Proceedings of I-KNOW '08 and I-MEDIA '08
Graz, Austria, September 3-5, 2008

A Knowledge-based Solution for Core Competence Evaluation in Human-Capital Intensive Companies

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

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

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

Abstract: Determining fields of excellence in the know-how of knowledge intensive companies is often a crucial decisional process, aimed e.g., at identifying the competence to be strengthened or to invest on in a long term strategy. In this paper we propose a semantic-based approach for automatic extraction of such a specializing knowledge, usually called Core Competence in knowledge management literature. The proposed approach exploits Description Logics as formalism for the representation of knowledge sources and implements novel reasoning services, in particular *informative common subsumers* specifically devised for Core Competence evaluation.

Key Words: Common Subsumers, Core Competence, Description Logics.
Category: M.4, M.7, H.3.3

1 Introduction

Since first investigations on the role of knowledge in human-capital intensive companies, the capability to focus on a significant portion of the organizational know-how has been identified as a crucial asset for business success. Such a belief is at the basis of the so-called *Resource-based Theory* of the firm [Wernerfelt, 1995], according to which unique company capabilities should be exploited to achieve competitive advantage [Barney, 1991, L. Halawi and McCarthy, 2005, Meso and Smith, 2000]. In particular, the term Core Competence was introduced [Hamel and Prahalad, 1990] to denote such a specializing portion of organizational know-how. It is intuitive that the hardness of identifying such an intellectual capital increases with the size of the company and with the ambiguity of company know-how descriptions.

In recent years we have been investigating knowledge-based approaches and solutions for a specific field of knowledge management, namely skills and competence management, in the framework of Description Logics (DLs)[Baader et al., 2002] and semantic technologies, both exploiting classical inference services and introducing new ones [Colucci et al., 2007b]. As it is nowadays well-known, semantic-based technologies ask for company intellectual capital to be unambiguously described in formal representations, according to a shared vocabulary provided by ontologies modeling skills domain. In particular, our solutions employ DLs for knowledge representation and exploit DL reasoning services to infer new knowledge on the elicited descriptions. Obviously, once company know-how has been formally represented in a common knowledge base in terms of individual profile descriptions and know-how, such a repository could be exploited to extract the most characterizing portion of company intellectual capital, *i.e.* company Core Competence. Nevertheless this is easier said than done, as well-known reasoning services fail to provide such information. As we show later on, the apparently best suited inference service for the above task, the *Least Common Subsumer* [Cohen et al., 1992], shows clear limits.

In this paper we therefore propose new non-standard inference services on collections of individual profile descriptions formalized in DL, for the automatic extraction of company Core Competence. Such specifically developed reasoning services are introduced in Section 2. Two different Core Competence evaluation approaches are then detailed in Section 3, before closing the paper with conclusions.

2 New Services Definition

In the automated Core Competence extraction we propose, we refer to $\mathcal{ALN}$ (Attributive Language with Number Restrictions) for formally describing knowledge sources of a company. $\mathcal{ALN}$ provides a limited set of constructs, which allow for describing the knowledge domain by combining the basic elements of a DL, namely **concept names**, representing objects of the domain — *i.e.* $ProductionManagement$, $AssetAllocation, Creativeness$ and $AssetManager$ — and **role names**, representing possible binary relationships among concepts, *i.e.* $knows, isAbleTo$. Every DL includes two concepts, $\top$ and $\bot$, representing a concept interpreted by the whole domain and by an empty set, respectively. $\mathcal{ALN}$ allows also for **qualified universal restrictions** — *i.e.* $\forall knows.AssetAllocation$ denotes an advanced knowledge in Asset Allocation — and **number restrictions** — *i.e.* $(\geq\ 3\ knows)$, $(\leq\ 2\ isAbleTo)$ denote the possession of at least three skills and at most two abilities — over roles. By using such constructs it is possible to detail concept **inclusions** and **definitions**, which constitute the intensional knowledge of a DL system, what is called a **TBox** in DL and **ontology** in knowledge representation. For example the inclusion $AssetManager \sqsubseteq Manager$ asserts that the set of asset managers in the domain is included in the one of managers; the concept definition $AssetManager \equiv Manager \sqcap \forall knows.AssetAllocation$ gives instead to managers endowed with Asset Allocation knowledge the name of Asset Manager, like the definition $Manager \equiv \sqcap \forall knows.Management$ gives the name manager to subsets of domain possessing Management knowledge.

Every DL allows for basic reasoning services inferring new knowledge from the descriptions elicited in the TBox; in particular *satisfiability* and *subsumption* are defined for every DL. In a nutshell, satisfiability checks for internal coherency of concept descriptions, evaluating the consistency of elicited information; subsumption checks instead whether a concept description is more generic than another one. Formally, subsumption is defined as follows, with respect to a domain interpretation function $\mathcal{I}$:

Definition 1 (Subsumption) Given two concept descriptions C and D and a TBox $\mathcal{T}$ in a DL $\mathcal{L}$, we say that D subsumes C w.r.t. $\mathcal{T}$ if for every model of $\mathcal{T}$, $C^{\mathcal{I}} \subset D^{\mathcal{I}}$. We write $C \sqsubseteq_{\mathcal{T}} D$, or simply $C \sqsubseteq D$ if we assume an empty TBox.

Having a collection of concept descriptions in a DL $\mathcal{L}$, the problem of determining the Least Common Subsumer(LCS) of the collection has been proposed by Cohen, Borgida and Hirsh [Cohen et al., 1992] as a non-standard reasoning service. By definition, the LCS of a collection of concept descriptions represents the most specific concept description subsuming all of the elements of the collection. Formally, we recall the following definition:

Definition 2 (LCS,[Cohen and Hirsh, 1994]) Let $C_1, \ldots, C_n$ be n concepts in a DL $\mathcal{L}$. An $LCS(C_1, \ldots, C_n)$, is a concept E in $\mathcal{L}$ such that the following conditions hold:
(i) $C_i \sqsubseteq E$ for $i = 1, \ldots, n$
(ii)E is the least $\mathcal{L}$-concept satisfying (i),*i.e.,* , if E' is an $\mathcal{L}$-concept satisfying $C_i \sqsubseteq E'$ for all $i = 1, \ldots, n$, then $E \sqsubseteq E'$

If the collection contains employee profile descriptions, as in our reference scenario, the LCS represents the competence shared by all the employees in the collection. Such a concept description is a good candidate for determining the Core Competence of the company at a first sight. Nevertheless the need for the LCS to subsume each concept in the collection causes its corresponding description to be too generic in most cases: if a competence has to be shared by the whole company personnel it needs to be quite generic. As a toy example, consider a small company in which only the following three employees work:

– *Nick*: $AssetManager \sqcap \forall isAbleTo.Creativeness$
– *Frank*: $\forall knows.AssetAllocation \sqcap \forall isAbleTo.Creativeness$
– *Robert*: $Engineer \sqcap \forall isAbleTo.Creativeness$

The only LCS of such a collection is Creativeness ability, which might result a not much significant knowledge. If we instead give up such a full skill coverage and accept the assumption that Core Competence needs to be possessed by a significant portion of company personnel, more interesting results can be achieved. Obviously the required degree of coverage may be set by company management. To this aim, we propose and introduce new reasoning services.

Definition 3 (k-CS) Let $C_1, \ldots, C_n$ be n concepts in a DL $\mathcal{L}$, and let be $k < n$. A k-*Common Subsumer* (k-CS) of $C_1, \ldots, C_n$ is a concept D such that D is an LCS of k concepts among $C_1, \ldots, C_n$.

If the example company management decides that $2/3$ of the employees have to possess some knowledge to consider it part of the Core Competence, Asset Allocation knowledge represents a commonality between two employees (according to the definitions at the beginning of the section) and then a Core Competence. Of course also Creativeness ability is a $k - CS$ of the collection, but it does not add any informative content to the LCS: for this reason we distinguish in the following k-Common Subsumers adding informative content to LCS.

Definition 4 (IkCS) Let $C_1, \ldots, C_n$ be n concepts in a DL $\mathcal{L}$, and let $k < n$. An *Informative k-Common Subsumer* (IkCS) of $C_1, \ldots, C_n$ is a k-CS E such that E is strictly subsumed by $LCS(C_1, \ldots, C_n)$.

Among possible $IkCSs$, some are characterized by maximal cardinality of the set of subsumed concepts: in our example scenario, if we set $k = 3$ Asset Allocation knowledge stops being a $k-CS$ and the only common subsumer is Creativeness ability, which is not informative by definition. We define in the following concepts like Asset Allocation as best informative common subsumers (with $k = 2$).

Definition 5 (BICS) Let $C_1, \ldots, C_n$ be n concepts in a DL $\mathcal{L}$. A *Best Informative Common Subsumer* (BICS) of $C_1, \ldots, C_n$ is a concept B such that B is an Informative k-CS for $C_1, \ldots, C_n$, and for every $k < j \leq n$ every j-CS is not informative.

For collections whose LCS is equivalent to the universal concept $\top$, the following definition makes also sense:

Definition 6 (BCS) Let $C_1, \ldots, C_n$ be n concepts in a DL $\mathcal{L}$. A *Best Common Subsumer* (BCS) of $C_1, \ldots, C_n$ is a concept S such that S is a k-CS for $C_1, \ldots, C_n$, and for every $k < j \leq n$ every j-CS $\equiv \top$.

Consider for example a new employee :

– *Fred* $= Manager \sqcap \forall knows.ProductionManagement.$

The only LCS of the collection including the four employees is the universal concept. On the contrary, for $k = 3$ we have Creativeness ability as k-common subsumer, which is informative w.r.t. the LCS (it is equivalent to the universal concept) and best: if we add one unit to k the k-CS reverts to the universal concept.

3 Solutions to Core Competence Evaluation Problem

In this paper we provide two processes for Core Competence evaluation: the first one exploits the services introduced in Section 2 to discover unknown fields of excellence of the company; the second one checks for the possession of a list of known target competencies by a significant portion of company personnel and explains how to reach the target in case the check fails.

Both of the approaches ask for the concepts to be written in *Concept Components* according to the following rules. If C is a concept description in a DL $\mathcal{L}$, with C written in a conjunction $C^1 \sqcap \cdots \sqcap C^m$, the *Concept Components* of C are defined as follows: if C^j, with $j = 1 \ldots, m$ is either a concept name or a negated concept name or a number restriction, then C^j is a *Concept Component* of C; if $C^j = \forall R.E$, with $j = 1 \ldots, m$,

then $\forall R.E^k$ is a *Concept Component* of of C, for each E^k concept component of E.
The definition of *Subsumers Matrix* in the following is preliminary to both processes of Core Competence evaluation.
Definition 7 (Subsumers Matrix) Let $C_1, \ldots, C_n$ be a collection of concept descriptions C_i in a Description Logic $\mathcal{L}$ and let $D_j \in \{D_1, \ldots, D_m\}$ be the *Concept Components* deriving from a set of concepts. We define the **Subsumers Matrix** $S = (s_{ij})$, with $i = 1 \ldots n$ and $j = 1 \ldots m$, such that $s_{ij} = 1$ if the component D_j subsumes C_i, and $s_{ij} = 0$ if the component D_j does not subsume C_i.
Referring to Subsumers Matrix, we define:
Concept Component Signature (sig_{D_j}): set of indeces of concepts $C_1, \ldots, C_n$ subsumed by D_j; observe that $sig_{D_j} \subseteq \{1, \ldots, n\}$;
Concept Component Cardinality (T_{D_j}): cardinality of sig_{D_j}, that is, how many concepts among $C_1, \ldots, C_n$ are subsumed by D_j. Such a number is $\sum_{i=1}^{n} s_{ij}$;
Maximum Concept Component Cardinality (M_S): maximum among all concept component cardinalities, that is, $M_S = max\{T_{D_1}, \ldots T_{D_m}\}$;
Second Maximum Concept Component Cardinality (PM_S): maximum among the cardinalities of concept components not subsuming all n concepts in the collection ($PM_S = max\{T_{D_j} | T_{D_j} < n\}$); by definition $PM_S < n$;
Common Signature Class($\bigcap_{sig_{D_j}}$): concept formed by the conjunction of all concept components whose signature contains D_j: $\sqcap\{D_h \mid sig_{D_j} \subseteq sig_{D_h}\}$

3.1 Core Competence Extraction

Definition 8 (Collection Subsumers Matrix) Let $C_1, \ldots, C_n$ be a collection of concept descriptions C_i in a Description Logic $\mathcal{L}$. We define the **Collection Subsumers Matrix** as a Subsumers Matrix in which $D_j \in \{D_1, \ldots, D_m\}$ are the concept components deriving from all concepts in the collection.
In the following we define, with respect to a collection of concept descriptions, $\underline{BCS}$ the set of BCSs, $\underline{BICS}$ the set of BICSs, $\underline{ICS_k}$ the set of IkCSs, given $k < n$ and $\underline{CS_k}$ the set of k-CSs, given $k < n$. In [Colucci et al., 2008] we proposed Algorithm 1 determining the sets $\underline{BICS}$, $\underline{CS_k}$, $\underline{ICS_k}$, $\underline{BCS}$ of a collection $\{C_1, \ldots, C_n\}$ of concepts in $\mathcal{ALN}$, whose Subsumers Matrix is given as input. In order to understand the rationale of the proposed algorithm, consider the company with the four employees (Nick, Frank, Robert and Fred) in the tiny example in Section 2. The concept components coming from the collection of employees are: $D_1 = \forall knows.Management$, $D_2 = \forall knows.AssetAllocation$, $D_3 = \forall isAbleTo.Creativeness$, $D_4 = Engineer$, $D_5 = \forall knows.ProductionManagement$. The collection subsumers matrix is shown in the left-hand side of Figure 1. If $k = 2$, the only components with cardinality at least equal to k are D_1, D_2 and D_3 and then their common class signature is added to the set $\underline{CS_k}$ (line 3), which contains the $k - CSs$ D_1, $D_2 \sqcap D_3$ and D_3. The check in line 4 results true for all of the three components, given that the only concept subsuming the four employee profiles is the universal concept; the three concepts in $\underline{CS_k}$ are then added to

Input : Collection Subsumers Matrix $S = (s_{ij})$ for a collection of concepts $\{C_1, \ldots, C_n\}$ in $\mathcal{ALN}$, integer $k < n$

Output : $\underline{CS}_k$; $\underline{ICS}_k$; $\underline{BICS}$; $\underline{BCS}$

1 $\underline{CS}_k := \emptyset$; $\underline{ICS}_k := \emptyset$; $\underline{BICS} := \emptyset$; $\underline{BCS} := \emptyset$;

2 **foreach** D_j *s.t.* $T_{D_j} \geq k$ **do**

3 $\quad \underline{CS}_k := \underline{CS}_k \cup \bigcap_{sig_{D_j}}$;

4 $\quad$ **if** $T_{D_j} < n$ **then** $\underline{ICS}_k := \underline{ICS}_k \cup \bigcap_{sig_{D_j}}$;

5 **if** $M_S = n$ **then** **foreach** D_j *s.t.* $T_{D_j} = PM_S$ **do** $\underline{BICS} := \underline{BICS} \cup \bigcap_{sig_{D_j}}$;

6 **else** **foreach** D_j *s.t.* $T_{D_j} = M_S$ **do** $\underline{BCS} := \underline{BCS} \cup \bigcap_{sig_{D_j}}$; $\underline{BICS} := \underline{BICS} \cup \bigcap_{sig_{D_j}}$;

7 **return** $\underline{CS}_k$, $\underline{BCS}$, $\underline{ICS}_k$, $\underline{BICS}$;

Algorithm 1: An algorithm for Common Subsumers enumeration

	D_1	D_2	D_3	D_4	D_5
$Nick$	1	1	1	0	0
$Frank$	0	1	1	0	0
$Robert$	0	0	1	1	0
$Fred$	1	0	0	0	1

	R_1	R_2
$Nick$	1	1
$Frank$	1	1
$Robert$	0	1
$Fred$	0	0

Figure 1: Example Collection and Concept Subsumers Matrixes

$\underline{ICS}_k$, too (line 4). The maximum cardinality is $M_S = 3$, so the check in line 5 fails and the flow in line 6 is followed. D_3 is the only concept component with cardinality M_S: its common signature class, equivalent to D_3 itself, is then added to $\underline{BCS}$ and $\underline{BICS}$ (line 6).

3.2 Target Core Competence Reaching Evaluation

An approach was proposed in [Colucci et al., 2007a] to evaluate whether a company possesses a given Core Competence, taken as target. The approach implemented an algorithm performing a subsumption check for each profile description in the repository, in order to determine the number of employees holding the target knowledge. We here add to this approach an explanation feature, providing , in case the target is not reached, the reasons why this happens.

Consider a collection of concept descriptions $C_1, \ldots, C_n$ in $\mathcal{ALN}$, representing employees knowledge profiles, and a target Core Competence description R, in $\mathcal{ALN}$. In the following, we specialize Definition 7, w.r.t. a concept description R.

Definition 9 (Concept Subsumers Matrix) Let $C_1, \ldots, C_n$ be a collection of concept descriptions C_i and R a concept description, both in a Description Logic $\mathcal{L}$.

We define the **Concept Subsumers Matrix** as a Subsumers Matrix in which $D_j \in \{D_1, \ldots, D_m\}$ are the concept components of R, denoted by R_j.

We here provide Algorithm 2 for evaluation of target Core Competence reaching, with explanation features. The algorithm checks first of all if there are components of the target Core Competence possessed by a number of employees at least equal to the required threshold value k (line 1). If such a preliminary condition does not hold, an

Input : Concept Subsumers Matrix $S = (s_{ij})$ for a collection of concepts $\{C_1, \ldots, C_n\}$ and a concept R in $\mathcal{ALN}$, integer $k < n$
Output: Set of Explanations $E = \{E_l | E_l = \{(C_i, R_j)\}\}$
1 **if** *there exists any* $T_{R_j} \geq k$ **then** compute $Min = min\{T_{R_j} | T_{R_j} \geq k\}$;
2 **else** $E = \emptyset$;
3 **foreach** $R_j \in \{R_1, \ldots, R_m\}$ *such that* $T_{R_j} = Min$ **do**
4 **if** $\bigcap_{sig_{R_j}} \equiv R$ **then return** $E = \{\top\}$;
5 **else foreach** R_h, $h \neq j$ *such that* $sig_{R_j} \not\subseteq sig_{R_h}$ **do**
6 **foreach** C_i **do if** $S_{ij} - S_{ih} = 1$ **then** $E_l := E_l \cup (C_i, R_h)$
7 $E := E \cup E_l$;
8 **return** E;

Algorithm 2: An Algorithm for Target Core Competence Reaching evaluation

explanation process makes no sense: there is no component of the target knowledge that belongs to the company know-how. The set of explanations is therefore empty (line 2). If the check does not fail, instead, only components with cardinality equal to Min, the minimum cardinality bigger than k, are considered (line 3). The target reaching evaluation process is started from any of these components: if any common signature class embeds all components of R (line 4), R is a Core Competence for the company and no explanation is required ($E = \{\top\}$, line 4). If the test in line 4 fails, a different explanation process is started from each component R_j with cardinality equal to Min, in order to return the set E of alternative solutions for reaching the target. In particular the test fails if a set of at least k employees possesses only part of the components of R. The objective is then that of individuating the employees which could undertake a learning process for the missing knowledge[Colucci et al., 2005]. For this reason, for each component not in the common signature class of the one initially selected (line 5), the employees to train are selected (line 6). The pairs component-employee selected according to the process detailed before are proposed as explanation E_l (line 6) corresponding to the initial R_j. The returned set E is then made up by all the alternative explanation solutions referred to different R_j with cardinality equal to Min.

Consider the tiny example running throughout our paper, and the Core Competence $CreativeAssetAllocation \equiv \forall knows.AssetAllocation \sqcap \forall isAbleTo.Creativeness$. The concept components deriving from R are $R_1 = \forall knows.AssetAllocation$ and $R_2 = \forall isAbleTo.Creativeness$, and the concept subsumers matrix is given in the right-hand side of Figure 1. If $k = 2$, only R_1 is selected in line 3 and, given that the check in line 4 is satisfied, the Core Competence may be considered reached, as shown by the value $\top$ in the returned explanation set. If $k = 3$, instead, only R_2 is selected in line 3; as the check in line 4 fails, in line 5 R_1 is selected and, according to the check in line 6, the pair (Robert, R_1) is added to E_l. No other explanation is needed in E_l which is proposed as the first alternative in the returned explanation set E (line 7). The loop in lines 3–7 stops because there are no more components satisfying the check in line 3; the only explanation alternative is then the pair (Robert, R_1): if Robert acquires Asset Allocation knowledge, the company may reach Creative Asset Allocation Core Competence.

4 Conclusions

We have presented a general framework for Core Competence evaluation in knowledge intensive companies. The proposed approach exploits the formalization of company skill sources in a DL knowledge base, which is investigated both for extracting unknown Core Competence and for checking for possession of known one, taken as target. The former process implements specifically introduced DL reasoning services; the latter enriches the check for target Core Competence possession with explanation solutions, suggesting useful learning solutions in case the target is not reached. Both of the approaches are being implemented in the framework of Impakt, a novel and optimized commercial system for competences and skills management [Colucci et al., 2007b], which will be released late this year by D.O.O.M.srl.

Acknowledgement

This work has been supported in part by EU-FP6-IST-26896 project and Apulia Region funded projects PE_013 *Innovative models for customer profiling* and PS_092 *DIPIS*.

References

[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.

[Barney, 1991] Barney, J. B. (1991). Firm Resources and Sustained Competitive Advantage. *Journal of Management*, 17(1):99–120.

[Cohen et al., 1992] Cohen, W., Borgida, A., and Hirsh, H. (1992). Computing least common subsumers in description logics. In *Proc. of AAAI-92*, pages 754–760. AAAIP.

[Cohen and Hirsh, 1994] Cohen, W. W. and Hirsh, H. (1994). Learning the CLASSIC description logics: Theorethical and experimental results. In *Proc. of KR'94*, pages 121–133.

[Colucci et al., 2005] Colucci, S., Di Noia, T., Di Sciascio, E., Donini, F., and Ragone, A. (2005). Semantic-based automated composition of distributed learning objects for personalized e-learning. In *The Semantic Web: Research and Applications. In Proc. of ESWC '05*, volume 3532 of *LNCS*, pages 633–648. Springer-Verlag.

[Colucci et al., 2008] Colucci, S., Di Sciascio, E., and Donini, F. (2008). Partial and informative common subsumers of concept collections in description logics. In *Proc. of DL 2008*.

[Colucci et al., 2007a] Colucci, S., Noia, T. D., Sciascio, E. D., Donini, F., and Ragone, A. (2007a). Measuring core competencies in a clustered network of knowledge. In *Knowledge management: Innovation, technology and cultures*, volume 6 of *Innovation and Knowledge Management*, pages 279–291.

[Colucci et al., 2007b] Colucci, S., Noia, T. D., Sciascio, E. D., Donini, F., and Ragone, A. (2007b). Semantic-based skill management for automated task assignment and courseware composition. *Journal of Universal Computer Science*, 13(9):1184–1212.

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

[L. Halawi and McCarthy, 2005] L. Halawi, J. A. and McCarthy, R. (2005). Resource-Based View of Knowledge Management for Competitive Advantage. 3(2):75–86.

[Meso and Smith, 2000] Meso, P. and Smith, R. (2000). A Resource-Based View of Organizational Knowledge Management Systems. *Journal of Knowledge Management*, 4(3):224–231.

[Wernerfelt, 1995] Wernerfelt, B. (1995). The Resource-Based View of the Firm: Ten Years After. *Strategic Management Journal*, 16(3):171–174.