Proceedings of I-KNOW '05
Graz, Austria, June 29 - July 1, 2005

Automated Task-Oriented Team Composition Using Description Logics

Simona Colucci
(Politecnico di Bari, Bari, Italy,and Knowledge Media Institute, MK, UK
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)

Azzurra Ragone
(Politecnico di Bari, Bari, Italy,
a.ragone@poliba.it)

Abstract: A logic-based approach to the semantic-based composition of task oriented teams using candidates profiles and task description is presented, in the framework of a skill management system. The selection process exploits Concept Covering, carried out using non-standard inference services in Description Logics. The approach is motivated and compared with other semantic-oriented proposals.
Key Words: Skill management, Concept Abduction, Concept Covering, Semantics
Category: I.2.4, K.6.1

1 Introduction

Knowledge intensive companies are characterized by having their most strategic asset in human resources. For such companies the return on investment is strongly affected by the management of available skills [14, 10]. In particular consulting firms usually carry out projects by engaging working teams, whose composition process becomes fundamental for efficiently employing knowledge and expertise of individuals. Composition of teams "on the fly", to work on specfic tasks is hence a challenging problem that skill management systems have to face. Large companies will probably have people with the required skills within the personnel, but also more difficult will be for them to rapidly individuate them. Skill management problems can be characterized in terms of multiplicity relationships between individuals skills and tasks to be carried out [2]; team composition amounts to a *one to many* relationship, *i.e.*, one task has to be assigned to several individuals that, altogether, cover the skills needed for task accomplishment. This problem can be dealt with under several perspectives and with various approaches [15, 7, 12, 17]; in this paper we propose a Description Logic (DL) [1] approach –fully compliant with the emerging Semantic Web vision– that exploits the recently devised Concept Abduction inference service and uses it to solve a Concept Covering problem, which can be seen as analogous –yet with noteworthy peculiarities– to a classical set covering problem in a DL framework. Our approach takes full advantage of structured, ontology-based, descriptions. It adopts an open world assumption, typical of knowledge representation, *i.e.*, 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.

2 A logic based approach

DLs [1] are a well known family of logic formalisms for knowledge representation. In this paper we refer to $\mathcal{ALN}$ (**A**ttributive **L**anguage with unqualified **N**umber restrictions), which can be mapped in a subset of OWL-DL. In [5] the Concept Abduction Problem (CAP) has been introduced and defined as a new non standard inference problem in DLs. Using Concept Abduction, given a TBox $\mathcal{T}$ and two concepts C and D it is possible to compute a complex concept H representing *what has to be hypothesized in C, and in a second step added to, in order to make C more specific than D*. In other words H is *what is expressed, explicitly or implicitly, in D and it is not present in C*, or in an equivalent way *which part of D is not covered by C*. A CAP is formally 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 $\mathcal{T} \models C \sqcap H \sqsubseteq D$.
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*. In [5] also minimality criteria for H and a polynomial algorithm to find solutions which are irreducible, for $\mathcal{ALN}$, have been proposed. A formal definition of *best covering problem* in DLs is given in [9]. Intuitively, given a concept C and a set of concept definitions in a TBox $\mathcal{T}$, find a set of concepts defined in $\mathcal{T}$such that

their conjunction can be considered an approximation of C. The definition exploits two non-standard inference services: the least common subsumer (*lcs*)[8] and the *difference* or *subtraction* operation [16]. The latter service asks for a DL with structural subsumption, which limits application to severeal DLs [16]. We propose an extension to the previous definition of Concept Covering Problem, eliminating limitations on the DL employed, and rewriting it in terms of Concept Abduction.

Definition 2. Let D be a concept, $\mathcal{R} = \{S_1, S_2, ..., S_k\}$ be a set of concepts, and $\mathcal{T}$ be a set of axioms, all in a DL $\mathcal{L}$, where D and $S_1, \ldots, S_k$ are satisfiable in $\mathcal{T}$. The *Concept Covering Problem* (CCoP) for $\mathcal{V}$=$\langle \mathcal{L}, \mathcal{R}, D, \mathcal{T} \rangle$ is finding a pair $\langle \mathcal{R}_c, H \rangle$ such that

1. $\mathcal{R}_c \subseteq \mathcal{R}$, and the conjunction of concepts in $\mathcal{R}_c$, $C = \sqcap_{S \in \mathcal{R}_c} S$ is satisfiable in $\mathcal{T}$;
2. $H \in SOL(\langle \mathcal{L}, C, D, \mathcal{T} \rangle)$, and $\mathcal{T} \not\models H \sqsubseteq D$.

We call $\langle \mathcal{R}_c, H \rangle$ a *solution* for $\mathcal{V}$, and say that $\mathcal{R}_c$ *(partially) covers* D. Finally, we denote $SOLCCoP(\mathcal{V})$ the set of all solutions to a CCoP $\mathcal{V}$.

Intuitively, $\mathcal{R}_c$ is the set of concepts that **partially** cover D w.r.t. $\mathcal{T}$, while the abduced concept H covers what is still in D and is not covered by C. From now on we denote CCoP with the symbol $\mathcal{V}$ and the set of all the solution to a CCoP $\mathcal{V}$with $SOLCCoP(\mathcal{V})$. There can be several solutions for a single CCoP, depending also on the strategy adopted for choosing concepts in $\mathcal{R}_c$. However, observe that – differently from the standard Set Covering Problem – a complete cover may not exist. Hence, minimizing the cardinality of $\mathcal{R}_c$ is not the aim of a CCoP; the aim is maximizing the covering, hence minimizing H. This argument leads us to the definition of *best cover* and *exact cover*.

Definition 3. A *best cover* for a CCoP $\mathcal{V}$, w.r.t. an order $\prec$ on $\mathcal{L}$, is a solution $\langle \mathcal{R}_c, H_b \rangle$ for $\mathcal{V}$ such that there is no other solution $\langle \mathcal{R}'_c, H' \rangle$ for $\mathcal{V}$ with $H' \prec H_b$.

In general, there could be two solutions $\langle \mathcal{R}_c, H_1 \rangle, \langle \mathcal{R}_c, H_2 \rangle$ such that the first is a best cover, while the second one is not. However, when a full cover exists, it is independent of any order $\prec$ on $\mathcal{L}$.

Definition 4. A *full cover* for a CCoP $\mathcal{V}$ is a solution $\langle \mathcal{R}_c, H_f \rangle$ for $\mathcal{V}$ such that $\mathcal{T} \models H_e \equiv \top$.

In the following we use the definition of CCoP to compose, given a task request D and a set of available individual profiles $\mathcal{R} = \{S_i\}$ modeled w.r.t. an ontology $\mathcal{T}$, a team *i.e.*, a subset $\mathcal{R}_c \subseteq \mathcal{R}$, such that individuals whose profile belongs to $\mathcal{R}_c$ are able to perform the requested task D as far as possible. We borrow from a classical greedy set-covering algorithm [4], to devise the *GREEDYsolveCCoP* algorithm, which can compute a solution to a CCoP, that is, create a team satisfying a task request. The algorithm takes as inputs the set $\mathcal{R}$ of available individual profiles S_i, the task to be covered D and the terminology $\mathcal{T}$, which S_i and D are described with reference to. *GREEDYsolveCCoP* tries to cover D as far as possible, using $S_i \in \mathcal{R}$.

Algorithm $GREEDYsolveCCoP(\mathcal{R}, D, \mathcal{T})$
input task D, individual profiles $S_i \in \mathcal{R}, i = 1..k$, 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
    $S_{min} = \top$;
    for each $S_i \in \mathcal{R}$
      if $\mathcal{R}_c \cup \{S_i\}$ *is a cover* for $D_{uncovered}$ **then**
        $H = solveCAP(\langle \mathcal{L}, S_i, D_{uncovered}, \mathcal{T} \rangle)$;
        if $H \prec H_{min}$ **then**
          $S_{min} = S_i$;
          $H_{min} = H$;
        end if
      end if
    end for each
    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
  while$(S_{min} \not\equiv \top)$;
  return $\langle \mathcal{R}_c, D_{uncovered} \rangle$;
end algorithm

In the main cycle, fundamental steps can be identified: First a greedy approach is used to choose the candidates for $\mathcal{R}_c$, that is, the individuals who can cover the task request without being inconsistent both with it and with the already created team; then the algorithm chooses among the candidates the one such that H, solution for the local CAP, is minimal w.r.t. an order $\prec$. At every step, the individual is chosen who contributes more in covering the remaining task $D_uncovered$; if the greedy search returns a new individual S_i, it is removed from $\mathcal{R}$ and added to the team $\mathcal{R}_c$; if no new useful profile is found $S_i \in \mathcal{R}$, that is any individual S_i such that it covers D more, the algorithm terminates.

For $\mathcal{ALN}$a polynomial algorithm (*findIrred*) [5] can be used to find irreducible solutions for a CAP, and in [6] the tractable *rankPotential* has been defined to rank concepts. Using calls to such algorithms, *GREEDYsolveCCoP* can be solved in polynomial time, too. Obviously we are not claiming that we solve a covering problem polynomially. The algorithm returns *a cover*, not the best one [1].

3 Illustrative example

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 team composed according to the following, simple, specifications: *engineers, with negotiation and communication skills and at least two years experience. In the project people are needed with Process Control, Web Technology and ERP systems, knowledge and oriented to team work.* Such a request can be formalized in DL as:

$D = \texttt{Engineer} \sqcap \exists\texttt{specialized} \sqcap \forall\texttt{specialized.}(\texttt{ProcessControl} \sqcap \texttt{WebBasedTechnology} \sqcap \texttt{ERPsystem}) \sqcap \exists\texttt{hasAbility} \sqcap \forall\texttt{hasAbility.}(\texttt{NegotiationSkills} \sqcap \texttt{CommunicationSkills} \sqcap \texttt{TeamCoordinator}) \sqcap \exists\texttt{hasExperience} \sqcap \forall\texttt{hasExperience.}(\geq\ 2\ \texttt{years})$

D is one of the inputs of $GREEDY solveCCoP(\mathcal{S}, D, \mathcal{T})$. Suppose now the four individuals described in the following are available as team members:

$Julia$: *Julia is a Process Engineer specialized in Business application, able to lead a team and coordinate meetings. She got a master degree recently.*

Tom : *Tom is a Web oriented Programmer. His favorite programming languages are Java and C++, used for web based application.*

$Richard$: *Richard is an engineer specialized in ERP systems. He has leading ability, particularly for working teams. He has a 3 years work experience.*

$Alison$: *Alison is an account manager specialized in IT-consulting and writing technical documents. She has a good knowledge of both English and German.*

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 $GREEDY solveCCoP(\mathcal{S}, D, \mathcal{T})$:

$S_1(Julia) = \texttt{ProcessEngineer} \sqcap \forall\texttt{specialized.BusinessApplication} \sqcap \forall\texttt{hasAbility.}(\texttt{MeetingsCoordination} \sqcap \texttt{TeamCoordinator}) \sqcap \texttt{NewGraduate}$

$S_2(Tom) = \texttt{WebProgrammer} \sqcap \exists\texttt{programmLanguageKnowledge} \sqcap \forall\texttt{programmLanguageKnowledge.}(\texttt{Java} \sqcap \texttt{C++}) \sqcap \forall\texttt{specialized.WebBasedTechnology}$

$S_3(Richard) = \texttt{Engineer} \sqcap \forall\texttt{specialized.ERPsystem} \sqcap \forall\texttt{hasAbility.}(\texttt{TeamCoordinator} \sqcap \texttt{Lead-role}) \sqcap \exists\texttt{hasExperience} \sqcap \forall\texttt{hasExperience.}(\geq\ 3\ \texttt{years})$

$S_4(Alison) = \texttt{AccountManager} \sqcap \forall\texttt{specialized.}(\texttt{ITConsulting} \sqcap \texttt{TechnicalWriting}) \sqcap \exists\texttt{hasLanguageKnowledge} \sqcap \forall\texttt{hasLanguageKnowledge.}(\texttt{English} \sqcap \texttt{German})$

[1] In [4] 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.

Applying $GREEDY solveCCoP(\mathcal{S}, D, \mathcal{T})$ for the example w.r.t. to the ontology $\mathcal{T}$ in Figure 1 the output will be $\langle \mathcal{R}_c, H \rangle$ with

- $\mathcal{R}_c = \{Tom, Richard\}$
- $H = \exists \texttt{specialized} \sqcap \forall \texttt{specialized.ProcessControl} \sqcap \exists \texttt{hasAbility} \sqcap \forall \texttt{hasAbility.}(\texttt{NegotiationSkills} \sqcap \texttt{CommunicationSkills})$

The above result corresponds to the following description: "*Using the available candidates, the system proposes a team composed by* Tom *and* Richard *to cover the proposed task. No one in the proposed team* is specialized in Process Control *and* has either communication or negotiation skills. *Furthermore, there is no one among the remaining candidates, due to the previously explained missing skills, who is compatible with the proposed task.* ". Notice that the composed team is strongly dependent on the request. In fact with respect to the previous result, if the Company retracts on the $\exists \texttt{hasExperience} \sqcap \forall \texttt{hasExperience.}(\geq\ 2\ \texttt{years})$ requirements and it reformulates the request as follows:

$D = \texttt{Engineer} \sqcap \exists \texttt{specialized} \sqcap \forall \texttt{specialized.}(\texttt{ProcessControl} \sqcap \texttt{WebBasedTechnology} \sqcap \texttt{ERPsystem}) \sqcap \exists \texttt{hasAbility} \sqcap \forall \texttt{hasAbility.}(\texttt{NegotiationSkills} \sqcap \texttt{CommunicationSkills} \sqcap \texttt{TeamCoordinator})$

the new composition result is:

- $\mathcal{R}_c = \{Tom, Richard, Julia\}$

It easy to see that the team covers all the requirements now, because of the selection of Julia. She was not in the previous team because her work experience information was incompatible with the request.

4 Discussion and conclusion

Although a complete review of related work on the subject is beyond the scope of this paper, we just comment on an alternative approach, which may appear close to our own [11]. The approach builds on the technique presented in [13] for ranking query results. The relevance of query results is computed taking into account the structure of the underlying domain (knowledge base content) and the inferencing process in which the answer is implied. The ranking, though providing a useful support to the choice between the returned answers, only classifies answers to queries formalized w.r.t. a well defined structure. Such an approach lacks then of expressiveness in the querying process. Moreover it lacks of the open world assumption, because only answers that explicitly provide the characteristics required by the query are ranked and it does not explain the rationale for the absence of match.

$\texttt{Graduate} \equiv \exists\texttt{hasMasterDegree}$
$\texttt{NewGraduate} \equiv \texttt{Graduate} \sqcap \forall\texttt{hasExperience}.\bot$
$\texttt{Engineer} \equiv \texttt{Graduate} \sqcap \forall\texttt{hasMasterDegree.Engineering}$
$\texttt{ProcessEngineer} \equiv \texttt{Engineer} \sqcap \exists\texttt{specialized} \sqcap \forall\texttt{specialized}.(\texttt{ProcessControl} \sqcap \texttt{PlantManagement})$
$\texttt{Programmer} \sqsubseteq \exists\texttt{hasTechnicalSkills} \sqcap \forall\texttt{hasTechnicalSkills.Programming}$
$\texttt{WebProgrammer} \sqsubseteq \texttt{Programmer}$
$\texttt{Lead-role} \sqsubseteq \texttt{Ability}$
$\texttt{TeamCoordinator} \sqsubseteq \texttt{Ability}$
$\texttt{NegotiationSkills} \sqsubseteq \texttt{Ability}$
$\texttt{CommunicationSkills} \sqsubseteq \texttt{Ability}$
$\texttt{MeetingsCoordination} \sqsubseteq \texttt{CommunicationSkills} \sqcap \texttt{NegotiationSkills}$
$\texttt{ERPsystem} \sqsubseteq \texttt{BusinessApplication}$
$\texttt{FinancialSystem} \sqsubseteq \texttt{BusinessApplication}$
$\texttt{WebBasedTechnology} \sqsubseteq \texttt{Technology}$
$\texttt{ProgrammingLanguage} \sqsubseteq \texttt{Technology}$
$\texttt{Java} \sqsubseteq \texttt{ProgrammingLanguage}$
$\texttt{C++} \sqsubseteq \texttt{ProgrammingLanguage}$
$\texttt{Programming} \sqsubseteq \texttt{TechnicalSkills}$
$\texttt{ITConsulting} \sqsubseteq \texttt{TechnicalSkills}$
$\texttt{TechnicalWriting} \sqsubseteq \texttt{TechnicalSkills}$
$\texttt{English} \sqsubseteq \texttt{Language}$
$\texttt{German} \sqsubseteq \texttt{Language}$

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

We faced the problem of semantic-based composition of task oriented teams exploiting Concept Abduction to compute a Concept Covering problem. Our approach has at least two remarkable features: it carries out a covering based on the actual semantics of descriptions; it automatically provides explanations on what is still missing when a full cover is not accomplished. Furthermore, at least for $\mathcal{ALN}$ a subset of OWL-DL, is polynomial in time. One may wonder whether such a process simply amounts to a different "fashionable" approach to issues that can be dealt with classical set covering. It is not so, as we argue a semantic-based approach to team composition is more powerful than a simple set based one. Just to motivate the claim with the minimum evidence, consider the following set-based example where we are looking for {`Engineer`, `ProcessControl`, `WebBasedTechnology`, `ERPsystem`} and the information about $Julia$ is represented as {`ProcessEngineer`, `BusinessApplication`, `MeetingsCoordination`, `TeamCoordinator`}. Although $Julia$ is a `ProcessEngineer`, in a set-based setting her skills would not cover –not even partially– the request because nothing is known about the semantics of both `ProcessEngineer` and `Engineer` and their relation with each other. Our covering algorithm, instead, *gets into* the semantics of the description, and $Julia$ would fit into the profile for the given task.

Also notice that, even if descriptions were represented as individuals in the ABox rather then concept descriptions w.r.t. the TBox, our approach is actually still valid. In fact it is sufficient to take into account the concept descriptions as instantiated by each individual. Furthermore classical reasoning services on ABox, such as instance checking and retrieval do not cope with the problems we addressed in this paper, and services as the ones adopted here have not –to the best of our knowledge– ever beeen defined or implemented. The approach presented here is part of an ongoing project for a complete knowledge-based skill management system, which faces the various issues of the probem as multiplicity relationships between individuals skills and tasks [2, 3], adopting DLs formalisms and services.

Acknowledgments

We acknowledge partial support of MIUR projects MS3DI, PITAGORA and CNOSSO.

References

1. F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook.* Cambridge University Press, 2002.
2. 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.
3. 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 (J.UCS)*, 10(6), 2004.
4. T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT, 1990.
5. T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. Abductive matchmaking using description logics. In *Proc. of IJCAI'03*, 2003.
6. T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. A system for principled Matchmaking in an electronic marketplace. *International Journal of Electronic Commerce*, 8(4):9–37, 2004.
7. 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.
8. F.Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In *Proc. of IJCAI'03*, pages 319–324, 2003.
9. M-S. Hacid, A. Leger, C. Rey, and F. Toumani. Computing concept covers: a preliminary report. In *Proc. of DL'02*, 2002.
10. G. Hamel and C. K. Prahalad. The core competence of the corporation. *Harvard Business Review*, pages 79–91, 1990.
11. 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*, 2004.
12. T. Lau and Y. Sure. Introducing ontology-based skills management at a large insurance company. In *Proc. of the Modellierung*, 2002.
13. R. Studer N. Stojanovic and L. Stojanovic. An approach for the ranking of query results in the semantic web. In *Proc. ISWC '03*, 2003.
14. O'Leary. Enterprise knowledge management. *IEEE Computer*, 31(3), 1998.
15. D. OLeary. Using AI in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems*, 13(3):34–39, 1998.
16. G. Teege. Making the difference: A subtraction operation for description logics. In *Proc. of KR'94*, 1994.
17. I. H. Toroslu. Personnel assignment problem with hierarchical ordering constraints. *Computer and Industrial Engineering*, 45(3):493–510, 2003, October.