Measuring Core Competencies in a Clustered Network of Knowledge

S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini*, A. Ragone
SisInfLab
Politecnico di Bari, Bari, Italy
* Università della Tuscia, Viterbo, Italy

Abstract

Sustainable competitive advantage is often hidden in unique company capabilities, creating knowledge differentials. In particular core competencies have been recognized as strategic knowledge for a company and defined as capabilities providing customer benefits, hard to be imitated from competitors and possessing leverage potential. Nevertheless the process of extracting core competencies results quite complex because of the intangibility of knowledge and its automation calls for competencies formalization. We propose an ontology based approach and a system facilitating knowledge elicitation and automating core competencies extraction process. The system deals also with the automatic clustering of companies in knowledge classes according to the competence tehy hold.

1 Introduction

Knowledge management (KM) represents a crucial research object in recent years. Several proposals of solutions and investigations populate literature about KM, so that a universal agreement on its definition is hard to reach [13, 16]. Nevertheless the role of KM in achieving competitive advantage seems to be universally recognized since first investigations. In particular many research contributions sustain the resource based theory of the firm [15, 14] suggesting to search for sustainable competitive advantage in unique company capabilities [8, 10, 2, 3].

In [7] the notion of *core competencies* was introduced to indicate the strategic knowledge of a company. A core competency is defined as a sort of capability providing customer benefits, hard to be imitated from competitors and possessing leverage potential.

Further definitions of core competencies have been proposed in literature in the attempt of finding methods for detecting such specializing knowledge [12, 9].

The process of individuating core competencies is in fact usually characterized by high complexity and low objectivity because of the intangibility of knowledge itself and difficulties inherent in formalizing them.

In order to make more objective and efficient such task, knowledge management systems in use by companies should be endowed with components facilitating automatic knowledge elicitation and core competencies extraction.

A variety of knowledge management systems(KMSs) have in fact been proposed to help companies in excelling among competitors by managing their own distinguishing capabilities[6].

Such approaches call for a formalization of knowledge description and extraction in order to disambiguate the knowledge elicitation process and take advantage from the formalization effort.

If the knowledge held by a company is formally represented, such structured information can be exploited in order to automatically extract possible fields of excellence.

In this paper we propose a KMS supporting management in learning whether a target core competency is held by the company. In the following, we give to this problem the name *learning problem*.

In the solution we propose we observe that core competencies are not necessarily held by the whole company personnel. In order to understand the rationale of this observation, consider the following simple example. In a small consulting company only three people are employed: *Alex*, an engineer with knowledge about C++; *Frank*, a TCP/IP expert, skilled in Java; *Mike*, a C programmer. Assume also that knowledge about Java and C++(respectively C) implies knowledge about Object Oriented Programming-OOP Language(respectively Procedural Programming Language-PP). Moreover knowledge about OOP and PP languages implies knowledge about Programming Language. If we give only to competence held by all of the employees the name of core competency, and we take OOP as target, we may simply assert that our small company does not reach the target. If we instead give up to such a full skill coverage and assume that core competencies are those held by at least half of the employees, we may assert that the target core competency is reached.

In order to implement a system automatically solving the proposed learning problem personnel profiles need to be formalized according to a shared ontology describing the skills domain.

Target core competencies need to be classified according to the same ontology for their semantics to be unambiguously defined.

The formalism we employ for knowledge representation is the one of Description Logics (DLs), a family of logical formalisms providing also reasoning services useful for the learning problem.

If the same skills ontology is shared by different companies, a classification of organizations in competence clusters is also possible.

Different clusters include companies sharing formalized classes of knowledge. Through the proposed learning process a company may discover its profile in terms of core competencies and check whether it can join a given knowledge class.

In particular we exploit standard reasoning services to perform the learning process, while non standard explanation services are exploited to investigate on possible classifications w.r.t. the formalized clusters.

In the next section we show basic Description Logics formalism, detailing also provided standard and non-standard reasoning services to make the paper self contained. In Section 3 the solution to the problem of learning the possession of target core competencies is presented, together with the companies clustering approach. The proposed framework for knowledge elicitation and measurement is exemplified in Section 4. Conclusion and future work close the paper.

2 Formal Background

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

DLs describe different knowledge domains through concept and role *expressions*, combining basic syntax elements, *concept* names and *role* names, by means of *constructors*. Intuitively concepts describe objects belonging to the knowledge domain taken into account; for example in our competence management domain `ManagerialEngineer`, `ProcessPlanner`, `AssetAllocation` represent concept names.

Analogously roles represent relations among concepts: `advancedKnowledge`, `toolsKnowledge` are examples of role names in competence management domain.

Each logic formalism in the DLs family is characterized by its distinguishing set of constructors, even though some elements of the set are shared by all DLs.

The base language is called $\mathcal{AL}$ and contains the following constructors:

- *conjunction*:
  $\texttt{Engineer} \sqcap \texttt{ProcessPlanner}$ denotes the class of engineers working as process planner.

- *universal role quantification*:
  e.g., $\texttt{ProcessPlanner} \sqcap \forall \texttt{hasMasterDegree.Mathematics}$ which describes process planners only with degrees in Mathematics.

- *unqualified existential role quantification*:
  e.g., $\texttt{Engineer} \sqcap \exists \texttt{advancedKnowledge}$ which describes the set of engineers endowed with some knowledge at an advanced level.

- *top*: $\top$, the universal concept, whose interpretation is the whole domain.

- *bottom*: $\bot$, the concept whose interpretation is always an empty set.

- *negation of atomic concepts*: $\neg\texttt{Skill}$, denotes the class of concepts which are not skills.

$\mathcal{AL}$ may be extended by adding new constructors, increasing representation expressiveness.

Concept expressions can be used in *inclusion assertions*, and *definitions*, which impose restrictions on possible interpretations according to the knowledge elicited for a given domain. For example, we could impose that graduates are entities with at least one master degree $\texttt{Graduate} \sqsubseteq \exists\texttt{hasMasterDegree}$. Definitions are useful to give a meaningful name to particular combinations, as in $\texttt{Engineer} \equiv \exists\texttt{hasMasterDegree} \sqcap \forall\texttt{hasMasterDegree.Engineering} \sqcap \exists\texttt{basicKnowledge} \sqcap \forall\texttt{basicKnowledge.}(\texttt{Mathematics} \sqcap \texttt{Physics} \sqcap \texttt{Design})$ defining an engineer as an entity holding only degrees in engineering and basic knowledge about mathematics, physics and design. Historically, sets of such inclusions are called TBox (Terminological Box) and represent what is called an *ontology* in a knowledge representation system. In simple DLs, only a concept name can appear on the left-hand side of an inclusion.

An excerpt of the ontology in DL all the examples in this paper refer to is shown in Figure 1. The ontology exploits $\mathcal{ALN}$: such a language represents a good trade-off between expressiveness and computational complexity of reasoning services [4] and adds to the base language $\mathcal{AL}$ the following constructor:

- *number restrictions*:
  e.g., $\texttt{Engineer} \sqcap (\leq 3\ \texttt{hasWorkExperience})$ expresses engineers having at most three working experiences, and $\texttt{AssetManager} \sqcap \forall\texttt{hasWorkExperience.}(\geq 3\ \texttt{years})$ describes asset managers endowed of at least 3 years of working experience.

The basic reasoning problems for concepts in a DL are satisfiability, which accounts for the internal coherency of the description of a concept (no contradictory properties are present), and subsumption [4], which accounts for the more general/more specific relation among concepts, that forms the basis of a taxonomy. More formally, given an ontology $\mathcal{T}$:

- a concept C is satisfiable w.r.t. $\mathcal{T}$ if there exists an interpretation in $\mathcal{T}$ mapping C into a nonempty set, unsatisfiable otherwise.

- a concept C subsumes a concept D according to $\mathcal{T}$ if every interpretation assigns to C a set included in the one assigned to D, a relation that we denote $\mathcal{T} \models C \sqsubseteq D$.

The framework we propose in this paper uses non standard reasoning services in order to classify companies w.r.t. competence clusters.

Explanations about both cluster fitting and competences missed by the company to fit the cluster result in fact useful.

In [5] the Concept Abduction Problem (CAP) was introduced and defined as a non standard inference problem for DLs to provide a logic-based explanation in the wider resource retrieval scenario. A CAP is formally defined as follows.

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

In [5] also minimality criteria for H and a polynomial algorithm which finds irreducible solutions in $\mathcal{ALN}$ bushy TBoxes [11] have been proposed.

$\texttt{Engineering} \sqsubseteq \texttt{Degree}$
$\texttt{Mathematics} \sqsubseteq \texttt{Degree}$
$\texttt{Physics} \sqsubseteq \texttt{Skill}$
$\texttt{Psychology} \sqsubseteq \texttt{Skill}$
$\texttt{QualityAssuranceTechniques} \sqsubseteq \texttt{Skill}$
$\texttt{StatisticalMethods} \sqsubseteq \texttt{Skill}$
$\texttt{DistributionManagement} \sqsubseteq \texttt{Skill}$
$\texttt{SupplyChain} \sqsubseteq \texttt{Skill}$
$\texttt{AssetAllocation} \sqsubseteq \texttt{Skill}$
$\texttt{Design} \sqsubseteq \texttt{Skill}$
$\texttt{ClientServerProtocol} \sqsubseteq \texttt{Skill}$
$\texttt{ComputerGraphics} \sqsubseteq \texttt{Skill}$
$\texttt{InformationSystems} \sqsubseteq \texttt{Skill}$
$\texttt{DBMS} \sqsubseteq \texttt{InformationSystems}$
$\texttt{ERPsystem} \sqsubseteq \texttt{InformationSystems}$
$\texttt{TCP/IP} \sqsubseteq \texttt{InformationSystems}$
$\texttt{InternetDevelopment} \sqsubseteq \texttt{Skill}$
$\texttt{WebDesigning} \sqsubseteq \texttt{InternetDevelopment}$
$\texttt{InternetTechnology} \sqsubseteq \texttt{Skill}$
$\texttt{InternetUse} \sqsubseteq \texttt{Skill}$
$\texttt{MarkUpLanguages} \sqsubseteq \texttt{Skill}$
$\texttt{HTML} \sqsubseteq \texttt{MarkUpLanguages}$
$\texttt{SoftwareEngineering} \sqsubseteq \texttt{Skill}$
$\texttt{UML} \sqsubseteq \texttt{SoftwareEngineering}$
$\texttt{Programming} \sqsubseteq \texttt{Skill}$
$\texttt{OOP} \sqsubseteq \texttt{Programming}$
$\texttt{Java} \sqsubseteq \texttt{OOP}$
$\texttt{C++} \sqsubseteq \texttt{OOP}$
$\texttt{StructuralProgramming} \sqsubseteq \texttt{Programming}$
$\texttt{C} \sqsubseteq \texttt{StructuralProgramming}$
$\texttt{ScriptLanguages} \sqsubseteq \texttt{Programming}$
$\texttt{Javascript} \sqsubseteq \texttt{ScriptLanguages}$
$\texttt{VBscript} \sqsubseteq \texttt{ScriptLanguages}$
$\texttt{OperationsOptimization} \sqsubseteq \texttt{Skill}$
$\texttt{DistributionManagement} \sqsubseteq \texttt{Skill}$
$\texttt{ProductionManagement} \sqsubseteq \texttt{Skill}$
$\texttt{ProcessPerformanceMonitoring} \sqsubseteq \texttt{Skill}$
$\texttt{WorkflowManagement} \sqsubseteq \texttt{Skill}$
$\texttt{ProcessPlanning} \sqsubseteq \texttt{Skill}$

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

$\texttt{AssetManager} \equiv \exists\texttt{hasMasterDegree} \sqcap \exists\texttt{hasWorkExperience} \sqcap \forall\texttt{hasWorkExperience.}(\leq 10\ \texttt{years}) \sqcap \exists\texttt{advancedKnowledge} \sqcap \forall\texttt{advancedKnowledge.AssetAllocation}$

$\texttt{ManagerialEngineer} \equiv \texttt{Engineer} \sqcap \forall\texttt{advancedKnowledge.ProductionManagement} \sqcap \exists\texttt{basicKnowledge} \sqcap \forall\texttt{basicKnowledge.SupplyChain}$

$\texttt{Mathematician} \equiv \exists\texttt{hasMasterDegree} \sqcap \forall\texttt{hasMasterDegree.Mathematics} \sqcap \exists\texttt{advancedKnowledge} \sqcap \forall\texttt{advancedKnowledge.StatisticalMethods}$

$\texttt{ProcessPlanner} \equiv \exists\texttt{hasMasterDegree} \sqcap \exists\texttt{advancedKnowledge} \sqcap \forall\texttt{advancedKnowledge.ProcessPlanning} \sqcap \exists\texttt{basicKnowledge} \sqcap \forall\texttt{basicKnowledge.}(\texttt{ProductionManagement} \sqcap \texttt{Design})$

Figure 1: Skills Ontology excerpt

The solution to a CAP can be interpreted as what has to be hypothesized in S, and in a second step added to, in order to make S more specific than D, which would make subsumption result true. Concept Abduction extends then subsumption.

Reverting to our organizational scenario, if K be a knowledge class and C is a company profile, explanations provided by the solution of the concept abduction problem $\langle \mathcal{ALN}, C, K, \mathcal{T} \rangle$ may be helpful to classify C or understand the reasons why C does not fit K.

3 An automatic system for core competencies extraction and companies classification

In this section we show how to exploit reasoning services provided by DLs to automatize both the learning and the classification problem. We propose here a semantic based system supporting such activities and sketch its architecture in Fig.2.

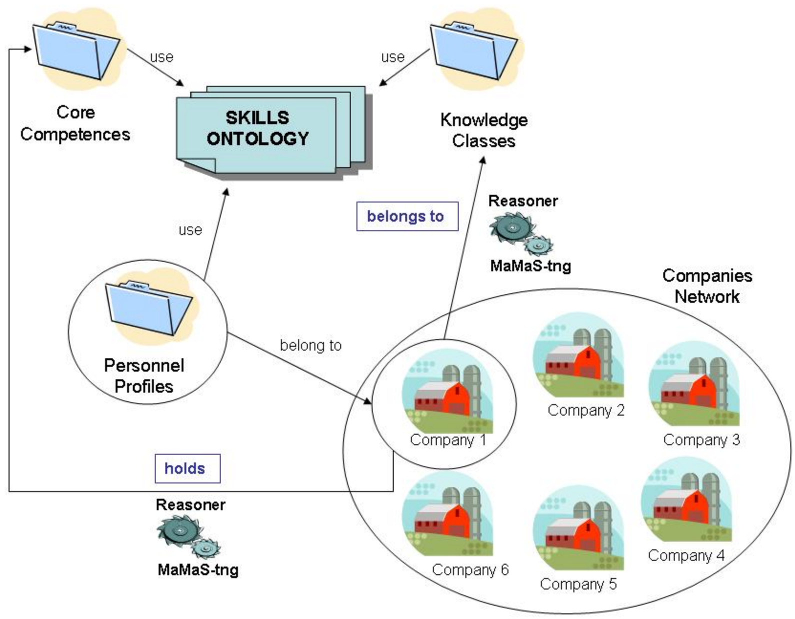

Figure 2: System Architecture

The system is aimed at serving a network of companies sharing a skills ontology as vocabulary for competence description. According to such ontology both core competencies and knowledge classes are formalized.

Figure 3 shows a snapshot of the Core Competencies Definition Interface our system provides.

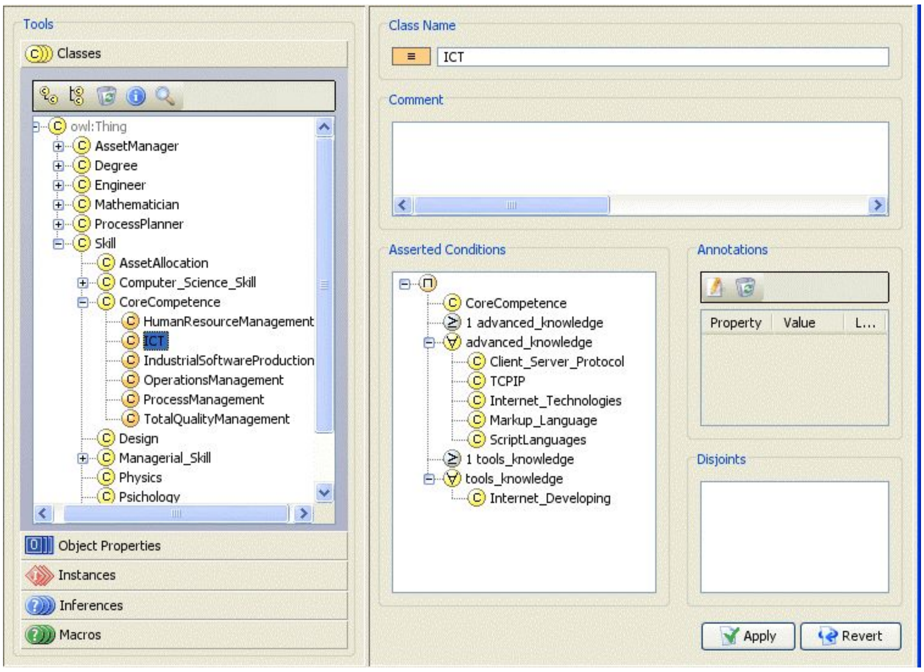

Figure 3: Core Competencies Definition Interface

Each company is characterized by a number of formalized representations describing the professional profiles of its internal personnel according to the skills ontology.

The proposed system supports each company of the network in the activity of checking whether it **holds** a given core competency and **belongs to** a given knowledge class.

Both activities ask for reasoning services: subsumption and abduction, respectively; in our system such services are provided by MaMaS-tng[1], a reasoning engine developed to implement also non standard reasoning services in $\mathcal{ALN}$.

In the following the two activities supported by the proposed system are detailed.

3.1 Learning target core competencies

Consider the problem of checking for the possession of a given core competency and remember the observation that such a target competence does not necessarily need to be held by whole company personnel. In order to solve the learning problem we intuitively need to check whether the target core competency is held

[1]**MA**tch**MA**ker**S**ervice at http://sisinflab.poliba.it:8080/MAMAS-tng.

by a sufficiently high number of people, with such a threshold value specified by the company management.

Formally, let T be a target and $P = P_i, i = 1, \ldots, n$ a set of individual profiles describing employees competences, both formalized as DL concepts according to an ontology $\mathcal{T}$. Moreover let k, with $k < n$, be the threshold number of profiles the company accepts to state the target is reached.

In order to solve our learning problem we have to check whether at least k of the n concepts composing P subsume T. An algorithm performing such a process is proposed in the following:

```
Input  : ontology 𝒯, P_i with i = 1...n and T concepts in DL,
         satisfiable w.r.t. 𝒯
Output: tr: number of concepts P_i subsuming T
i = 1;
tr = 0;
foreach i <= n do
    if P_i ⊑ T then
        tr = tr + 1;
    end
    i = i + 1;
end
return tr;
```

Algorithm 1: An algorithm for target core competency reaching evaluation

The algorithm returns the number tr of concepts subsuming the target core competency; if $tr > k$ the management may assume the target has been reached, and evaluate how crucial is the competence for its strategy by comparing tr with n.

3.2 Clustering companies according to their competences

In the depicted scenario core competencies need to be formalized as DL concepts $CC_j, j = 1 \ldots m$ in the ontology $\mathcal{T}$ describing the skill domain.

Classes of competences may be defined on the basis of such formalizations; examples of knowledge classes are listed below:

- ICT based solutions for human resource management
- Software production for operations management
- Total Quality management solutions

The core competencies resulting form the target reaching evaluation process constitute the conjuncts of a description formalizing the crucial knowledge held by a company, indicated by C.

After a company detects its core competencies through the process detailed in Section 3.1 it could be useful to know whether it can join a competence class or, in case it cannot, the reasons why it happens.

The company clustering process is formally made up by the following steps:

1. formalization of the knowledge domain in an ontology $\mathcal{T}$
2. formalization w.r.t. $\mathcal{T}$ of the profile of the company to cluster: $C = \sqcap_j CC_j$
3. formalization w.r.t. $\mathcal{T}$ of the set of knowledge classes clustering the network of knowledge: $K = \{K_i\}, i = 1 \dots n$
4. solution of one Concept Abduction Problem for each K_i:
   $H = \langle \mathcal{ALN}, C, K_i, \mathcal{T} \rangle$

The solution H of the Concept Abduction Problem in Step 4 represents an explanation about company cluster fitting. In particular H represents the competences to be hypothesized to make C more specific than K_i , in other words the knowledge the company lacks to join K_i. If the CAP returns $H = \top$ no additional competence is required to join K_i, and we can assume the company belongs to the knowledge class.

4 Application scenario

The effectiveness of the proposed framework is shown in the following through an extended example.

A company is supposed to belong to a network of organizations sharing the same skills ontology $\mathcal{T}$ for knowledge representation. The ontology is sketched in Figure 1 and provides the vocabulary to each company of the network for describing the held competences in terms of employees profiles or project specifications documents. The vocabulary provided by the skills ontology is exploited also for the definition of target core competencies shared by the different companies of the network. Examples of such formalizations are shown in Table 1. Imagine now the scenario in which one of the companies of the network needs

Core Competence	Formal Definition
ICT: Information and Communication Technology	∃advancedKnowledge ⊓ ∀advancedKnowledge.ClientServerProtocol ⊓ TCP/IP ⊓ InternetTechnology ⊓ MarkUpLanguages ⊓ ScriptLanguages ⊓ ∃toolsKnowledge ⊓ ∀toolsKnowledge.InternetDevelopment
HumanResourcesManagement	∃advancedKnowledge ⊓ ∀advancedKnowledge.AssetAllocation ⊓ ∃basicKnowledge ⊓ ∀basicKnowledge.Psychology
IndustrialSoftwareProduction	∃advancedKnowledge ⊓ ∀advancedKnowledge.(ComputerGraphics ⊓ OOP ⊓ StructuralProgramming ⊓ SoftwareEngineering) ⊓ ∃basicKnowledge ⊓ ∀basicKnowledge.(Design ⊓ InformationSystems)
OperationsManagement	∃advancedKnowledge ⊓ ∀advancedKnowledge.(OperationsOptimization ⊓ ProductionManagement ⊓ DistributionManagement) ⊓ ∃basicKnowledge ⊓ ∀basicKnowledge.SupplyChain
ProcessManagement	∃advancedKnowledge ⊓ ∀advancedKnowledge.(ProcessPlanning ⊓ ProcessPerformanceMonitoring) ⊓ ∃basicKnowledge ⊓ ∀basicKnowledge.ProductionManagement
TotalQualityManagement	∃advancedKnowledge ⊓ ∀advancedKnowledge.(QualityAssuranceTechniques ⊓ StatisticalMethods) ⊓ ∃basicKnowledge ⊓ ∀basicKnowledge.ProcessPerformanceMonitoring

Table 1: Core Competencies Definitions

to check for the possession of all the core competencies in Table 1.

Let the company be composed by the set of employees whose profiles are formalized in Table 2.

Profile	**Formal Definition**
Alan	ProcessPlanner ⊓ Mathematician ⊓ ∃advancedKnowledge ⊓ ∀advancedKnowledge.ProcessPerformanceMonitoring ⊓ InformationSystems ⊓ QualityAssuranceTechniques
Jack	AssetManager ⊓ ∃basicKnowledge ⊓ ∀basicKnowledge.Psychology
John	Engineer ⊓ ∃advancedKnowledge ⊓ ∀advancedKnowledge.(ClientServerProtocol ⊓ HTML ⊓ InternetTechnology ⊓ Javascript ⊓ TCP/IP ⊓ SoftwareEngineering ⊓ C++ ⊓ C ⊓ ComputerGraphics) ⊓ ∃basicKnowledge ⊓ ∀basicKnowledge.(InternetUse ⊓ InformationSystems) ⊓ ∃toolsKnowledge ⊓ ∀toolsKnowledge.WebDesigning ⊓ ∃hasWorkExperience ⊓ ∀hasWorkExperience.(≥ 2 years)
Mark	Engineer ⊓ ∃advancedKnowledge ⊓ ∀advancedKnowledge.(VBscript ⊓ Java ⊓ C ⊓ UML ⊓ ComputerGraphics ⊓ ClientServerProtocol) ⊓ ∃basicKnowledge ⊓ ∀basicKnowledge.InformationSystems
Simon	Mathematician ⊓ ∃advancedKnowledge ⊓ ∀advancedKnowledge.QualityAssuranceTechniques⊓ ∃basicKnowledge ⊓ ∀basicKnowledge.ProcessPerformanceMonitoring
Tom	ManagerialEngineer ⊓ ∃advancedKnowledge ⊓ ∀advancedKnowledge.(InternetTechnology ⊓ OperationsOptimization ⊓ ProductionManagement ⊓ DistributionManagement) ⊓ ∃toolsKnowledge ⊓ ∀toolsKnowledge.WebDesigning

Table 2: Employees profiles definitions

The algorithm presented in Section 3.1 may be exploited to learn whether the company, identified by the set of profile descriptions in Table 2, holds any of the target core competencies defined in Table 1. In other words the algorithm is applied taking as input one core competency T at a time together with the whole set P. Suppose now the company accepts a threshold value $k = n/3$ as limit to consider reached one of the targets. The algorithm returns the following results:

1. **inputs**: $T =$ ICT, $P = \{Alan, Jack, John, Mark, Simon, Tom\}$
   output: $tr = 1$

2. **inputs**: $T =$ HumanResourcesManagement,
   $P = \{Alan, Jack, John, Mark, Simon, Tom\}$
   output: $tr = 1$

3. **inputs**: $T =$ IndustrialSoftwareProduction,
   $P = \{Alan, Jack, John, Mark, Simon, Tom\}$
   output: $tr = 2$

4. **inputs**: $T =$ OperationsManagement, $P = \{Alan, Jack, John, Mark, Simon, Tom\}$
   output: $tr = 1$

5. **inputs**: $T =$ ProcessManagement, $P = \{Alan, Jack, John, Mark, Simon, Tom\}$
   output: $tr = 1$

6. **inputs**: $T =$ TotalQualityManagement,
   $P = \{Alan, Jack, John, Mark, Simon, Tom\}$
   output: $tr = 2$

The cardinality of set P equals to $n = 6$, so that the accepted threshold value is $k = 2$. On the basis of such hypothesis we can assert that the company reaches only the target core competencies: `IndustrialSoftwareProduction` and `TotalQualityManagement`.

In fact both $John$ and $Mark$ is subsumed by `IndustrialSoftwareProduction`, while both $Simon$ and $Alan$ are subsumed by `TotalQualityManagement`.

On the contrary only $John$ is subsumed by `ICT`, $Jack$ is subsumed by `HumanResourcesManagement`, Tom is subsumed by `OperationsManagement`, and $Alan$ is subsumed by `ProcessManagement`.

The learning process showed so far allows the company to formalize its own profile by simply conjunct the discovered core competencies:
$C = \texttt{IndustrialSoftwareProduction} \sqcap \texttt{TotalQualityManagement}$.

Revert now to the whole network of organizations and consider the knowledge classes formally defined in table 3. Such definitions represent some of the

Knowledge Class	Formal Definition
ICT Solutions For Human Resources Management	$\texttt{ICT} \sqcap \texttt{HumanResourcesManagement} \sqcap \exists\texttt{basicKnowledge} \sqcap \forall\texttt{basicKnowledge.WorkflowManagement}$
Industrial Software Production For Operations Management	$\texttt{IndustrialSoftwareProduction} \sqcap \texttt{OperationsManagement} \sqcap \exists\texttt{basicKnowledge} \sqcap \forall\texttt{basicKnowledge.ProcessPerformanceMonitoring}$
Total Quality Management Solutions	$\texttt{TotalQualityManagement} \sqcap \exists\texttt{basicKnowledge} \sqcap \forall\texttt{basicKnowledge.InformationSystems}$

Table 3: Knowledge Classes Definitions

clusters in which the network is fractioned. By solving a concept abduction problem between each of the clusters in Table 3 and the company profile C we can evaluate whether the company belongs to the corresponding knowledge class or, if not, the reasons why this does not happen.

The explanations relative to the example of our company (whose profile is defined by C) are shown in the following:

- **Knowledge Class**: `ICT Solutions For Human Resources Management`
  Explanation Results:
  $H_1 = \exists\texttt{toolsKnowledge} \sqcap \forall\texttt{toolsKnowledge.InternetDevelopment}$
  $\sqcap\forall\texttt{basicKnowledge.(Psychology} \sqcap \texttt{WorkflowManagement)}$
  $\sqcap\forall\texttt{advancedKnowledge.(AssetAllocation}\sqcap\texttt{ClientServerProtocol}\sqcap\texttt{TCP/IP}\sqcap\texttt{InternetTechnology}\sqcap$
  $\texttt{MarkUpLanguages} \sqcap \texttt{ScriptLanguages)}$

- **Knowledge Class**: `Industrial Software Production For Operations Management`
  Explanation Results:
  $H_2 = \forall\texttt{basicKnowledge.SupplyChain}$
  $\sqcap\forall\texttt{advancedKnowledge.(ProductionManagement}\sqcap\texttt{DistributionManagement}\sqcap\texttt{OperationsOptimization)}$

- **Knowledge Class**: `Total Quality Management Solutions`
  Explanation Results: $H_3 = \top$

The results above show that our company belongs only to the knowledge class `Total Quality Management Solutions`: no additional knowledge is needed to join the cluster as shown by the explanation: $H_3 = \top$. In order to join the other knowledge classes the company needs instead some extra knowledge, detailed by the explanation results H_2 and H_3. In particular, in order to fit the `ICT Solutions For Human Resources Management` cluster the company would need knowledge about tools for Internet Development, basic knowledge about psychology and work flow management and advanced knowledge about asset allocation, client server protocol, TCP/IP, Internet technology, mark up languages and script languages. In order to fit the `Industrial Software Production For Operations Management` cluster, the company would need instead of basic knowledge about supply chain and advanced knowledge about production, distribution and operations management.

5 Conclusions

We have shown how a company may extract its core competencies and collocate itself in a clustered knowledge network thanks to a knowledge based framework. It is noteworthy that the extracted competences may belong to different knowledge fields, such as managerial skills and computer science skills in the given example. Such distinctions makes the manual detection process much harder, given that a company is usually focused on a specific competence branch and may neglect hidden skills relative to different branches. An automatic system implementing the proposed framework represents then a knowledge differential leading to sustainable competitive advantage.

References

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

[2] J. B. Barney. Strategic Factor Markets: Expectations, Luck and Business Strategy. *Management Science*, 32:1231–1241, 1986.

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

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

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

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

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

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

[9] C. C. Markides and P. J. Williamson. Related diversification, core competences and corporate performance. *Strategic Management Journal*, 15:49–65, 1994.

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

[11] B. Nebel. Terminological Reasoning is Inherently Intractable. *AIJ*, 43:235–249, 1990.

[12] R. R. Nelson. Why do firms differ, and how does it matter? *Strategic Management Journal*, 12:61–74, 1991.

[13] S. Stucky P. Seeman, D. DeLong and E. Guthrie. Building intangible assets: A strategic framework for investing in intellectual capital. In *Proc. of Second International Conference on the Practical Applications of Knowledge Management (PAKeM99)*, 1999.

[14] B. Wernerfelt. A Resource-Based View of the Firm. *Strategic Management Journal*, 5(2):171–180, 1984.

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

[16] K. Wiig. Knowledge management: An introduction and perspective. *Journal of Knowledge Management*, 1(1):6–14, 1997.