcs weight, employing a methodology that takes into account the semantics of profile descriptions to be matched. The approach is implemented in a skill matching system, that takes curricula and project specifications as inputs and extracts from them individual profiles respectively provided and required, according to a common ontology. The system then matches such descriptions to evaluate the suitability between each couple individual-task before the assignment. The remaining of this paper is organized as follows: in the next section we formally describe the approach, an example of which is presented in section 3. Then in section 4 we briefly describe the system implementing our approach and finally draw the conclusions.

2 The Assignment Process

The Assignment Problem is a linear programming problem whose objective is to assign a number of assignees to a number of tasks to be performed. The problem classical application is to assign jobs to employees minimizing an objective function measuring the total cost of assignment. We may think of the cost function in term of suitability of persons to tasks. This assumption causes the objective function to measure quantitatively the effectiveness of performing all the tasks instead of the total cost of the assignment.

Evaluating the suitability of an individual to a job is a task traditionally performed by companies management on the basis of personal knowledge of workers. As a result, knowledge about coefficients measuring suitability of different matches is subjective and implicit, not allowing end users to clearly determine the reasons for match suggestions and to eventually revise them.

In our approach, profile descriptions are modeled using a Description Logic ([5, 1]) and share a common ontology, modeling skill management context. Description, or Terminological, Logics (DLs) are a family of logic formalisms for Knowledge Representation. All DLs are endowed of a syntax, and a semantics, which is usually model-theoretic. The basic syntax elements of DLs are: *concept* names, *role* names, *individuals*. Intuitively, concepts stand for sets of objects, and roles link objects in different concepts. Individuals are used for special named elements belonging to concepts.

More formally, a semantic *interpretation* is a pair $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, which consists of the *domain* Δ and the *interpretation function* $\cdot^{\mathcal{I}}$, which maps every concept to a subset of Δ, every role to a subset of $\Delta \times \Delta$, and every individual to an element of Δ. We assume that different individuals are mapped to different elements of Δ, *i.e.*, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for individuals $a \neq b$. This restriction is usually called *Unique Name Assumption* (UNA). 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* and *universal role quantification*. Other constructs may involve counting, as *number restrictions*. Many other constructs can be defined, increasing the expressive power of the DL, up to n-ary relations [3]. Expressions are given a semantics by defining the interpretation function over each construct. The interpretation of constructs involving quantification on roles needs to make domain elements explicit. 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. Definitions are useful to give a meaningful name to particular combinations. Sets of such inclusions are called TBox (Terminological Box). The semantics of inclusions and definitions is based on set containment: an interpretation $\mathcal{I}$ satisfies an inclusion $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and it satisfies a definition $C = D$ when $C^{\mathcal{I}} = D^{\mathcal{I}}$. A *model* of a TBox $\mathcal{T}$ is an interpretation satisfying all inclusions and definitions of $\mathcal{T}$.

DL-based systems usually provide two basic reasoning services: *Concept Satisfiability*: given a TBox $\mathcal{T}$ and a concept C, does there exist at least one model of T assigning a non-empty extension to C? *Subsumption*: given a TBox $\mathcal{T}$ and two concepts C and D, is C more general than D in any model of $\mathcal{T}$?

Let T be a TBox, and J and I concepts respectively representing the task and the profile description. Our approach first checks that $J \sqcap I$ is satisfiable(otherwise the job description and the personal profile are assumed incompatible). Then it gives a measure of "‘how far”’ is I from a concept I' such that I' is subsumed by J (that is, all requirements of J are fulfilled by I', plus possibly other capabilities).

The algorithm at the basis of our approach takes as input two descriptions J and I that potentially match and returns a *score* $n \geq 0$ of I w.r.t. J using the algorithm proposed in [4]. The score measures the number of characteristics needed for the task but absent in the assignee profile, with $n = 0$ when I is subsumed by J.

The skill matching system uses computed scores as measure of suitability between assignees and jobs.

Given n persons and m jobs, the assignment process starts by applying $n * m$ times the algorithm to match each concept I_i with each concept J_j. The algorithm returns a score r_{ij} of I_i w.r.t. J_j, if I_i and J_j potentially match; otherwise the algorithm does not return any score. Scores returned by the algorithm are then used to model the assignment problem as a bipartite graph matching with cost function minimization:

Minimize

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} r_{ij} x_{ij}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \quad for\ j = 1, 2, ...n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad for\ i = 1, 2, ...n$$

and

$$x_{ij} \in \{0, 1\} \ \ (all \ i, j)$$

where x_{ij} are the decision variables such that $x_{ij} = 1$ if assignee i performs task j and $x_{ij} = 0$ otherwise. The first set of functional constraints imposes that each person is assigned to exactly one task, whereas the second set forces each task to be performed by exactly one person. Coefficients r_{ij} denote the suitability of individual i to job j and take the place of cost coefficients c_{ij} in the problem general model, for all the concepts I_i, J_j that potentially match. If $I_i \sqcap J_j$ is not a satisfiable concept, we assign to r_{ij} a value M great enough to force the corresponding value of x_{ij} to be zero in the final solution. The assignment is then obtained by adopting the well known Kuhn algorithm [10].

3 An Assignment Example

We here outline the assignment process we propose with the help of a simple example. Suppose the management of a Company has to assign four tasks, for which the following profiles are required:
Profile 1: Consultant, TCP/IP and C++ skilled, required as programmer.
Profile 2: Engineer, belonging to the internal personnel, living in Italy and expert in TCP/IP, C and C++.
Profile 3: Researcher, graduated in Computer Science, TCP/IP skilled.
Profile 4: Engineer, living in Italy, working as Project Manager.
The four individuals described in the following are available to be assigned:
Simona: Engineer, Project Manager, Programmer, expert in C++.
Gianvito: Engineer, Consultant, Programmer, expert in C and C++, living in Italy.
Marina: Researcher graduated in Computer Science, Project Manager, Programmer,living in Italy and belonging to internal personnel, expert in C, C++ and Java.
Tommy: Engineer, working as researcher and belonging to internal personnel, expert in C, C++ and Java.

First we evaluate the suitability on each pair individual-task. Resulting scores are shown in Figure 1. Notice that scores measure the number of characteristics required for performing the task, but absent in the individual description. So, a high score means low suitability and vice versa. Following the approach previously outlined we solve the problem in Figure 2. In Figure 3 we show the problem representation as a bipartite graph. By applying the Kuhn solving algorithm[10] to this assignment problem we obtain one possible solution:

$$x_{14} = 1 \ \ x_{21} = 1 \ \ x_{33} = 1 \ \ x_{42} = 1$$

	Profile 1	Profile 2	Profile 3	Profile 4
Simona	2	4	3	1
Gianvito	1	2	3	1
Marina	2	1	1	1
Tommy	3	1	2	2

Figure 1: Suitability matrix

$$
\begin{aligned}
Z = \; & 2x_{11} + \; 4x_{12} + \; 3x_{13} + \; x_{14} + \\
& x_{21} + \; 2x_{22} + \; 3x_{23} + \; x_{24} + \\
& 2x_{31} + \; x_{32} + \; x_{33} + \; x_{34} + \\
& 3x_{41} + \; x_{42} + \; 2x_{43} + \; 2x_{44}
\end{aligned}
$$

$$\sum_{j=1}^{4} x_{ij} = 1 \quad for\ j = 1, 2, ...4$$

$$\sum_{i=1}^{4} x_{ij} = 1 \quad for\ i = 1, 2, ...4$$

$$x_{ij} \in \{0, 1\} \;\; (all\ i, j)$$

Figure 2: Example Problem Formalization

with objective value: $Z = 4$. The assignment corresponding to this solution is shown in Figure 4.

4 The Skill Matching System

The assignment process so far outlined has been implemented in a skill matching system, whose architecture is proposed in figure 5. The system is made up of three main components, all implemented in Java. The first one analyzes text files and extracts individual profiles by employing a methodology enriching classical information retrieval techniques with semantics. Text files contain either curricula vitae or project specification, both referring to a vocabulary typical of skill matching context. This context has been then modeled in an ontology, to which our system refers for the extraction of terms to be included in the profiles. The second component of our system is made up of a communication interface component that sends extracted profiles to a matchmaker service (MAMAS), which embeds a modified NeoClassic reasoner. Profiles are formatted according to the DIG [2] syntax, in order to be managed by the service. The service returns for each received pair of profiles a score. The last component implements the Khun algorithm and performs the assignment on the basis of scores returned by the previous component.

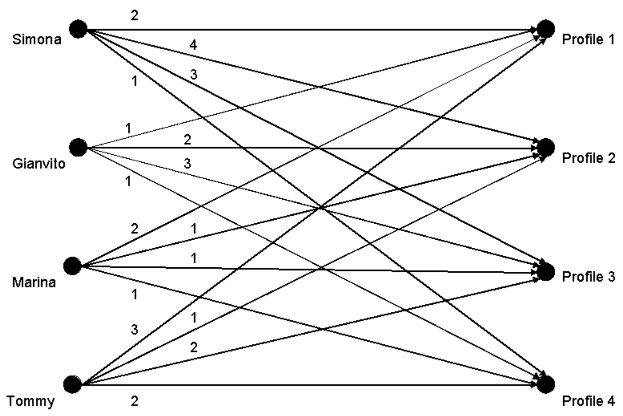

Figure 3: Corresponding bipartite graph

Simona	Profile 4
Gianvito	Profile 1
Marina	Profile 3
Tommy	Profile 2

Figure 4: Final Assignment

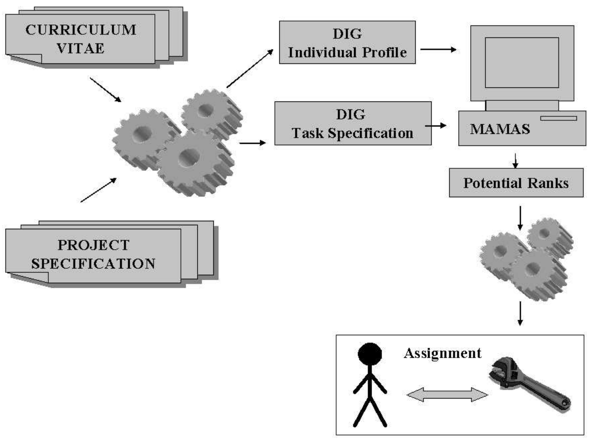

Figure 5: Skill Matching System Architecture

5 Conclusions

In this paper a system and an approach have been proposed, providing a mechanism to make explicit and objective a process traditionally implicit and subjective as suitability assessment. The devised framework allows to easily update profile and job descriptions, and supports the automatic recomputation of cost matrix after each update. This feature, typical of any organizational context, becomes fundamental for the suitability of the system in recruitment agencies, in which both sets of required and provided profiles are continuously updated.

References

1. F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook.* Cambridge University Press, 2002.
2. S. Bechhofer, R. Mller, and P. Crowther. The DIG Description Logic Interface. In *Proceedings of the 16th International Workshop on Description Logics (DL'03)*, volume 81 of *CEUR Workshop Proc.*, September 2003.
3. D. Calvanese, G. De Giacomo, and M. Lenzerini. 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, 1998.
4. S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, M. Mongiello, and M. Mottola. A formal approach to ontology-based semantic match of skills descriptions. *Journal of Universal Computer Science (J.UCS)*, 9(12):1437–1454, Dec. 2003.
5. F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in Description Logics. In Gerhard Brewka, editor, *Principles of Knowledge Representation*, Studies in Logic, Language and Information, pages 193–238. CSLI Publications, 1996.
6. Zvi Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Comput. Surv.*, 18(1):23–38, 1986.
7. A. Garro and L. Palopoli. An xml multi-agent system for e-learning and skill management. In *Agent Technologies, Infrastructures, Tools, and Applications for E-Services*, volume 2592 of *LCNS*, pages 283–294. Springer, 2003.
8. F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research.* McGraw-Hill, New York, 1995.
9. J. Kennington and Z. Wang. An empirical analysis of the dense assignment problem: Sequential and parallel implementations. *ORSA Journal on Computing*, 3(4):299–306, 1991.
10. H. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.
11. T. Lau and Y. Sure. Introducing ontology-based skills management at a large insurance company. In *Proceedings of the Modellierung*, 2002.
12. H. Baier Saip and C. Lucchesi. Matching algorithms for bipartite graph. Technical Report DCC -03/93. Departamento de Cincia da Computao, Universidade Estudal de Campinas, 1993.
13. 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.
14. D. Veit, J.P. Muller, M. Schneider, and B. Fiehn. Matchmaking for Autonomous Agents in Electronic Marketplaces. In *Proc. AGENTS '01*, pages 65–66. ACM, 2001.