Knowledge Based Approach to Semantic Composition of Teams in an Organization

S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini*, G.Piscitelli, S. Coppi
Politecnico di Bari Via Re David, 200 I-70125, Bari, Italy
*Università della Tuscia via San Carlo, 32 I-01100, Viterbo, Italy
{s.colucci,t.dinoia,disciascio,piscitel, s.coppi}@poliba.it;donini@unitus.it

ABSTRACT

Finding rapidly suitable experts in an organization to compose a team able to solve specific tasks is a typical problem in large consulting firms. In this paper we present a Description Logics approach to the semantic-based composition of ad-hoc teams based on individuals skill profiles and on task description. The selection process is carried out using a novel Concept Covering algorithm that exploits the recently proposed Concept Abduction inference service in Description Logics. The approach has been deployed as part of a skill management system that takes text files with curricula and project specifications as inputs and extracts from them available individual profiles and task descriptions, according to an ontology modeling skills.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Decision Support; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages

General Terms

Algorithms, Languages, Economics

Keywords

Concept Covering, Description Logics, Concept Abduction

1. INTRODUCTION

Systems and techniques for Skills management have recently become the object of a growing interest, as knowledge and expertise of individuals have been acknowledged strategic assets of knowledge-intensive companies [25, 18]. Information systems developed for knowledge management have been integrated with skill management systems, whose impact on return on investment has been investigated in [16]. The use of ontologies as knowledge repositories has now become almost common in novel knowledge management architectures, in order to give a common vocabulary and to use inference services on elicited knowledge[25, 26]. Once ontologies have been built, the arising issue is "how should we use them?". There is the need for reasoners and reasoning services able to take full advantage of the effort placed in structuring an ontology. Skill management can be characterized in terms of multiplicity relationships between individuals skills and tasks to be accomplished [8]. *One to one*: *one* task or job profile has to be matched with *one* individual; *one to many*: *one* task has to be assigned to *several* individuals that, together, are endowed of all skills requested for task realization; *many to one*: several tasks have to be matched with the skills of an individual able to accomplish them; *many to many*: several tasks have to be assigned to several available individuals. In this paper we propose a Description Logic [1] approach to the semantic-based composition of ad-hoc teams in an organizational context, *i.e.*, we concentrate on *one to many* relationship between a task and the team of individuals that has to be composed to accomplish it. In particular we exploit the recently devised Concept Abduction inference service and use it to solve a Concept Covering problem, which can be seen as the analogous set covering problem adapted to a Description Logics framework. Our approach takes full advantage of structured, ontology-based, descriptions. It adopts an open world assumption, typical of knowledge representation. In practice, the absence of a characteristic in a description is not interpreted as a constraint of absence; instead, it is considered as a characteristic that could be either refined later, or left open if it is irrelevant. It obviously allows to find a set of individuals that, based on provided skills descriptions, cover the requested task, but also, when a completely satisfactory team cannot be composed due to lack of requested skills, provides a logic-based answer to what is missing. The remaining of the paper is structured as follows: next section revises related work on skill management; section 3 presents Description Logics basics, in order to make the paper self-contained, then we present our approach based on Concept Abduction and provide an algorithm for team covering. Section 5 describes the rationale of the proposed algorithm with the aid of a simple case study, while section 6 briefly describes the system the proposed approach is deployed in. Conclusions close the paper.

2. RELATED WORK

Since the first investigations in the knowledge management area, the ability of employing internal competencies has been recognized as one of the key factors for the success of a company. In [18] the focus of companies on core competencies in developing their business is pointed out as the main source of competitive advantage. Such an emphasis to competencies has led to the study of Skill Management as a field of knowledge management. In [16] the role of skill management systems in organizational activities, such as expert finding, personnel recruitment, personnel development and project management is underlined. All these activities need the definition of the skills workers are endowed with. Such an activity traditionally involves human judgment in individuating and classifying skills hold by workers, in evaluating the degree of competence and in keeping up-to-date workers profiles. IT-supported system use is then suggested in managing companies competencies in order to downsize the subjectivity of human evaluation. The problem of expert finding is also modeled in [33], in which the requirements of a skill finding approach are formally outlined in a domain analysis. A server architecture for expert finding, based on the previously mentioned domain analysis is proposed there. Skill management systems presented in literature, almost all embedding skill searching facilities, may be classified in two categories including respectively non ontology-based and ontology-based systems. Among non ontology-based approaches, database querying and similarity between weighted vectors of terms, typical of text-based Information Retrieval, have been used to evaluate possible matches [32]. Obviously, forcing profiles to be expressed by data structures or 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 vertexes includes assignees and the second one includes tasks to be performed [28]. 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 [21, 14, 20]. Among proposal on the subject, in [30] 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. Nevertheless both systems lack of an ontology as skill repository, allowing to infer on previously introduced profiles. In [3] two People Finder Knowledge Management Systems were proposed: the Searchable Answer Generating Environment(SAGE) and the Expert Seeker. Both systems use databases as skill repositories, even though the second one provides more search options. Although proposing a database approach, the paper underlines the need to employ artificial intelligence technologies in People Finder Knowledge Management Systems in order to infer new knowledge from elicited skills and to keep automatically up-to-date profiles employing data mining techniques. Also agent technologies have been employed to support the search for the right expert: in [15] an XML multi-agent system is proposed, providing, among several features, support to management in searching the most suitable employee for a specific job. In [29] an agent-based application for supporting job matchmaking is proposed, focusing on the telework scenario. In [22] an ontology based skill management system is proposed, allowing employees to elicit their skills and providing an advanced expert search within a company intranet. In [19] a semantic based portal is presented for the composition of organizational teams. The user request is formalized as a query, searching the competences required for the task in an ontology used as skills repository. The system returns a set of one or more workers able to cover all the competences required for the task. All the available sets are ranked on the basis of the ontological closeness of query concepts to concepts formalizing skills hold by proposed individuals. In [23] a system integrating the accuracy of concept search with the flexibility of keyword search is proposed to match expertise within academia. The system is based on the use of semantic web technologies and in particular on RDF and XML in order to extract expertise integrated profiles from heterogeneous information sources. In [8] a semantic based approach to the problem of skills finding in an ontology supported framework. The aim is at 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 and provides not only a logical categorization, but also a ranking of matches within each category. In [9] an approach is proposed to endow with semantics the process of searching solutions to the Assignment Problem [20]. The work proposes an assignment based on the maximization of suitability between individuals and tasks and proposes a formal DL-based approach, implementing the algorithm in [8], to evaluate such a suitability in an objective and explicit way. In [6] individual profile matching has been applied to dating services context. The problem of matching user profiles, when the demander's and supplier's profiles can have missing or conflicting information was addressed. A DL-based framework for expressing user profiles in this setting and a language suited for dating services were proposed together with an ad-hoc structural algorithm for matching profiles that, given a demander's and a supplier's profile, returns a penalty: the higher the penalty, the less the two profiles are compatible.

3. DESCRIPTION LOGICS

Description Logics (DLs) [1, 13] are a family of logic formalisms for knowledge representation.

The basic elements of DLs syntax 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*. Based on the set of constructors adopted different DLs can be defined. Every DL allows one to form a *conjunction* of concepts, usually denoted as $\sqcap$; some DLs include also disjunction $\sqcup$ and complement $\neg$ to close concept expressions under boolean operations. Roles can be combined with concepts using *existential role quantification* $\exists.$, *e.g.*, $\texttt{Graduate} \sqcap \exists \texttt{hasDegree.Engineering}$, which describes the set of graduates with an engineering degree, and *universal role quantification* $\forall.$, *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. Many other constructs can be defined, increasing the expressive power of the DL, up to n-ary relations [7]. 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 $\texttt{TeamMember} \sqsubseteq \texttt{InternalPersonnel} \sqcup \texttt{Consultant}$ and $\texttt{InternalPersonnel} \sqsubseteq \neg\texttt{Consultant}$. Or that working teams have at least two members as $\texttt{Team} \sqsubseteq (\geq\ 2\ \texttt{hasTeamMember})$. Historically, sets of such inclusions are called TBox (Terminological Box). DL-based systems usually provide at least two basic reasoning services:
1. *Concept Satisfiability*: given a TBox $\mathcal{T}$ and a concept C, does there exist at least one model of $\mathcal{T}$ assigning a non-empty extension to C? For instance, the concept $\texttt{Member1} \sqsubseteq \texttt{InternalPersonnel} \sqcap \texttt{Consultant}$ is clearly unsatisfiable w.r.t. the TBox containing the inclusion $\texttt{InternalPersonnel} \sqsubseteq \neg\texttt{Consultant}$.
2. *Subsumption*: given a TBox $\mathcal{T}$ and two concepts C and D, is C more general than D in any model of $\mathcal{T}$?

As it is obvious, adding new constructors increases DL languages expressiveness. Nevertheless, it is a well known result [5] that this usually leads to an explosion in computational complexity of inference services. Hence a trade-off is necessary. In this paper we refer to an $\mathcal{ALN}$ (**A**ttributive **L**anguage with unqualified **N**umber restrictions) DL, which can be mapped in a subset of OWL-DL, the recently proposed language for ontologies in the Semantic Web framework [27]. The following constructs are available in an $\mathcal{ALN}$ DL:
$\top$ *universal concept*: all the objects in the domain.
$\bot$ *bottom concept*: the empty set.
A *atomic concepts*: all the objects belonging to the set represented by A.
$\neg A$ *atomic negation*: all the objects not belonging to the set represented by A.
$C \sqcap D$ *intersection*: the objects belonging both to C and D.
$\forall R.C$ *universal restriction*: all the objects participating to the R relation whose range are all the objects belonging to C.
$\exists R.\top$ *unqualified existential restriction*: there exists at least one object participating in the relation R.
$(\geq\ n\ R)|(\leq\ n\ R)|(=\ n\ R)$ *number restrictions*: respectively the minimum, the maximum and the exact number of objects participating in the relation R.

4. TEAM COMPOSITION VIA DESCRIPTION LOGICS

Recently, the Concept Abduction Problem (CAP) has been introduced and defined as a non standard inference problem in DLs [11]. In a formal way a CAP can be defined as follows:

DEFINITION 1. Let C, D, be two concepts in a Description Logic $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms, *i.e.*, an ontology, where both C and D are satisfiable in $\mathcal{T}$. A *Concept Abduction Problem* (CAP), denoted as $\langle \mathcal{L}, C, D, \mathcal{T} \rangle$, is finding a concept H such that $\mathcal{T} \not\models C \sqcap H \equiv \bot$, and $\mathcal{T} \models C \sqcap H \sqsubseteq D$.

$\mathcal{P}$ denotes in the following a CAP, and $SOL(\mathcal{P})$ the set of all solutions to a CAP $\mathcal{P}$.
Given a CAP, if H is a conjunction of concepts and no sub-conjunction of concepts in H is a solution to the CAP, then H is an irreducible solution. Informally, the solution to a CAP can be interpreted as *what have I to hypothesize in C, and in a second step add to, in order to make C more specific than D?* In other words H is *what is expressed, explicitly or implicitly, in D and is not present in C*, or in an equivalent way *which part of D is not covered by C*. In [11] also minimality criteria for H and a polynomial algorithm to find solutions which are irreducible, for $\mathcal{ALN}$ DL, have been proposed. A numerical version of the algorithm has also been defined, *rankPotential*, which computes the length of H solution of the CAP $\langle \mathcal{L}, C, D, \mathcal{T} \rangle$[12].

In [17] the *best covering problem* in Description Logics was introduced as "...a new instance of the problem of rewriting concepts using terminologies". That is, given a concept C and a set of concept definitions in a terminology $\mathcal{T}$, find concepts defined in $\mathcal{T}$such that their conjunction can be an approximation of C. In order to define a concept covering two non standard inferences were then used: the least common subsumer (lcs)[2] and the *difference* or *subtraction* operation [31]. Unfortunately, as the authors admitted, the difference operator makes sense only for a small set of DLs, and surely not for the $\mathcal{ALN}$ (we do not delve into details, for a complete description see [31]). In a more formal way the authors of [17] defined *cover* as follows.

DEFINITION 2. Let $\mathcal{L}$ be a Description Logic with structural subsumption, $\mathcal{T}$ be a terminology using operator allowed by $\mathcal{L}$, $\mathcal{R}$ be the set of concept definitions in $\mathcal{T}$, $\mathcal{R} = \{S_i, i \in [1..n]\}$, and D be a concept in $\mathcal{L}$ such that $\mathcal{T} \not\models D \equiv \bot$. A *cover* of a D using $\mathcal{T}$is finding a set $\mathcal{R}_c \subseteq \mathcal{R}$ such that $\sqcap S_i$, conjunction of all the $S_i \in \mathcal{R}_c$, is such that $D - lcs_{\mathcal{T}}(D, \sqcap S_i) \not\equiv D$.

A cover is then finding a set of concepts defined in $\mathcal{T}$such that they contain the information in D. Notice that a DL with structural subsumption is needed in order to use *concept difference*. In [17] also an hypergraphs based methodology is presented to compute best covers. We now extend the previous definition, in terms of a Concept Covering Problem, both eliminating limitations on $\mathcal{L}$ to be used, and rewriting it in terms of Concept Abduction.

DEFINITION 3. Let D be a concept, $\mathcal{R} = \{S_1, S_2, ..., S_k\}$ be a set of concepts in a Description Logic $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms, where D and $S_i, i = 1..k$ are satisfiable in $\mathcal{T}$.
1. A *Concept Covering Problem* (CCoP), denoted as $\langle \mathcal{L}, \mathcal{R}, D, \mathcal{T} \rangle$, is finding, if it exists, a set $\mathcal{R}_c \subseteq \mathcal{R}$, such that both for each $S_j \in \mathcal{R}_c$, $\mathcal{T} \not\models \sqcap S_j \equiv \bot$, and $H \in SOL(\langle \mathcal{L}, \sqcap S_j, D, \mathcal{T} \rangle)$ is such that $H \not\sqsubseteq D$.
2. We call $\langle \mathcal{R}_c, H \rangle$ a *solution* for the CCoP $\langle \mathcal{L}, \mathcal{R}, D, \mathcal{T} \rangle$.
In the above definition the elements for the solution $\langle \mathcal{R}_c, H \rangle$ of a CCoP represent respectively:
$\mathcal{R}_c$: which concepts in $\mathcal{R}$ represent the cover for D w.r.t. $\mathcal{T}$.
H: what is still in D and is not covered by concepts in $\mathcal{R}$.

We use the symbol $\mathcal{V}$ for CCoP and $SOLCCoP(\mathcal{V})$ for the set of all the solution to a CCoP $\mathcal{V}$. Actually, there are several solution for a single CCoP, depending also on the strategy adopted for searching concepts belonging to $\mathcal{R}_c$.
Based on the definition of Concept Covering Problem we

now define the *best cover* and the *exact cover*.

DEFINITION 4. Given $\mathcal{V}$, a *best cover* for $\mathcal{V}$, w.r.t. an order $\prec$ for H, is a solution $\langle \mathcal{R}_c, H_b \rangle \in SOLCCoP(\mathcal{V})$ such that there is no other $\langle \mathcal{R}'_c, H' \rangle \in SOLCCoP(\mathcal{V})$ with $H' \prec H_b$.

There is no solution $\langle \mathcal{R}'_c, H' \rangle$ for $\mathcal{V}$ such that H', the remaining part of D yet to be covered, is *smaller* than H_b.

DEFINITION 5. Given $\mathcal{V}$, an *exact cover* for $\mathcal{V}$ is a solution $\langle \mathcal{R}_c, H_e \rangle \in SOLCCoP(\mathcal{V})$ such that $H_e \equiv \top$.

Having a set $\mathcal{R}$ of concepts $S_i, i = 1..k$, we want to find a subset $\mathcal{R}_c$ of $\mathcal{R}$, if exists, such that the conjunction of all the concepts in $\mathcal{R}_c$ is more specific than, *i.e.*, is subsumed by, D. In other words, we are looking for a set of concepts which completely cover D.

4.1 Solving a Concept Covering Problem

Based on the GREEDY-SET-COVER presented in [10], we now present a tractable algorithm to compute a solution to a CCoP. In [10] is also proved that, for a set covering problem, the solution grows logarithmically in the size of the set to be covered with respect to the minimal one.

Algorithm $GREEDYsolveCCoP(\mathcal{R}, D, \mathcal{T})$
input concepts D, $S_i \in \mathcal{R}, i = 1..k$, where D and S_i are satisfiable in $\mathcal{T}$
output $\langle \mathcal{R}_c, H \rangle$
begin algorithm
  $\mathcal{R}_c = \emptyset$;
  $D_{uncovered} = D$;
  $H_{min} = D$;
  do
    $S_{min} = \top$;
    /* [♣] *Perform a greedy search among* $S_i \in \mathcal{R}$ */
    for each $S_i \in \mathcal{R}$
      if $\mathcal{R}_c \cup \{S_i\}$ *is a cover* for $D_{uncovered}$ **then**
        $H = solveCAP(\langle \mathcal{L}, S_i, D_{uncovered}, \mathcal{T} \rangle)$;
        /* [♢] *Choose* S_i *based on an order* */
        if $H \prec H_{min}$ **then**
          $S_{min} = S_i$;
          $H_{min} = H$;
        end if
      end if
    end for each
    /* [♠] *If a new* S_i *is found then add* S_i to $\mathcal{R}_c$ *and remove it from* $\mathcal{R}$ */
    if $S_{min} \not\equiv \top$ **then**
      $\mathcal{R} = \mathcal{R} \backslash \{S_i\}$;
      $\mathcal{R}_c = \mathcal{R}_c \cup \{S_i\}$;
      $D_{uncovered} = H_{min}$;
    end if
    /* [♡] *Continue searching until no* S_i *is found* */
  while$(S_{min} \not\equiv \top)$;
  return $\langle \mathcal{R}_c, D_{uncovered} \rangle$;
end algorithm

The algorithm tries to cover D *as much as possible*, using the concepts $S_i \in \mathcal{R}$.
♡ If it is not found any new useful $S_i \in \mathcal{R}$, that is any S_i such that it covers D more, then the algorithm terminates.
♣ A greedy approach is used to choose the *candidates* for $\mathcal{R}_c$.
♢ Among the candidates the one such that H, solution for the local CAP, is minimal w.r.t. an order $\prec$ is chosen.
♠ If the greedy search returns a new S_i, it is removed from $\mathcal{R}$ and added to $\mathcal{R}_c$.

The above algorithm itself is polynomial in time and the complexity source is in the solution of the CAPs and the comparison in [♢]. For the $\mathcal{ALN}$ DL, in [11] a polynomial algorithm (*findIrred*) is proposed to find irreducible solutions for a CAP, and in [12] the tractable *rankPotential* is presented to rank concepts. Using such algorithms, $GREEDYsolveCCoP$ can be solved in polynomial time.

5. ALGORITHM BEHAVIOR

In this section we present the rationale of our approach with the aid of a simple example and with reference to a tiny ontology, reported in figure 1, needed in the example. Let us suppose a Company has the problem to assign the realization of a project to an ad-hoc created working team composed according to the following, simple, specifications: *graduated internal personnel, with experience in programming, DBmanagement, project and human resources management.* Such a request can be formalized in DL as:

$D =$ `Graduate`$\sqcap$
  $\forall$`hasTechnicalSkills.`(`Programming`$\sqcap$`DBmanagement`)
  $\sqcap\forall$`hasManagementSkills.`(`ProjectManagement`
  $\sqcap$`HumanResourcesManagement`)
  $\sqcap\forall$`personnelRole.InternalPersonnel`

D is one of the inputs of $GREEDYsolveCCoP(\mathcal{R}, D, \mathcal{T})$. Suppose now the four individual profiles described in the following are available as team members:
Tom: Computer science engineer belonging to internal personnel.
$Jerry$: Graduated programmer, working as consultant, expert in DBmanagement and SAP.
$Parker$: Internal worker, graduated in Economy, with experience in project management.
$Clark$: Biologist specialized in micro-biology, working as consultant.

Such candidate team members can be formalized as DL concepts to be used as the $S_i \in \mathcal{R}, i = 1..4$, inputs of $GREEDYsolveCCoP(\mathcal{R}, D, \mathcal{T})$:

$S_1(Tom) =$ `CSEngineer` $\sqcap\,\exists$`personnelRole`
  $\sqcap\forall$`personnelRole.InternalPersonnel`

$S_2(Jerry) =$ `Programmer` $\sqcap\,\exists$`hasDegree`
  $\sqcap\forall$`hasTechnicalSkills.`(`DBmanagement` $\sqcap$ `SAP`)
  $\sqcap\exists$`personnelRole` $\sqcap\,\forall$`personnelRole.Consultant`

$S_3(Parker) =$ `Economist` $\sqcap\,\exists$`personnelRole`
  $\sqcap\forall$`personnelRole.InternalPersonnel`

$S_4(Clark) =$ `Biologist` $\sqcap\,\exists$`personnelRole`
  $\sqcap\forall$`personnelRole.Consultant`
  $\sqcap\forall$`hasTechnicalSkills.MicroBiology`

D and S_i are satisfiable in $\mathcal{T}$, the TBox containing skill management domain assertions, shown in Figure 1. Applying $GREEDYsolveCCoP(\mathcal{R}, D, \mathcal{T})$ to the example, the output will be $\langle \mathcal{R}_c, H \rangle$ with
$\mathcal{R}_c = \{Tom, Parker\}$
$H = \forall$`hasTechnicalSkills.DBmanagement`$\sqcap$
$\forall$`hasManagementSkills.HumanResourcesManagement`

The above result has to be interpreted as: "*Using the available candidates, the system proposes a team composed*

$\texttt{Graduate} \equiv \exists\texttt{hasDegree}$
$\texttt{Consultant} \sqsubseteq \neg\texttt{InternalPersonnel}$
$\texttt{ManagementSkills} \sqsubseteq \neg\texttt{TechnicalSkills}$
$\texttt{ProjectManagement} \sqsubseteq \texttt{ManagementSkills}$
$\texttt{HumanResourcesManagement} \sqsubseteq \texttt{ManagementSkills}$
$\texttt{Programming} \sqsubseteq \texttt{TechnicalSkills}$
$\texttt{SAP} \sqsubseteq \texttt{TechnicalSkills}$
$\texttt{DBmanagement} \sqsubseteq \texttt{TechnicalSkills}$
$\texttt{Programming} \sqsubseteq \texttt{TechnicalSkills}$
$\texttt{CSEngineer} \equiv \texttt{Graduate} \sqcap \forall\texttt{hasDegree.Engineering} \sqcap \forall\texttt{hasTechnicalSkills.Programming}$
$\texttt{Programmer} \sqsubseteq \forall\texttt{hasTechnicalSkills.Programming}$
$\texttt{Economist} \equiv \texttt{Graduate} \sqcap \forall\texttt{hasDegree.Economy} \sqcap \forall\texttt{hasManagementSkills.ProjectManagement}$

Figure 1: The toy ontology used as reference in the example

by Tom *and* Parker *to cover the proposed task. There is no one among the remaining candidates, due to the previous missing skills, who is compatible with the proposed group. No one in the proposed team* has human resources management skills *and* is able to manage a Database." Based on the previous result, if the user revises her request and retracts on the `personnelRole` requirement, the request can be reformulated as follows:

$$D = \texttt{Graduate} \sqcap \forall\texttt{hasTechnicalSkills.}(\texttt{Programming} \sqcap \texttt{DBmanagement}) \sqcap \forall\texttt{hasManagementSkills.}(\texttt{ProjectManagement} \sqcap \texttt{HumanResourcesManagement})$$

the new composition result is:
$\mathcal{R}_c = \{Tom, Parker, Jerry\}$
$H = \forall\texttt{hasManagementSkills.HumanResourcesManagement}$.
That is:" *Using the available candidates, the system proposes a team composed by* Tom*,* Parker *and* Jerry *to cover the proposed task. There is no one among the remaining candidates, owning the previous missing skills, who is compatible with the proposed group. No one in the proposed team* has human resources management skills."

6. A SEMANTIC-BASED SKILL MANAGEMENT SYSTEM

The semantic-based team composing process so far outlined has been deployed as part of a larger skill matching system, whose architecture is proposed in figure 2. The system is made up of two 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, our system refers to for the extraction of terms to be included in the profiles. This component is used to extract both the skills request from the specification of the project to perform and the profiles of individuals considered available in the team composition from their curricula vitae. The second component of our system is a team composition interface implementing the algorithm $GREEDY solveCCoP$. Such interface sends extracted profiles and the skills request to a matchmaker service (MAMAS), which embeds a modified

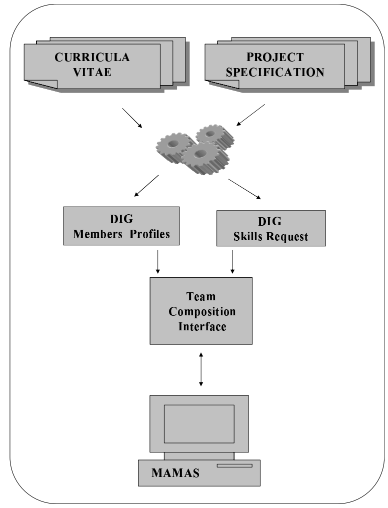

Figure 2: Skill Matching System Architecture

version of the NeoClassic reasoner. MAMAS is an HTTP service, available on-line, which extends the DIG specification [4] adding new TAGs both for Concept Abduction and its numerical version [24, 11].

7. CONCLUSIONS

In this paper we proposed a Description Logics framework for to the semantic-based composition of teams based on individuals skill profiles and on tasks description. To this aim we devised a novel Concept Covering algorithm exploiting the Concept Abduction inference service in DLs. The framework is currently embedded as part of a complete logic-based skill management system.

The approach is tailored for consulting companies and makes objective and explicit the process of searching team members, an organizational task traditionally performed on the basis of implicit and subjective criteria. Although a thorough and comprehensive evaluation of the system and of its benefits in an organizational context are still in progress, initial results appear encouraging and confirm the validity of the approach.

We are currently investigating contemporary composition of many ad-hoc teams, in order to avoid non-optimal human resources employment.

Acknowledgements

The authors acknowledge partial support of projects MS3DI, CNOSSO and PRIN *Metodi statistici per l'integrazione di dati da fonti diverse.*

8. 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] Franz Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In *Proc. Intl. Joint Conf. on Artificial Intelligence (IJCAI2003)*, pages 319–324, 2003.

[3] Irma Becerra-Fernandez. The role of artificial intelligence technologies in the implementation of people-finder knowledge management systems. *Knowl.-Based Syst.*, 13(5):315–320, 2000.

[4] S. Bechhofer, R. Mller, and P. Crowther. The DIG Description Logic Interface. In *Proc. of the 16th Intl. Workshop on Description Logics (DL'03)*, volume 81 of *CEUR Workshop Proceedings*, September 2003.

[5] R.J. Brachman and H.J. Levesque. The tractability of subsumption in frame-based description languages. In *Proc. of the Fourth National Conference on Artificial Intelligence (AAAI-84)*, pages 34–37. Morgan Kaufmann, Los Altos, 1984.

[6] A. Calí, D. Calvanese, S. Colucci, T. Di Noia, and F.M. Donini. A description logic based approach for matching user profiles. In *Proc. of the 17th Intl. Workshop on Description Logics (DL'04)*, volume 104. CEUR Workshop Proceedings, 2004.

[7] 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.

[8] 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.

[9] S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, M. Mongiello, and G. Piscitelli. Semantic-based approach to task assignment of individual profiles. *Journal of Universal Computer Science*, 10(6):723–730, 2004.

[10] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The Massachusetts Institute of Technology, 1990.

[11] T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. Abductive matchmaking using description logics. In *Proc. of the 18th Intl. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 337–342. Morgan Kaufmann, Los Altos, 2003.

[12] T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. A system for principled Matchmaking in an electronic marketplace. In *Proc. International World Wide Web Conference (WWW '03)*, pages 321–330. ACM, New York, 2003.

[13] 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.

[14] Zvi Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Comput. Surv.*, 18(1):23–38, 1986.

[15] 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.

[16] N. Gronau and M. Uslar. Integrating knowledge management and human resources via skill management. In *Proc. I-KNOW '04*, pages 135–142. J.UCS, 2004.

[17] Mohand-Sad Hacid, Alain Leger, Christophe Rey, and Farouk Toumani. Computing Concept Covers: a Preliminary Report. In *Proc. of the 15th Intl. Workshop on Description Logics (DL'02)*, volume 53 of *CEUR Workshop Proceedings*, 2002.

[18] G. Hamel and C. K. Prahalad. The core competence of the corporation. *Harvard Business Review*, pages 79–91, 1990, May-June.

[19] M. Hefke and L.Stojanovic. An ontology-based approach for competence bundling and composition of ad-hoc teams in an organization. In *Proc. I-KNOW '04*, pages 126–134. J.UCS, 2004.

[20] F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, New York, 1995.

[21] 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.

[22] T. Lau and Y. Sure. Introducing ontology-based skills management at a large insurance company. In *Proc. of the Modellierung*, 2002.

[23] P. Liu and P. Dew. Using semantic web technologies to improve expertise matching within academia. In *Proc. I-KNOW '04*, pages 370–378. J.UCS, 2004.

[24] T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. A system for principled Matchmaking in an electronic marketplace. *Intl. Journal of Electronic Commerce*, 8(4):9–37, 2004.

[25] O'Leary. Enterprise knowledge management. *IEEE Computer*, 31(3):54–61, 1998.

[26] D. OLeary. Using AI in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems*, 13(3):34–39, 1998.

[27] OWL. www.w3.org/TR/owl-features/.

[28] 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.

[29] K. Sugawara. Agent-based application for supporting job matchmaking for teleworkers. In Dilip Patel, Shushma Patel, and Yingxu Wang, editors, *Proc. of the 2nd IEEE Intl. Conference on Cognitive Informatics (ICCI 2003), 18-20 August 2003, London, UKIEEE ICCI*, pages 137–142. IEEE Computer Society, 2003.

[30] 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.

[31] G. Teege. Making the Difference: A Subtraction Operation for Description Logics.

[32] D. Veit, J.P. Muller, M. Schneider, and B. Fiehn. Matchmaking for Autonomous Agents in Electronic Marketplaces. In *Proc. Intl. Conference on Autonomous Agents '01*, pages 65–66. ACM, 2001.

[33] D. Yimam. Expert finding systems for organizations: Domain analysis and the demoir approach. In *Beyond Knowledge Management: Sharing Expertise*. MIT Press, 2000.