Journal of Universal Computer Science, vol. 13, no. 9 (2007), 1184-1212
submitted: 15/8/06, accepted: 1/3/07, appeared: 28/9/07 © J.UCS

Semantic-based Skill Management for Automated Task Assignment and Courseware Composition

Simona Colucci
(SisInfLab–Politecnico di Bari, Bari, Italy, and D.O.O.M. s.r.l., Matera, Italy
s.colucci@poliba.it)

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

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

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

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

Abstract: Knowledge management is characterized by many different activities ranging from the elicitation of knowledge to its storing, sharing, maintenance, usage and creation. Skill management is one of such activities, with its own peculiarities, as it focuses on full exploitation of knowledge individuals in an organization have, in order to carry out at best given tasks. In this paper a semantic-based automated Skill Management System is proposed, which supports competences search and creation. The system implements an approach exploiting the formalism and the reasoning services provided by Description Logics. The approach embeds also non standard Description Logics reasoning services to extend the set of provided features. Here we present main characteristics of our system and focus on a novel algorithm exploiting advanced inference services for the one-to-one assignment of a set of individuals to a set of tasks, endowed of logical explanation features for missing/conflicting skills.

Key Words: Skill Management, Competence creation, E-learning, Knowledge representation, Description Logics.

Category: I.2.4., K.6.1., I.2.1.

1 Introduction

Skill Management is a specific area of Knowledge Management, which has recently gained attention, both in industry and academia, as knowledge intensive companies –particularly consulting companies– strive to fully benefit from the know-how their personnel holds, and try to match effectively the right person with the right task in minimum time. The competences of the workforce have been nowadays recognized as strategic assets of paramount importance in the achievement of competitive advantage [Hamel and Prahalad, 1990]. Other studies [Gronau and Uslar, 2004] show that the return on investment is significantly impacted by enriching knowledge management systems companies use, with components for the specific management of skills(*Skill Management Systems*, SMS).

The management of skills, anyway, involves a large number of different activities ranging form the elicitation of knowledge held by individuals to all the possible usages of such a formalized know-how. In [Draganidis and Mentzas, 2006] a recent survey of systems and approaches to competency management is proposed, which testifies the interest in the problem and the diversity of approaches taken.

In recent years we have been working on an integrated framework and system along the lines traced by the Semantic Web initiative [Berners-Lee et al., 2001]. In particular our aim is to fully benefit from structuring available information using formal languages such as OWL-DL, to build an infrastructure where skills and tasks can best match. Obviously, there are cases when available skills within the workforce are not sufficient to cope with needs; in this case there are two basic possibilities: either hire new personnel or **increase** skills of internal personnel through targeted and specific learning procedures. The availability of an increasing number and variety of effective e-learning modules may make this second option appealing and cost-effective. Our semantic-based approach tries to smoothly integrate the skills finding with the possibility to provide "new" knowledge when needed.

We characterize a skill matching problem in terms of multiplicity relationships between assignees and tasks to be accomplished [Colucci et al., 2003b]. In this paper we illustrate our framework and system, with particular reference to a novel algorithm for the one-to-one assignment of a set of individuals to a set of tasks (that we call multiple one-to-one), able to deal with partial matches, endowed of logical explanation features for missing/conflicting skills.

The rest of the paper is structured as follows: next section outlines our framework and the scenario our system deals with; section 3 recalls inference services that are used in our approach; then we present, in section 4, devised algorithms for skill matching in the various categories we consider. Section 5 illustrates our system behavior. Section 6 discusses relevant related work and conclusions close the paper.

2 Framework and features

In this paper we present a Skill Management System (SMS) mainly focused on semantic-based assignment of individuals to tasks and creation of new competence inside a group of holders. The scenario is one in which the management of a company has some knowledge-intensive tasks to face, *e.g.*, a consulting company that has received an order. It may exploit both the internal know-how, by employing its internal personnel, and the knowledge provided by consultants, to be hired from outside the company. Another possibility is open by internal personnel training: when the required knowledge is not available inside the company the management may revert to knowledge creation by asking employees to learn the lacking competences.

The system has to be automated as fully as possible, so knowledge needs to be formalized unambiguously through a machine understandable language, yet it still has to work as a decision support system, leaving when necessary, some choices to the management.

Description Logics (DL)[Baader et al., 2002] formalism is used to represent knowledge in our framework. DLs provide a number of standard reasoning services helpful in the process of using knowledge; moreover we propose non-standard reasoning services from DLs, which provide peculiar logical explanation features to our system.

The proposed SMS ensures knowledge elicitation, sharing, storing and maintenance through the use of formal languages. The shared vocabulary for the knowledge domain is given by the ontology skills descriptions are referred to. Descriptions of skills in such an ontology are relative to three kinds of knowledge domain entities:

- **knowledge providers**: describes entities able to achieve tasks by making their knowledge available(persons in most of the cases)
- **tasks**: describes required knowledge for activities to be performed
- **learning objects**: describes knowledge to be gained thanks to the fruition of e-learning modules.

The HR-XML Consortium work group[1] defines a **competency** as *a specific, identifiable, definable, and measurable knowledge, skill, ability and/or other deployment-related characteristic (e.g. attitude, behavior, physical ability) which a human resource may possess and which is necessary for, or material to, the performance of an activity within a specific business context.*

In the description of our domain of interest we accept such a definition, especially in its emphasis on the connection between competencies and activity performance. In particular, the descriptions we introduce as case study in the paper are mostly based on measurable skills possessed by human resources. Such a choice causes the term skill to be often used as a synonym of competency all over the paper.

[1] The HR-XML Consortium. http://hr-xml.org

The representation of knowledge domain is exploited by the algorithms implemented by our SMS to perform a semantic based assignment of knowledge providers to tasks and a semantic based composition of learning objects to cover the skills required by the task and missed by the providers.

Such an integration of assignment with learning process is made possible thanks to the explanation of missing skills our SMS provides.

In SMS literature many systems have been in fact proposed both for skill matching and for courses composition; nevertheless also systems exploiting the semantics of skills descriptions are not able to give explanations either on the reasons for possible mismatches or on those parts of the task which remains uncovered in case of not-full match.

Additionally our SMS provides the possibility of revising skill requests if no provider is potentially able to satisfy task requirements; the revision process starts by the elicitation of conflicting information in the request with each provider.

Both explanation and revision exploit non standard reasoning services from DL, which are detailed in Section 3.

Our SMS provides different choices for assignment of knowledge providers to tasks depending, as hinted before, on the multiplicity of match relation and on the effort on simultaneous optimization. In particular, a task may be assigned to one or more providers (and vice versa a provider may be assigned to one or more tasks) and in the matching process the management may decide to take or not into account possible other tasks to assign.

Such a choice is up to the system user and generates different matching processes, whose output in each case includes an assignment and a logical explanation describing the skills not yet covered by the chosen providers.

Such missing skills can be used to determine the learning need to be covered by a different learning process, automatically suggested by the system to each provider by composing available learning objects formalized in the ontology.

The proposed learning processes are different for each provider, as they strictly depend on the background knowledge providers hold. The system is also able to suggest the provider whose learning process requires the least effort; in other words the candidate for which the process of covering the missing skills is expected to be simpler.

3 Basic and Nonmonotonic Inferences

In the following we will refer to Description Logics (DL) whose formal semantics is the basis of the Ontology Web Language OWL-DL [OWL, 2004], and model a DL-based framework to cope with the issues presented in the following sections. OWL has been conceived to allow for representation of machine understandable, unambiguous, description of web content through the creation of domain ontologies, and aim at increasing openness and interoperability in the web environment. The strong relations between DLs and the above introduced language for the Semantic Web [Baader et al., 2003] is also evident in the definition of the three OWL sub-languages:

OWL-Lite: allows class hierarchy and simple constraints on relation between classes;

OWL-DL: is based on Description Logics theoretical studies, it allows a great expressiveness keeping computational completeness and decidability;

OWL-Full: using such a language, there is a great syntactic flexibility and expressiveness. This freedom is paid in terms of no computational guarantee.

The sub-language OWL-DL is expressive enough to map the subset of DLs formalism we exploit for knowledge representation (see Table 1 in Appendix A).

The interested reader may refer to [Baader et al., 2002] for a comprehensive survey on DLs [2]. In this section we just recall inference services in DLs useful in the rest of the paper, taking into account an ontology $\mathcal{T}$. The basic reasoning problems for concepts in a DLs are satisfiability, which accounts for the internal coherency of the description of a concept (no contradictory properties are present), and subsumption, which accounts for the more general/more specific relation among concepts, that forms the basis of a taxonomy.

Definition 1 ***Subsumption.*** Let $\mathcal{L}$ be a DL, P and T be two concept in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$. A concept T subsumes a concept P w.r.t. $\mathcal{T}$ if every interpretation of $\mathcal{T}$ assigns to P a subset of the set assigned to T. We write $\mathcal{T} \models P \sqsubseteq T$ to indicate T subsumes P w.r.t. $\mathcal{T}$.

As an example, consider $\mathcal{T}$ as the skills domain ontology, and T (for Task) and P (for Profile) two concept descriptions representing respectively a request for a task to be assigned and the skills extracted from a curriculum vitae (CV), more generally a knowledge provider. If $\mathcal{T} \models P \sqsubseteq T$, *i.e.*, the information represented by P are more specific than the ones requested in T, it means the curriculum's owner has at least all the skills required to execute the requested task and a full match occurs. Obviously, this is the best match case.

Definition 2 ***Satisfiability.*** Let $\mathcal{L}$ be a DL, P be a concept in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$. P is satisfiable w.r.t. $\mathcal{T}$ if there exists at least one model of $\mathcal{T}$ assigning a non-empty extension to P. Since a concept P is satisfiable w.r.t. $\mathcal{T}$ iff P is not subsumed by $\bot$, we write $\mathcal{T} \models P \not\sqsubseteq \bot$ to indicate P is satisfiable w.r.t. $\mathcal{T}$.

[2] A small appendix is provided at the end of the paper that briefly illustrates the specific DL we adopt in our system

In the task/skills scenario of the above example, if $\mathcal{T} \models P \sqsubseteq \bot$ (respectively $\mathcal{T} \models T \sqsubseteq \bot$) then the description P (respectively T) is in conflict with the information modeled in $\mathcal{T}$. That is, the request (respectively the curriculum) is self-contradicting. On the other hand, if $C \equiv P \sqcap T$, with P and T satisfiable w.r.t. $\mathcal{T}$, the unsatisfiability of C can be read as an incompatibility of P and T.

Although very useful in many matchmaking settings, both subsumption and satisfiability return a yes/no answer. The scenario we outlined requires instead both explanation and belief revision in order to cope with cases in which no perfect match exists. Hereafter we recall basic definitions of two non-standard inference services that will be used to overcome highlighted limitations of subsumption and satisfiability; for a thorough discussion of the rationale of such inferences see [Di Noia et al., 2007, Colucci et al., 2007b]. If $P \sqcap T$ is unsatisfiable in the ontology $\mathcal{T}$, *i.e.*, the task and the profile are not compatible with each other, we may want, as in a belief revision process, to retract some requirements G (for *Give up*) in T, to obtain a new contracted task request K (for *Keep*) which is compatible with P. In other words, such that $K \sqcap P$ is satisfiable in $\mathcal{T}$.

Definition 3 ***Concept Contraction.*** Let $\mathcal{L}$ be a DL, P, T, be two concepts in $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms in $\mathcal{L}$, where both P and T are satisfiable in $\mathcal{T}$. A *Concept Contraction Problem* (CCP), identified by $\langle \mathcal{L}, T, P, \mathcal{T} \rangle$, is finding a pair of concepts $\langle G, K \rangle \in \mathcal{L} \times \mathcal{L}$ such that $\mathcal{T} \models T \equiv G \sqcap K$, and $K \sqcap P$ is satisfiable in $\mathcal{T}$. We call K a *contraction* of T according to P and $\mathcal{T}$.

Obviously, there is always the trivial solution $\langle G, K \rangle = \langle T, \top \rangle$ to a CCP, that is give up everything of T. In our skill matching framework, it models the (infrequent) situation in which, in front of some very appealing profile P, incompatible with the requested task, a recruiter just gives up completely her specifications T in order to meet P. On the other hand, when $P \sqcap T$ is satisfiable in $\mathcal{T}$, the "best" possible solution is $\langle \top, T \rangle$, that is, give up nothing — if possible. Since usually one wants to give up as few things as possible, some minimality in the contraction must be defined [Gärdenfors, 1988]. In most cases a pure logic-base approach could be not sufficient to decide between which beliefs to give up and which to keep. There is the need of modeling and defining some extra-logical information. One approach is to give up minimal information [Colucci et al., 2003a]. Another one sets different importance levels for demands characteristics, modeling them as negotiable or strict constraints [Di Noia et al., 2004a].

When P and T are satisfiable w.r.t. each other (the task and the profile do not contain conflicting information) but subsumption does not hold (*i.e.*, a full match is unavailable) one may want to hypothesize some explanation on which are the causes of this result.

Definition 4 ***Concept Abduction.*** Let P, T, be two concepts in a Description Logic $\mathcal{L}$, and $\mathcal{T}$ be a set of axioms, where both P and T are satisfiable in $\mathcal{T}$. A

Concept Abduction Problem (CAP), denoted as $\langle \mathcal{L}, T, P, \mathcal{T} \rangle$, is finding a concept H such that $\mathcal{T} \not\models P \sqcap H \equiv \bot$, and $\mathcal{T} \models P \sqcap H \sqsubseteq T$.

The solution to a CAP [Di Noia et al., 2003] can be interpreted as what has to be hypothesized in P, and in a second step added to, in order to make P more specific than T, which would make subsumption result true. If a CV is compatible with respect to a requested task — $\mathcal{T} \not\models P \sqcap T \sqsubseteq \bot$ — but the latter is not completely satisfied by the former, it may be very interesting to know which part of the task is not covered by the CV. Notice that the absence of information in the curriculum P has not to be interpreted using a negation by default approach. In an *open world semantics*, it is interpreted as under-specified information: something the CV's owner forgot to describe or simply did not care about. That is why we formulate hypotheses (using an abductive process) on what is unknown.

3.1 Logic based ranking function

In a retrieval process, as the one for required-task/knowledge-provider, a ranking function is needed in order to establish the suitability of an offered resource with respect to a request.

Given a task request T and a knowledge provider P, both described w.r.t. an ontology $\mathcal{T}$, in order to have T completely satisfied by P and reach a full match, the subsumption relation $\mathcal{T} \models P \sqcap T$ must hold. If a full match does not occur, we can have either T and P compatible with each other — $\mathcal{T} \not\models P \sqcap T \sqsubseteq \bot$ — or, in the worst case, incompatible — $\mathcal{T} \models P \sqcap T \sqsubseteq \bot$. In case of compatibility, solving a CAP, hypotheses can be provided in order to formulate possible explanations H on why a full match does not occur; in case of incompatibility, solving a CCP, possible explanations G on why P and T are incompatible can be also computed and a contracted request K provided, which is compatible with the knowledge provider P. More formally the following two situations may ensue:

$$\mathcal{T} \models P \sqcap T \sqsubseteq \bot \Longrightarrow \langle G, K \rangle = solveCCP \Longrightarrow \mathcal{T} \not\models P \sqcap K \sqsubseteq \bot \tag{1}$$

$$\mathcal{T} \not\models P \sqcap T \sqsubseteq \bot \Longrightarrow \quad H = solveCAP \quad \Longrightarrow \mathcal{T} \models P \sqcap H \sqsubseteq T \tag{2}$$

Looking at (1) and (2) it is easy to notice that in both cases we are able to compute explanations on mismatch. Trivially, if P and T are compatible, then H is the explanation; if P and T are incompatible, then solving a CCP on P and T we compute both incompatibility explanation G and a new request K compatible with T. Now, solving a CAP on P and K explanation H' on why a full match does not occur between P and the new K is computed. Summarizing, G and H' are the explanations on mismatch w.r.t. a full match in case of incompatibility.

Numerical measures of the above mismatch explanations can be obviously useful to evaluate a match score, and then an ordering, in our skill matching framework. Then, given a task request T and a knowledge provider P both satisfiable w.r.t. an ontology $\mathcal{T}$, we define a match degree function U (for utility) as:

$$U : \langle G, H, \mathcal{T} \rangle \longrightarrow \Re \tag{3}$$

U can be used both in case of incompatibility between P and T and in case of compatibility, and in the latter case $G = \top$. Notice that U takes into account also the ontology $\mathcal{T}$. In fact in $\mathcal{T}$, the semantics of G and H is modeled. In Section 5 we will present a function U for the $\mathcal{ALN}$(see Appendix A) subset of OWL-DL.

4 Automated Task Assignment and Courseware Composition

The first issue we consider is the choice on the match multiplicity ad on the contemporary optimization of different assignments. Three different matching processes may occur depending on the previous choice:

- Single one to one matching
- Multiple one to one matching
- Many to one matching

4.1 The Single One to One Matching Process

In [Colucci et al., 2007a] we proposed the algorithm *Assign* for single one to one skill matching; the algorithm automatically assigns, to the only task T taken into account, the provider P_i minimizing a function U measuring both explanation hypotheses H and belief revision needs G as explained in the previous section.

4.2 The Multiple One to One Assignment Problem

With reference to the multiple one to one matching scenario we proposed a solving approach in [Colucci et al., 2004]. The paper takes a semantic based approach to the solution of the classical Assignment Problem in operational research [Cormen et al., 1990]. Such a problem usually performs the assignment by minimizing an objective function expressing the global cost. In our contribution we introduced a method to maximize the suitability of assignees to tasks instead of minimizing the cost of matching. In particular the proposed objective function only took into account compatible profiles P_j with respect to given task requests T_i. By choosing the assignment minimizing such a function, the system optimizes the suitability of individuals to tasks.

In such an approach only a measure of the underspecified skills explanation concept was taken into account.

Here we show how to extend the approach, making our system able to cope with cases of incompatible profiles and take also belief revision procedures into account.

The formal definition of the new problem arising can be summarized as follows:

Minimize

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

subject to

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

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

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 individual is assigned to exactly one task, whereas the second set forces each task to be performed by exactly one individual. Such constraints cause the decision variables to be independent of each other in the formulation of the problem.

Coefficients u_{ij} are computed according to the function U previously introduced and denote the suitability of individual i to job j and take the place of cost coefficients c_{ij} in the problem general model.

In the new model the assignment is evaluated both for the concepts T_i, P_j that potentially match and for those that match only partially. The function U takes in fact into account both the measure of the concept to contract and of the one to hypothesize in order to compute the suitability as explained in Section 3.1.

The algorithm *MultipleAssign* shown in the following performs the multiple one to one matching process. It takes as input both a set of tasks to perform $\mathcal{R} = \{T_i\}$ with $i = 1 \dots n$ and a set of knowledge providers descriptions $\mathcal{P} = \{P_j\}$ with $j = 1 \dots m$. The output is made up by a set of quadruples $A = \{(T_i, P_j, H_{ij}, G_{ij})\}$ containing respectively the task to assign, the chosen assignee, the concept H_{ij} representing the knowledge to hypothesize in the assignee description to perfectly match the task, and G_{ij} representing the requests to contract in the task, to potentially match the assignee description.

1: **Algorithm** *MultipleAssign*$(\mathcal{R}, \mathcal{P}, \mathcal{T})$
2: **input** $P_j \in \mathcal{P},\ T_i \in \mathcal{R}$, concepts in $\mathcal{L}$ such that both
    $\mathcal{T} \models P_j \not\sqsubseteq \bot$ and $\mathcal{T} \models T_j \not\sqsubseteq \bot$
3: **output** set of quadruples $A = \{(T_i, P_j, H_{ij}, G_{ij})\}$
4: **begin algorithm**
5:   **for each** $T_i \in \mathcal{R}$
6:     **for each** $P_j \in \mathcal{P}$
7:       **if** $\mathcal{T} \models T_i \sqcap P_j \equiv \bot$ **then**
8:         $\langle G_{ij}, K_{ij} \rangle = contract(P_j, T_i, \mathcal{T})$;
9:       **else**
10:         $\langle G_{ij}, K_{ij} \rangle = \langle \top, T_i \rangle$;
11:       **end if**
12:       $H_{ij} = abduce(P_j, K_{ij}, \mathcal{T})$;
13:       $u_{ij} = U(H_{ij}, G_{ij}, \mathcal{T})$;
14:     **end for each**
15:   **end for each**
16:   $x_{ij} = solveKhun(u_{ij})$
17:   **for each** i and j
18:     **if** $x_{ij} = 1$
19:       **then** $A_{ij} = (T_i, P_j, H_{ij}, G_{ij})$
20:     **end if**
21:   **end for each**
22:   **return** $A = \{A_{ij}\}$;
23: **end algorithm**

The algorithm computes the values of function U for each pair individual-task and uses such values u_{ij} as coefficients of the objective function of the Assignment Problem.

The Problem is then solved by adopting the well known Kuhn algorithm [Kuhn, 1955]; *MultipleAssign* calls Khun solving algorithm in row 16: the coefficients u_{ij} are given as input to $solveKhun$ algorithm, which returns the set of solution variables x_{ij}. *MultipleAssign* finally returns the set A, whose elements are the assignment quadruples corresponding to each $x_{ij} = 1$.

4.3 The Many to One Assignment Problem

In [Colucci et al., 2007a], the algorithm *TeamComposer* was proposed to cope with the many to one assignment problem. The algorithm uses a greedy approach to the team composition process and exploits the minimization of function U to choose the candidates to team composition. The algorithm takes as input a task T to be solved and a set of knowledge providers $\mathcal{P} = \{P_j\}$ with $j = 1 \dots m$ and returns the set P_c of employees composing the team, the part $T_{uncovered}$ of the task description not covered by the ad-hoc created team and the part $G_{contraction}$ of the task description to be given up at the end of the whole team composition process.

4.4 Courseware Composition

All of the three matching processes outlined so far return an explanation concept describing skills still not available after the assignment. Such missing skills represent the learning need at the basis of a possible learning process proposed by automatically composing available learning objects to create a personalized courseware.

In [Colucci et al., 2005c] we proposed a general framework based on semantic technologies for composition using Concept Covering via Concept Abduction; such framework can be easily integrated in existing metadata specifications, such as SCORM [SCORM, 2004], LOM [IEEE, 2002], IMS [IMS, 2001], Dublin Core [DublinCore, 1999], although we currently use a LOM extended header. The courseware composition was there devised as a learning objects (λ) retrieval problem. In that perspective, if there is a learning need and a repository of learning objects potentially satisfying the learner specifications, a solution to a λ-retrieval problem is:

retrieve (a sequence of) some λs from the repository such that their composition satisfies the learning need as far as possible.

In case a perfect covering of the learning need is not found, an approximate solution has to be taken into account, together with explanation hypothesis of what remains missing. In this case, missing information represent what the courseware does not specify to teach w.r.t. the learning needs. This can be due to:

- underspecification of the λ description
- lack of learning objects coping with the requires learning needs
- not sufficient background knowledge of the learner

Formally learning objects and learning needs are defined as follows:

Learning Object: [3] $\lambda = \langle \lambda_D, \lambda_{\mathcal{BK}} \rangle$. *$\lambda_D$ describes the knowledge the user will acquire after λ fruition. Using a language endowed with a well-defined semantics, it models the* offered knowledge. *$\lambda_{\mathcal{BK}}$ is a representation of prerequisites in order to benefit from λ.*

Learning Need: $\rho = \langle \rho_D, \rho_{\mathcal{BK}} \rangle$, *where ρ_D is the description of the requested learning need and $\rho_{\mathcal{BK}}$ represents the background knowledge owned by the requester before looking for the courseware.*

Obviously, we did not take belief revision into account during courseware composition, as the situation in which the learning need is incompatible with a learning object description may not ensue. The skills ontology never models in fact different competences as disjoint concepts, because of the nature of knowledge itself: knowledge about a given skill is always compatible with any other sort of knowledge.

Our system proposes a different courseware to each provider in order to cover the learning need resulting from the assignment process. Such a personalization is needed because of the composition of learning objects in a background knowledge and a description. Only the providers holding the required prerequisites may be asked to learn topics detailed in the learning object description.

The algorithm *teacher* presented in [Colucci et al., 2005c] automatically computes each composite courseware. The algorithm takes as input a set of learning objects $\mathcal{R} = \{\lambda^i = \langle \lambda^i_D, \lambda^i_{\mathcal{BK}} \rangle\}$, the learning need $\rho = \langle \rho_D, \rho_{\mathcal{BK}} \rangle$, and an ontology $\mathcal{T}$ and returns the composite courseware $\Lambda(\rho, \mathcal{R})$ and the uncovered part, $\rho_{D_{uncovered}}$, of the request description ρ_D.

A composite courseware is hence a sequence of learning objects such that both the following conditions hold: it can be started using some background knowledge the requester owns ($\rho_{\mathcal{BK}}$) and the provided composite courseware covers the user request description (ρ_D).

Our system supports user's decision also in choosing the learning process which requires the least effort for covering the learning need, given that different personalized processes are possible. In our opinion a completely automatic selection is not the most suitable solution in this phase, because several highly subjective choice factors have to be taken into account. The information needed for the choice may be considered embedded in the following factors:

– **courseware complexity**: each proposed composite courseware is characterized by a complexity which cannot be measured by only objective features. Some factors like the number of composing learning objects and the time needed to learn the whole courseware can be objectively compared and an automatic learner choice can be made by the system according to such

[3] Without loss of generality here we consider only the information needed for a semantic discovery and composition.

factors. Nevertheless some factors are highly subjective, like the familiarity of the individual to the topic or the specificity of the skills to learn. By presenting to the user the semantic based descriptions of all the composite coursewares we believe s/he can evaluate their complexity by taking both objective and subjective factors into account.

– **evaluation of missing skills**: a measure of the concept $\rho_{D_{uncovered}}$ is needed in order to evaluate how relevant are the skills the learner will still be lacking after the courseware fruition. Such measure has to take into account also how specific is $\rho_{D_{uncovered}}$: knowledge about generic *object programming languages* may be gained more easily than specific competence about *Java*.

– **additional knowledge learned by courseware fruition**: the proposed courseware may be more specific than the learning need. Such a situation, not affecting the candidate choice at a first sight, is instead noteworthy. The extra knowledge gained may be a factor of selection among possible learners. The explanation on extra knowledge is returned by solving a Concept Abduction Problem between the learning object description and the learning need: $H_i = abduce(\Lambda(\rho, \mathcal{R}), \rho_D, \mathcal{T})$

Instead of proposing an automatic learner selection we implemented the approach presenting an explanation facility for these three factors to the system user, so making available the whole information relevant for her decision. The final selection of the candidate is then up to the system user.

5 System behavior

In this section we outline our system behavior for the $MultipleAssign$ algorithm illustrated in the previous section, with the aid of a simple example. We begin by describing the way our system computes a semantic match degree.

5.1 Match Degree Function for $\mathcal{ALN}$

Given a Concept Abduction Problem (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 [Di Noia et al., 2003] CAP was introduced for the first time and also a minimality criteria for H and a polynomial algorithm to find solutions which are irreducible, for $\mathcal{ALN}$ (see Appendix A) subset of OWL-DL, have been proposed. In [Di Noia et al., 2004b] $rankPotential$ was originally proposed to evaluate a numerical score given an irreducible solution to a CAP w.r.t. to an ontology $\mathcal{T}$. Based on $rankPotential$, the function U was originally introduced in [Colucci et al., 2005a] and computed according to the following closed form:

$$U(\langle T, P, G, H, \mathcal{T}\rangle) = \left|1 - \frac{N}{N-g} * (1 - \frac{h}{k})\right|$$

with the following meaning for parameters:

k : evaluation of K that belongs to the solution of a concept contraction problem between P_i and T — $k = rankPotential(\top, K, \mathcal{T})$

h : evaluation of H solution of a concept abduction problem between $K(T$ if no contraction is needed) and P — $h = rankPotential(P, K, \mathcal{T})$

g : evaluation of G that belongs to the solution of a concept contraction problem between P and T — $g = rankPotential(K, T, \mathcal{T})$

N : evaluation of T — $N = rankPotential(\top, T, \mathcal{T})$

By choosing the candidate minimizing U, the algorithm takes into account both g and h, *i.e.*, a numerical measure of how much it has to be given up in the request T and how much to hypothesize in the profile P analyzed.

5.1.1 An Illustrative Example

We present here an illustrative example of $MultipleAssign$ to better clarify its behavior. All the descriptions are modeled w.r.t. the simplified ontology $\mathcal{T}$ shown in Figure 1; for compactness reasons, we straightforwardly adopt DL formalization.

The model of skills we provide in the ontology can be integrated with the draft standard [IEEE, 2007] proposing a data model for Reusable Competency Definition(RCD) [IMS, 2002]. In particular our skill descriptions fill the *definition* element of RCD data model.

Let $\mathcal{R}$ be composed by the following task descriptions:

– $T_1 = \exists \mathtt{basicKnowledge} \sqcap \forall \mathtt{basicKnowledge}.$ $(\mathtt{InternetUse} \sqcap \mathtt{MarkupLanguages})$

– $T_2 = \exists \mathtt{advancedKnowledge} \sqcap \forall \mathtt{advancedKnowledge}.$ $(\mathtt{ClientServerProtocol} \sqcap \mathtt{ProcessManagement}) \sqcap \exists \mathtt{toolsKnowledge}$ $\sqcap \forall \mathtt{toolsKnowledge.InternetDevelopment}$

– $T_3 = \exists \mathtt{advancedKnowledge} \sqcap \forall \mathtt{advancedKnowledge}.$ $(\mathtt{TotalQualityManagement} \sqcap \mathtt{C{+}{+}}) \sqcap \exists \mathtt{hasMasterDegree} \sqcap \exists \mathtt{hasExperience}$ $\sqcap \forall \mathtt{hasExperience}.((\geq\ 3\ \mathtt{years}))$

and $\mathcal{P}$ be composed by the following knowledge provider descriptions:

$\texttt{ComputerScienceSkill} \sqsubseteq \texttt{Skill}$
$\texttt{InternetUse} \sqsubseteq \texttt{ComputerScienceSkill}$
$\texttt{OOP} \sqsubseteq \texttt{ComputerScienceSkill}$
$\texttt{C++} \sqsubseteq \texttt{OOP}$
$\texttt{HTML} \sqsubseteq \texttt{MarkupLanguages}$
$\texttt{MarkupLanguages} \sqsubseteq \texttt{ComputerScienceSkill}$
$\texttt{ClientServerProtocol} \sqsubseteq \texttt{ComputerScienceSkill}$
$\texttt{InternetDevelopment} \sqsubseteq \texttt{ComputerScienceSkill}$
$\texttt{WebDesigning} \sqsubseteq \texttt{InternetDevelopment}$
$\texttt{TotalQualityManagement} \sqsubseteq \texttt{Skill}$
$\texttt{VBScript} \sqsubseteq \texttt{ComputerScienceSkill}$
$\texttt{Design} \sqsubseteq \texttt{Skill}$
$\texttt{ComputerGraphics} \sqsubseteq \texttt{ComputerScienceSkill}$
$\texttt{Engineering} \sqsubseteq \texttt{degree}$

$\texttt{Engineer} \equiv \exists\texttt{hasMasterDegree} \sqcap \forall\texttt{hasMasterDegree.Engineering} \sqcap \exists\texttt{advancedKnowledge} \sqcap \forall\texttt{advancedKnowledge.Design} \sqcap \exists\texttt{basicKnowledge} \sqcap \forall\texttt{basicKnowledge.}(\texttt{Mathematics} \sqcap \texttt{Physics})$

$\texttt{ManagerialEngineer} \equiv \texttt{Engineer} \sqcap \exists\texttt{advancedKnowledge} \sqcap \forall\texttt{advancedKnowledge.ProcessManagement}$

Figure 1: Skills ontology excerpt

- $P_1 = \exists\texttt{basicKnowledge} \sqcap \forall\texttt{basicKnowledge.}(\texttt{InternetUse} \sqcap \texttt{ComputerGraphics} \sqcap \texttt{HTML}) \sqcap \exists\texttt{hasMasterDegree} \sqcap \exists\texttt{hasExperience} \sqcap \forall\texttt{hasExperience.}((\leq\ 2\ \texttt{years}))$
- $P_2 = \texttt{ManagerialEngineer} \sqcap \exists\texttt{advancedKnowledge} \sqcap \forall\texttt{advancedKnowledge.InternetTechnologies} \sqcap \exists\texttt{toolsKnowledge} \sqcap \forall\texttt{toolsKnowledge.WebDesigning}$
- $P_3 = \texttt{Engineer} \sqcap \exists\texttt{advancedKnowledge} \sqcap \forall\texttt{advancedKnowledge.}(\texttt{TotalQualityManagement} \sqcap \texttt{ClientServerProtocol} \sqcap \texttt{VBScript} \sqcap \texttt{OOP})$

The algorithm executes the loop in rows 5–15 nine times in order to compute the nine values for u_{ij} with $i = j = 1, 2, 3$. Such values fill the suitability matrix shown in Figure 2. The call to $solveKhun$ in row 16 returns the following values for the set of decision variables: $x_{11} = 1, x_{22} = 1, x_{33} = 1$. The loop in rows 17–21 finally returns the following three quadruples corresponding to the the values of i and j such that $x_{ij} = 1$ in the assignment solution:

	T_1	T_2	T_3
P_1	0	1	0.86
P_2	0.6	0.11	0.56
P_3	0.6	0.56	0.33

Figure 2: Suitability matrix

- $x_{11} = 1$ $(T_1, P_1, H_{11} = \top, G_{11} = \top)$
  P_1 represent a full match for T_1: the result H_{11} of the Concept Abduction Problem between T_1 and P_1 shows that nothing has to be hypothesized in P_1 to perform T_1: even knowledge about `MarkupLanguages`, apparently lacking, is implied by knowledge about `HTML`(in the ontology is in fact modeled that `MarkupLanguages` knowledge subsumes `HTML` knowledge); the result G_{11} of the Concept Contraction Problem between T_1 and P_1 shows instead that T_1 does not need any contraction to gain compatibility with P_1;

- $x_{22} = 1$ $(T_2, P_2, H_{22} = \forall\mathtt{advancedKnowledge.ClientServerProtocol}, G_{22} = \top)$
  P_2 is compatible with T_2: the result H_{22} of the Concept Abduction Problem between T_2 and P_2 shows that hypotheses have to be formulated on the advanced knowledge about Client Server Protocol while knowledge about `InternetDevelopment` is embedded into knowledge about `WebDesigning` (in the ontology is in fact modeled that `InternetDevelopment` knowledge subsumes `WebDesigning` knowledge);the result G_{22} of the Concept Contraction Problem between T_2 and P_2 shows instead that T_2 does not need any contraction to gain compatibility with P_2;

- $x_{33} = 1$ $(T_3, P_3, H_{33} = \exists\mathtt{hasExperience} \sqcap \forall\mathtt{hasExperience.}(\geq\ 3\ \mathtt{years})\sqcap$
  $\forall\mathtt{advancedKnowledge.C{+}{+}}, G_{33} = \top)$
  P_3 is compatible with T_3: the result H_{33} of the Concept Abduction Problem between T_3 and P_3 shows that hypotheses have to be formulated on the work experience requirements and on advanced knowledge about `C++`: even if P_3 knows about `OOP`, her knowledge about `C++` can only be hypothesized because the second one is more specific than the first one(in the ontology is in fact modeled that `OOP` knowledge subsumes `C++` knowledge and not vice versa); the result G_{33} of the Concept Contraction Problem between T_3 and P_3 shows instead that T_3 does not need any contraction to gain compatibility with P_3;

5.2 System Operating mode

The approaches outlined in the previous sections for *single one to one*, *multiple one to one* and *many to one* matching were implemented and tested with the aid of a system whose operating mode is sketched in Figure 3.

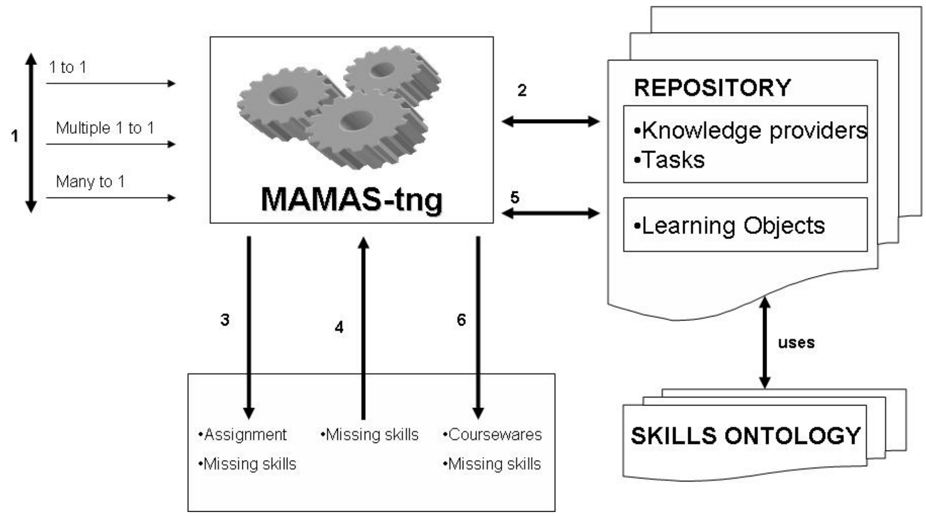

Figure 3: System Operating Mode

1. In the initial step, the user chooses the matching scenario (arrow 1) and models, using a GUI as the one shown in Figure 4, the task to be satisfied by the knowledge provider profiles, available within the system. All the required tasks and the knowledge provider profiles have to be formalized using the terminology of the skills ontology, regardless of the chosen matching scenario. Notice that we use MaMaS-tng[4] as reasoning engine. To the best of our knowledge, MaMaS-tng is currently the only reasoning engine able to solve Concept Abduction and Concept Contraction problems. Figure 4 shows the formalization of a needed task w.r.t. the ontology within the system GUI. In an analogous way, figure 5 shows the formalization of a knowledge provider profile.

2. Exploiting nonmonotonic services exposed by MaMaS-tng, the system is able to perform the matching process selected in the previous step. Given the knowledge providers profiles stored within the repository (arrow 2), the system is able to return (arrow 3):
   - the knowledge provider profile P_i best matching the request for a single task, together with explanations H_i ang G_i for non full match, according

[4] **MA**tch**MA**ker**S**ervice is available at http://dee227.poliba.it:8080/MAMAS-tng.

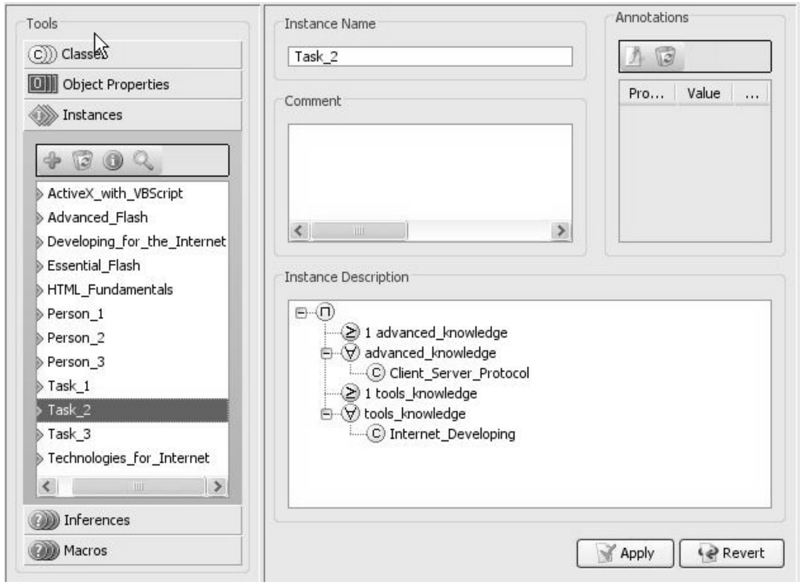

Figure 4: Task Description

to the approach described in Section 4.1 — in case of *single one to one* matching

- a set of quadruples $\langle T_i, P_j, H_{ij}, G_{ij}\rangle$ whose elements represent the different task/profile assignments (T_i, P_j), together with explanations for non full matches (H_{ij}, G_{ij}), according to the approach presented in Section 4.2 — in case of *multiple one to one* matching
- a set of knowledge provider profiles *i.e.*, a team, whose conjunction covers the required task, a concept $T_{uncovered}$ representing the part remaining uncovered of the task, and the part to contract in the request, $G_{contraction}$, as presented in Section 4.3 — in case of *many to one* matching

3. In all the above cases, non full match explanations $H_i, H_{ij}, T_{uncovered}$ represent the learning needs at the basis of an automated composite courseware composition. In this step, the system follows the approach detailed in Section 4.4: it takes as inputs the learning needs (arrow 4) and the learning objects descriptions (see Figure 6) stored within the repository (arrow 5). By calling MaMaS-tng, this module produces as output a set of candidate composite coursewares and their relative $\rho_{D_{uncovered}}$ (arrow 6).

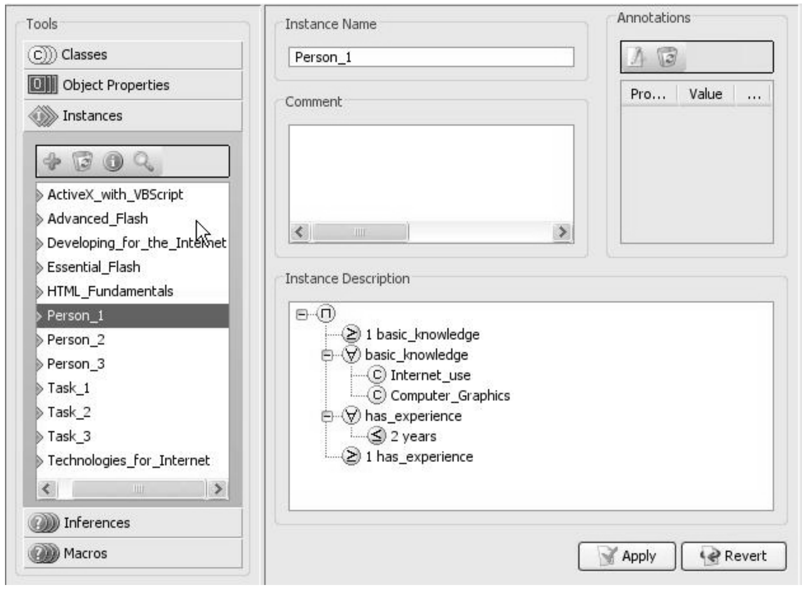

Figure 5: Knwowledge Provider Description

6 Related Work

Skill management systems presented in the 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 stemmed terms, typical of text-based Information Retrieval, have been used to evaluate possible matches [Veit et al., 2001]. 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 vertices includes assignees and the second one includes tasks to be performed [Saip and Lucchesi, 1993]. 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 [Kennington and Wang, 1991, Galil, 1986, Hillier and Lieberman, 1995].

Among proposal on the subject, in [Sure et al., 2000] 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

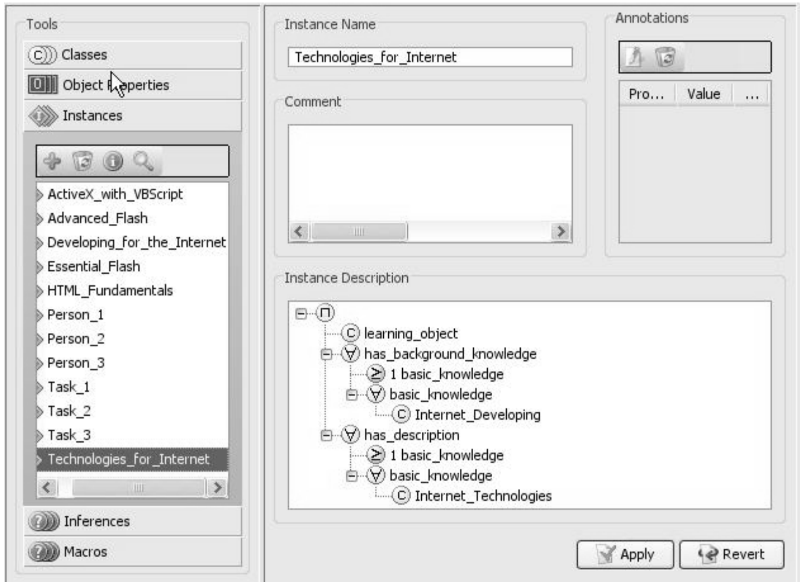

Figure 6: Learning Object Description

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. But both systems lack of an ontology as skill repository, allowing to infer on previously introduced profiles.

In [Becerra-Fernandez, 2000] two People Finder Knowledge Management Systems are proposed: the Searchable Answer Generating Environment(SAGE) and the Expert Seeker. Both systems use databases as skill repositories and query engines performing a keyword search for expertise, even if the second one provides more search options. Even though 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 [Garro and Palopoli, 2003] it is proposed an XML multi-agent system providing, among many other facilities, support to management in searching the most suitable employee for a specific job.

In [Sugawara, 2003] an agent-based application for supporting job matchmaking is proposed, focusing on the telework scenario.

The use of ontologies as knowledge repositories has been largely recognized useful to provide a common vocabulary and to use inference services on elicited knowledge ([OLeary, 1998a], [OLeary, 1998b]).

A general purpose ontology to model Knowledge Management procedures has also been proposed in [Holsapple and Joshi, 2004]. Also skill management systems have then to employ ontologies as skill repositories in order to achieve such goals.

In [Lau and Sure, 2002] an ontology based skill management system is proposed, allowing employees to elicit their skills and providing an advanced expert search within the intranet.

In [Hefke and Stojanovic, 2004] a semantic based portal is proposed. The portal answers users queries about tasks to perform by providing ad-hoc organizational teams. The user request is formalized as a query searching the competences required for the task in the ontology used as skill 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 people.

In [Liu and Dew, 2004] 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.

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.

In [Colucci et al., 2003b] a semantic based approach to the problem of skills finding in an ontology supported framework was presented. The framework considers skill management as an electronic marketplace of knowledge in which skills are a peculiar kind of goods that have distinguishing characteristics with respect to traditional assets; buyers are entities that need the skills of people, such as projects, departments and organizations. On the other hand, knowledge sellers are individuals that offer their own skills. Obviously, descriptions of profiles share a common skills ontology.

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

The approach proposed in [Colucci et al., 2003b] is oriented to finding the best individual for a given task or project, based on profile descriptions sharing a common ontology. The approach is able to cope with cases in which no perfect matches exist, *i.e.*, finding those available profiles that, for a given skill request best match, also if not identical, the task and vice versa. It is noteworthy that the approach allows not only a logical categorization, but also a ranking of matches within each category. In [Colucci et al., 2004] an approach to endow with semantics the process of searching solutions to task assignment was also presented.

The Assignment Problem[Hillier and Lieberman, 1995] 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 used for weighting arcs 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.

The proposed solutions to skill matching presented so far are all relative to the case of *one to one* multiplicity. When instead an ad-hoc team has to be created for performing a task we revert in the case of many to one multiplicity.

In [Colucci et al., 2005b] the process of team composition is carried out by solving an extended Concept Covering Problem. The possible assignees represent the set of people to be used to cover as much as possible the skills requested to perform the task that starts the matching process. The presented approach takes into account also explanation about skills not covered by the team, although requested by the task. In [Colucci et al., 2007a]such an approach was extended to take belief revision into account.

In [Hefke and Stojanovic, 2004] an alternative approach, which may appear close to that proposed in [Colucci et al., 2003b] is presented. The approach builds on the technique presented in [Stojanovic et al., 2003] 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.

All the systems and approaches so far outlined deal with the search for skills among the available assignees. Creating new competencies when the available ones are not enough to perform all the needed tasks may represent a competitive advantage opportunity. In order to achieve such knowledge creation, SMS may integrate components supporting the training process of employees, exploiting e-learning technologies. The term e-learning has become common, describing several concepts, from complete web-based courses to distance learning and tutoring. Recently, also thanks to various standardization efforts [IEEE, 2003], emphasis has been placed on the concept of learning object *i.e.*, small and easily reusable educational resources to be composed to allow personalized instruction and courseware creation [Ip et al., 2002, Cabezuelo and Beardo, 2004, Ajami, 2004, Vossen and Jaeschke, 2003].

Obviously, discovery and composition of such learning objects in an automated way requires the association of unambiguous and semantically rich metadata, defined in accordance with shared ontologies ([Sicilia and Lytras, 2005], [Sanchez-Alonso and Frosch-Wilke, 2005]). The LOM –Learning Object Metadata [IEEE, 2002]– standard, though limited in the basic annotation items, allows to freely define annotated metadata describing a learning resource.

The semantic-based annotation of educational resources is hence fully in the stream of the Semantic Web initiative [Berners-Lee et al., 2001], and it can share with it both techniques and approaches [Sanchez-Alonso and Sicilia, 2004, Bennacer et al., 2004, Gasevic et al., 2004].

In particular, as more and more learning objects become available on the Web as services with well-defined machine interpretable interfaces as described *e.g.*, in OWL-S [OWL-S, 2004, Sycara et al., 2003], personalized learning units can be built by scratch, by retrieving learning resources. Automated composition of learning resources, exposed as web services for example, can then match a personalized learning need.

Recent studies [Draganidis and Mentzas, 2006] underline the rarity of approaches integrating knowledge and learning management.

In [Colucci et al., 2007a] a SMS integrating both a skill management and an automatic courseware composition component is proposed. The system takes into account both explanation and belief revision in the skill matching component in many to one and one to one multiplicity case. The missing skills resulting from the matching process are given as input for a courseware composition process which takes into account also explanation by employing an algorithm presented in [Colucci et al., 2005c].

The semantic-based integration of competences with learning needs is also tackled *e.g.*, in [Draganidis et al., 2006]. Nevertheless, relying on standard services and on RDQL, such systems cannot deal with approximation nor provide explanation services.

One attempt to integration mostly focused on knowledge modeling is proposed in [Sicilia et al., 2006], which present a case study for modeling e-learning procedures in the general purpose ontology for knowledge management proposed in [Holsapple and Joshi, 2004].

7 Conclusions

Both knowledge management systems and e-learning solutions provide a significant support to strategical human resource management. Nevertheless their roles are often kept separated, and the processes of managing skills and planning training programs are scheduled independently on each other. The SMS proposed in this paper shows instead as the integration of knowledge and learning management may represent a promising synergy in organizational vision. The ability to assign "the right man to the right job" is universally shared and recognized by companies as a crucial success factor, to invest on. On the contrary, the impact of e-learning solutions on return on investments may appear less straightforward and training programs are often considered low priority investments.

In this paper we contextualize the learning need to the solution of an assignment problem, proposing training programs targeted at covering a task rather than at increasing the employees background knowledge. Of course the proposed coursewares, when fruited, enrich organizational knowledge and open new business chances to the company, but we believe that the explicit connection of training programs to the solution of a needed task more effectively stimulates companies in investing on learning solutions.

Future research will be devoted to evaluate the correspondence of the proposed system with human judgment in the processes of assignees and learning paths selection and to improve system usability by common users. The prototype system described in this paper has been the basis for a novel and optimized commercial system for skills managment, *Impakt*, which will be released next year by D.O.O.M.srl.

Acknowledgments

We acknowledge partial support of Apulia Region project PE_043 "ACAB-C2: System for Automated Composition of business processes in ERPs using Abduction-Based Concept Covering".

References

[Ajami, 2004] Ajami, K. (2004). Specifying and implementing interoperable and reusable learning objects: one step beyond. In *Proc. of Intl. Conf. on Information and Communication Technologies: From Theory to Applications*, pages 111–112.

[Baader et al., 2002] Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D., and Patel-Schneider, P., editors (2002). *The Description Logic Handbook*. Cambridge University Press.

[Baader et al., 2003] Baader, F., Horrocks, I., and Sattler, U. (2003). Description logics as ontology languages for the semantic web. In Hutter, D. and Stephan, W., editors, *Festschrift in honor of Jörg Siekmann*, Lecture Notes in Artificial Intelligence. SV.

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

[Bennacer et al., 2004] Bennacer, N., Bourda, Y., and Doan, B. (2004). Formalizing for querying learning objects using OWL. In *Proc. of Intl. Conf. on Advanced Learning Technologies*, pages 321–325. IEEE.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 248(4):34–43.

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

[Cabezuelo and Beardo, 2004] Cabezuelo, A. and Beardo, J. (2004). Towards a model of quality for learning objects. In *Proc. of Intl. Conf. on Advanced Learning Technologies*, pages 822–825. IEEE.

[Colucci et al., 2003a] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., and Mongiello, M. (2003a). Concept Abduction and Contraction in Description Logics. In *Proc. of the 16th Intl. Workshop on Description Logics (DL'03)*, volume 81 of *CEUR Workshop Proceedings*.

[Colucci et al., 2005a] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., and Mongiello, M. (2005a). Concept Abduction and Contraction for Semantic-based Discovery of Matches and Negotiation Spaces in an E-Marketplace. *Electronic Commerce Research and Applications*, 4(4):345–361.

[Colucci et al., 2003b] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., Mongiello, M., and Mottola, M. (2003b). A formal approach to ontology-based semantic match of skills descriptions. *Journal of Universal Computer Science (J.UCS)*, 9(12):1437–1454.

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

[Colucci et al., 2005b] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., and Ragone, A. (2005b). Automated task-oriented team composition using description logics. In *5th International Conference on Knowledge Management I-Know 05*, pages 229–236. Springer.

[Colucci et al., 2005c] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., and Ragone, A. (2005c). Semantic-based automated composition of distributed learning objects for personalized e-learning. In *The Semantic Web: Research and Applications. 2nd European Semantic Web Conf.*, volume 3532 of *LNCS*, pages 633–648. SV.

[Colucci et al., 2007a] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., and Ragone, A. (2007a). Integrated semantic-based composition of skills and learning needs in knowledge-intensive organizations. In *Competencies in Organizational E-Learning:Concepts and Tools*, pages 270–298. Idea Group Publishing.

[Colucci et al., 2007b] Colucci, S., Di Noia, T., Pinto, A., Ragone, A., Ruta, M., and Tinelli, E. (2007b). A non-monotonic approach to semantic matchmaking and request refinement in e-marketplaces. *International Journal of Electronic Commerce*, 12(2).

[Cormen et al., 1990] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (1990). *Introduction to Algorithms*. The Massachusetts Institute of Technology.

[Di Noia et al., 2004a] Di Noia, T., Di Sciascio, E., and Donini, F. (2004a). Extending Semantic-Based Matchmaking via Concept Abduction and Contraction. In *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge*

Management (EKAW 2004), LNAI. Springer.

[Di Noia et al., 2007] Di Noia, T., Di Sciascio, E., and Donini, F. (2007). Semantic Matchmaking as Non-Monotonic Reasoning: A Description Logic Approach. *Journal of Artificial Intelligence Research (JAIR)*, 29:269–307.

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

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

[Draganidis et al., 2006] Draganidis, F., Chamopoulou, P., and Mentzas, G. (2006). An ontology based tool for competency management and learning paths. In *Proc. of I-KNOW '06*.

[Draganidis and Mentzas, 2006] Draganidis, F. and Mentzas, G. (2006). Competency based management: A review of systems and approaches. *Information Management and Computer Security*, 14(1):51–64.

[DublinCore, 1999] DublinCore (1999). Dublin core metadata element set, version 1.1: Reference description. http://dublincore.org/documents/1999/07/02/dces/.

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

[Gärdenfors, 1988] Gärdenfors, P. (1988). *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. Bradford Books, MIT Press, Cambridge, MA.

[Garro and Palopoli, 2003] Garro, A. and Palopoli, L. (2003). 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.

[Gasevic et al., 2004] Gasevic, D., Jovanovic, J., and Devedzic, V. (2004). Enhancing learning object content on the semantic web. In *Proc. of Intl. Conf. on Advanced Learning Technologies*, pages 714–716. IEEE.

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

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

[Hefke and Stojanovic, 2004] Hefke, M. and Stojanovic, L. (2004). 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.

[Hillier and Lieberman, 1995] Hillier, F. S. and Lieberman, G. J. (1995). *Introduction to Operations Research*. MGH.

[Holsapple and Joshi, 2004] Holsapple, C. W. and Joshi, K. D. (2004). A formal knowledge management ontology: Conduct, activities, resources, and influences: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 55(7):593–612.

[IEEE, 2002] IEEE (2002). *IEEE Standard for Learning Object Metadata*, std 1484.12.1-2002 edition.

[IEEE, 2003] IEEE (2003). *IEEE standard for learning technology-learning technology systems architecture (LTSA)*, std 1484.1-2003 edition.

[IEEE, 2007] IEEE (2007). *Draft standard for learning technology- Data Model for Reusable Competency Definition*, 1484.20.1/draft 7 edition. "http://www.ieeeltsc.org/working-groups/wg20Comp/wg20rcdfolder/IEEE_1484.20.1.D7.pdf".

[IMS, 2001] IMS (2001). Ims, learning resource meta-data best practices and implementation guide version 1.1 - final specification. http://www.imsproject.org/metadata/mdbestv1p1.html.

[IMS, 2002] IMS (2002). Ims reusable definition of competency or educational objective- information model.

http://www.imsglobal.org/competencies/rdceov1p0/imsrdceo_infov1p0.html.

[Ip et al., 2002] Ip, A., Young, A., and Morrison, I. (2002). Learning objects - whose are they? In *Proc. of 15 th Conf. of the National Advisory Committee on Computing Qualifications*, pages 315–320.

[Kennington and Wang, 1991] Kennington, J. and Wang, Z. (1991). An empirical analysis of the dense assignment problem: Sequential and parallel implementations. *ORSA Journal on Computing*, 3(4):299–306.

[Kuhn, 1955] Kuhn, H. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

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

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

[OLeary, 1998a] OLeary, D. (1998a). Enterprise knowledge management. *IEEE Computer*, 31(3):54–61.

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

[OWL, 2004] OWL (2004). www.w3.org/TR/owl-features/.

[OWL-S, 2004] OWL-S (2004). The owl services coalition. www.daml.org/services/owl-s/1.0/owl-s.html.

[Saip and Lucchesi, 1993] Saip, H. B. and Lucchesi, C. (1993). Matching algorithms for bipartite graph. Technical Report DCC -03/93. Departamento de Cincia da Computao, Universidade Estudal de Campinas.

[Sanchez-Alonso and Frosch-Wilke, 2005] Sanchez-Alonso, S. and Frosch-Wilke, D. (2005). On the semantics of aggregation and generalization in learning object contracts. In *Proc. of Intl. Conf. on Advanced Learning Technologies*, pages 425–429. IEEE.

[Sanchez-Alonso and Sicilia, 2004] Sanchez-Alonso, S. and Sicilia, M. (2004). An ontological representation of learning objects and learning designs as codified knowledge. *The Learning Organization: An International journal*, 12(5):471–479.

[SCORM, 2004] SCORM (2004). Advanced distributed systems (adl) lab, sharable content object reference model (scorm). http://www.adlnet.org/index.cfm?fuseaction=scormabt.

[Sicilia and Lytras, 2005] Sicilia, M. and Lytras, M. (2005). The semantic learning organization. *The learning organization: An international journal*, 12(5):402–410.

[Sicilia et al., 2006] Sicilia, M. A., Lytras, M., Rodriguez, E., and Garcia-Barriocanal, E. (2006). Integrating descriptions of knowledge management learning activities into large ontological structures: a case study. *Data Knowledge Engineering*, 57(2):111–121.

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

[Stojanovic et al., 2003] Stojanovic, N., Studer, R., and Stojanovic, L. (2003). An approach for the ranking of query results in the semantic web. In *Proc. of International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*, pages 500–516.

[Sugawara, 2003] Sugawara, K. (2003). Agent-based application for supporting job matchmaking for teleworkers. In Patel, D., Patel, S., and Wang, Y., editors, *Proc. of the 2nd IEEE Intl. Conference on Cognitive Informatics (ICCI 2003)*, pages 137–142.

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

[Sycara et al., 2003] Sycara, K., Paolucci, M., Ankolekar, A., and Srinivasan, N. (2003). Automated discovery, interaction and composition of semantic web services. *Journal of Web Semantics*, 1(1):27–46.

[Trastour et al., 2002] Trastour, D., Bartolini, C., and Priest, C. (2002). Semantic Web Support for the Business-to-Business E-Commerce Lifecycle. In *Proc. International World Wide Web Conference (WWW) '02*, pages 89–98. ACM.

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

[Vossen and Jaeschke, 2003] Vossen, G. and Jaeschke, P. (2003). Learning objects as a uniform foundation for e-learning platforms. In *Proc. of Intl. Symp. on Database Engineering and Applications Symposium*, pages 278–287.

A The $\mathcal{ALN}$ subset of OWL-DL

Description Logics (DLs) [Baader et al., 2002] are a family of logic formalisms for knowledge representation whose basic syntax elements are *concept* names and *role* names. Concepts stand for sets of objects, *e.g.*, `ProcessEngineer`, `Graduate`, `BusinessApplication`, while roles, *e.g.*, `hasAbility`, `specialized`, link objects in different concepts. 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 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.*, $\mathtt{Graduate} \sqcap \exists \mathtt{hasAbility.NegotiationSkills}$, which describes the set of graduated people with negotiation skills, and *universal role quantification* ($\forall$), *e.g.*, $\mathtt{Programmer} \sqcap \forall \mathtt{hasMasterDegree.Engineering}$, which describes programmers having only an engineering degree. Other constructs may involve counting, as number restrictions:

- $\mathtt{Graduate} \sqcap (\leq\ 3\ \mathtt{hasAbility})$ expresses graduates having at least three abilities
- $\mathtt{AccountManager} \sqcap (\geq\ 2\ \mathtt{hasTechnicalSkills})$ describes account managers endowed of at least two skills belonging to the technical area

The representation of knowledge is achieved in DLs formalism by using concepts expressions to structure *inclusion assertions* and *definitions*. For example we could impose that programming may be partitioned into structural and object oriented using the two inclusions $\mathtt{Programming} \sqsubseteq \mathtt{StructuralProgramming} \sqcup \mathtt{ObjectProgramming}$ and $\mathtt{StructuralProgramming} \sqsubseteq \neg \mathtt{ObjectProgramming}$.

We can state also that working teams have to be composed by at least two members as $\mathtt{Team} \sqsubseteq (\geq\ 2\ \mathtt{hasTeamMember})$. Historically, sets of such inclusions are called TBoxes (Terminological Box).

It is easy to see that $\mathcal{T}$ in DLs represents what is called an ontology in a knowledge representation system. The DL we adopt in our system is $\mathcal{ALN}$(**A**ttributive **L**anguage with unqualified **N**umber restrictions) DL. The choice of such a DL

OWL syntax	**DL syntax**
$< owl : Thing/ >$	$\top$
$< owl : Nothing/ >$	$\bot$
$< owl : Classrdf : ID = "C"/ >$	C
$< owl : ObjectPropertyrdf : ID = "R"/ >$	R
$< rdfs : subClassOf/ >$	$\sqsubseteq$
$< owl : equivalentClass/ >$	$\equiv$
$< owl : disjointWith/ >$	$\neg$
$< owl : intersectionOf/ >$	$\sqcap$
$< owl : allValuesFrom/ >$	$\forall$
$< owl : someValuesFrom/ >$	$\exists$
$< owl : maxCardinality/ >$	$\leq$
$< owl : minCardinality/ >$	$\geq$
$< owl : cardinality/ >$	$=$

Table 1: Correspondence between OWL-DL and $\mathcal{ALN}$ DL syntax

is due to a trade off between language expressiveness and computational complexity of inference services[Borgida and Patel-Schneider, 1994].

Constructs allowed in an $\mathcal{ALN}$ DL are:

- $\top$ *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 in 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.
- $\exists R$ *unqualified existential restriction.* There exists at least one object participating in the relation R.
- $(\geq\ n\ R)$,$(\leq\ n\ R)$,$(=\ n\ R)$. Respectively the minimum, the maximum and the exact number of objects participating in the relation R.

We use a *simple-TBox* in order to express the relations among objects in the domain. With a *simple-TBox*, in all the axioms (for both inclusion and definition) the left side is represented by a concept name. The subset of OWL-DL Tags allowing to express an $\mathcal{ALN}$ DL is presented in Table 1.