Journal of Universal Computer Science, vol. 18, no. 20 (2012), 2893-2919
submitted: 23/11/11, accepted: 28/11/12, appeared: 1/12/12 © *J.UCS*

Ontology-based Approach to Competence Profile Management

Vladimir Tarasov
(School of Engineering at Jönköping University, Sweden
Vladimir.Tarasov@jth.hj.se)

Abstract: Competence management has received much attention during recent years because it contributes to achieving organizational goals and solving problems such as improvement of information flow or competence supply. Many approaches were proposed to modelling competence and using competence models but there is still a lack of research into structures and utilisation of competence profiles in a competence management system. This article addresses this problem by proposing a formal approach to competence profile management. Four project cases are first analysed to elicit requirements of competence profile management, including competence profile operations. After that, an abstract model of competence profile management is formally defined based on the requirements. Finally, an ontology-based implementation of the abstract model is presented including a software architecture of a competence profile management system. The main contribution of this work is formalization of operations on competence profiles and ontology-based implementation of these operations. The proposed implementation architecture can facilitate construction of a competence profile management system.

Key Words: competence, competence management, competence profile, formalisation, ontology-based profile, OWL, Jena
Category: H.1, H.4, M.4

1 Introduction

Competence management can support performing many tasks in an enterprise such as knowledge management [RM09] or expert search [JV11], which contributes to achieving organizational goals. Competence management relies on representing skills and knowledge of a person in the form of a competence model or, more specifically, a competence profile. Much research has been done into modelling competences possessed by a person, e.g. by describing competences required by a role assigned to a worker [BC07] or by creating an information model for learning outcomes and competences [InL12]. Ontologies are suggested for use in competence management systems, e.g. in [BH07] or [JV11]. Competence management systems have been used for competence retrieval [JMP07], determining competency gaps [Paq07] or skill management including competence search and competence matching [CNS$^+$07].

However, despite many approaches proposed to modelling competence and using competence models, there is still lack of research into structures and utilisation of competence profiles in a competence management system that is how profiles can be managed as separate entities. On the other hand, the use of competence profiles in many practical cases, such as competence supply in business cooperation or information supply in healthcare or mass media (see Sect. 3), requires more structured and formal methods for competence profile management. This article addresses this problem and proposes an approach for competence profile management that aims at identification and specification of how competence profiles can be managed in an enterprise or organisation. The main contribution of this work is a formalization of operations on competence profiles and an ontology-based implementation of these operations. The proposed implementation architecture can facilitate construction of a competence profile management system to support an enterprise in many important areas such as information supply, competence supply or analysis of changes in the organisational competence.

The article is structured as follows. Sect. 2 introduces concepts important for competence management and provides an overview of related work. Sect. 3 analyses four project cases to elicit requirements to competence profile management and operations on competence profiles. Sect. 4 formally defines an abstract model of competence profile management that is based on the established requirements. An ontology-based implementation of the abstract model is presented in Sect. 5. An example of using the implementation in a case is shown in Sect. 6. Sect. 7 provides a summary of the work and discussion of the results as well as insight into future work.

2 Background and Related Work

This section starts with competence definitions, and then proceeds with structuring of competences. Findings in construction of competence profiles emphasize the use of ontological representations. It concludes with a description of approaches to competence management.

2.1 Competence Definitions and Competence Models

There are many different definitions of competence in the literature. They usually define competence in relation to a task. The authors in [BK98] conducted two studies and found 21 factors related to competence. Most of the factors can be considered as competences possessed by the individual, which can be applied in work situations. More specifically, [BK98] define a competence as a set of all knowledge forms and personal abilities that are required for performing tasks. The European e-Competence Framework (e-CF) implicitly mentions tasks to be performed by an individual when providing a generic description for a competence [CEN10]. This is further clarified by listing specific knowledge and skills required to perform these tasks. [Gra10] provides an extensive discussion of the nature and structure of competences, which relates abilities with generic work roles from occupational standards and tasks from job descriptions.

Competence levels have been distinguished in various papers that present competence models. The Unified Enterprise Competence Modelling Language (UECML) defines unit competence, which is the competence itself, as representing skills and abilities needed for performing an activity with corresponding levels [PCFG07]. [HBW03] found that cultural competences could be measured against a six-point scale. e-CF defines five proficiency levels: from an “associate” to a “principal” [CEN10]. An approach to define and assign competence levels is investigated in the eCOTOOL project [GS11]. It distinguishes between binary and rankable ability and allows for assignment of levels to the latter. Several levels can be assigned from different schemes. Competence levels and representation of level relationship are also discussed in [Gra10].

Structuring of competence is also addressed by many authors who used different perspectives on competence. [BK98] identified general abilities that are applicable in different situations, e.g. ability to plan, ability to form teams, creativity, ability to provide support and guidance, etc. [HBW03] describe the concept of intercultural sensitivity, which shows how people can perceive cultural differences and act in multicultural environments. e-CF gives an example of structuring ICT-related competences [CEN10]. There are also approaches that encompass several perspectives on competence, e.g. [JMP07] describes individual competences as knowledge gained through education, skills mastered with experience, and behavioural characteristics.

2.2 Competence Profiles and Competence Management

Although not explicitly named in every paper, a competence profile is considered to be composed of all competences of a person or worker. A competence profile often consists of competences related to a work situation or processes. [BC07] describes an approach, where competence is necessitated by an activity. The latter can be related to one or several business domains and a human resource acts in a role in each domain. Every role requires a set of competences. UECML treats individual competence as describing a competence profile of a worker that is non-material resources needed for all activities, which involve this worker [PCFG07]. The approach proposed by [BH07] suggests the CRAI model (competence, resource, aspect, individual). In this model, competences are linked to individual actors and are characterized by sets of knowledge, know-how and behaviour associated to a context.

Many authors suggest the use of ontological implementations of varying expressivity—from RDF graphs to full-fledged ontologies—to represent competence profiles. Modelling of research supervisors is detailed in [LCD05]. A supervisor profile is represented as an RDF (Resource Description Framework) graph. Each graph consists of data about projects, papers and specific descriptions of expertise. An ontology database with expert profiles is created in [JV11] and populated from different data sources. [BH07] propose to implement competence with an ontology and emphasize that a reasoner will support evolution of the competence model through automatic classification. In [Paq07] the author argues that the use of ontologies provides for a well-structured competence model that contains associations between different constituting part of the competence as well as relationships between the domain elements and relevant knowledge.

Recent developments in computerized management of competences can be found in the Integrating Learning Outcomes and Competences (InLOC) project [InL12]. It defines an information model that can be used to represent both learning outcomes and competences (LOCs). The model aims at achieving interoperability between different kinds of ICT systems. The model allows for description of competences acquired by people using LOC definitions and combining several competences using LOC structures and LOC associations. This provides for creation of professional profiles, e.g. for employee management. Each competence can be related to level and topics from a subject classification. Machine-readable representations of competences and levels including RDF triples are also discussed in [Gra10].

Competence models and profiles are employed in competence management systems to support performing many tasks in an organisation. Competences formalized in the form of an ontology were used as part of a competence retrieval system to support analysis, planning and control of business performance of an enterprise in [JMP07]. Competence search is addressed in [CNS+07] that proposes a skill management system, which can be used to assign individuals to tasks. [JV11] describes an ontology-based system that supports creation of project teams with the required competence—it is used for expert search with SPARQL and OntoWiki. The work in [Paq07] describes an ontology-driven e-learning system that supports evaluation of competencies by determining competency gaps, which are used to plan activities to achieve learning goals. Although these examples demonstrate different applications of competence profiles and competence management, there is still lack of work describing structures and utilisation of competence profiles in terms of explicit operations on them.

3 Project Cases

This section introduces four project cases and elicits requirements to competence profile management. Each case deals with competence profiles and demonstrates a practical need for competence profile management. After the requirements analysis, operations on competence profiles to meet the requirements are listed.

3.1 Media-ILOG

The first case concerns the domain of mass media. The goal of the Media Information Logistics project (Media-ILOG), was to improve information flow in a local newspaper, JönköpingsPosten. The newspaper usually gets a lot of input concerning diverse topics from its readers, which come through different channels like e-mails. The problem was that e-mails had to be sorted out manually to be assigned to reporters according to their specialisation. The approach taken in the project was to match reporters' profiles against the arriving e-mails to determine which reporter each e-mail should be forwarded to.

A reporter's competence profile in Media-ILOG has the simplest form—it is a set of domain topics, which the reporter is writing about. Each profile was represented as a fragment of the domain ontology that modelled the article areas of JönköpingsPosten and contained 457 classes totally [SÖS+07]. When a new e-mail arrived, keywords were extracted from the text and mapped to the ontology classes. Then, the found classes were matched against all ontology fragments representing the reporters' profiles. Finally, the comparison results were ranked to list reporters with best matching profiles [BBL07]. In this project an interesting question was to determine changes in the readers' interests because it would help the newspaper to adapt to the changing market. One way to accomplish this could be first to aggregate the competence profiles of all reporters as it shows the current set of the readers' interests. Then tracking changes in the competence profiles can show how the readers' interests change. A reporter's profile may also change when a reporter starts to write about a new topic or stops writing about a topic that is no longer of interest to the readers. Hence, topics can be removed from or added to a competence profile.

3.2 BTG-QR

The second case comes from the healthcare area. One of the subprojects in the project Bridging the Gaps (coordinated by Jönköping County Council) addresses the use of quality registries to enhance the quality of healthcare. Quality registries are databases that contain data regarding treatment of diseases in particular areas.[1] Quality registries are regularly used for generation and dissemination of recent data. However, reports based on few standard templates are too complicated and too long for many categories of users. To improve this, our method is to individually customize the content of a report based on a reader's competence profile, which reflects professional interests of the reader [GLT10].

In this case a competence profile is defined by the roles a professional is assigned to in an organisation. Each role is described by competences/qualifications needed to carry out the work. An ontology containing 160 classes was created to

[1] Many quality registries in Sweden are run by Uppsala Clinical Research Center.

model four groups of workers, which are doctors, nurses, pharmaceutical personnel and healthcare officials [AJ09]. A quality registry report can be adjusted by using the variables that determine the length and content of a report. By matching the person's profile against the ontology fragment representing the variables, one can choose a matching subset of the variables to generate report content relevant to the reader's professional interests. When new roles are assigned to a professional or old ones become obsolete due to changes in the work duties, the profile needs to be updated.

3.3 BTG-UR

Healthcare is also the domain of the third case. Another subproject in the project Bridging the Gaps aims at improving information flow in healthcare organisations. A part of this subproject is intended to support patient treatment in the Urology departments of Jönköping County [IJ12]. The treatment process includes a number of steps, e.g. making urine tests or physical examination of a patient. At each step a urology specialist or nurse may need several documents that describe guidelines for performing activities of this step. The number of documents is relatively big (several dozens) and it takes time to search through the document tree manually. Our approach to improvement is to match the documents against a medical specialist's competence profile to deliver only documents relevant to the current step.

A competence profile in this case is composed of roles a nurse or doctor may act in. Each role includes a number of tasks, which are modelled by the domain ontology. Documents are marked up with semantic tags describing their content. To find relevant documents, first a subprofile should be created by limiting the competence profile to the current task and role. Otherwise, documents may be found that are relevant to other tasks but the current one. Then the extracted subprofile is matched against the semantic tags of documents. The best matching documents are delivered to the nurse/doctor. A medical professional's competence profile may change when new roles and/or tasks are assigned to the person or transferred to another worker.

3.4 BR-IComp

The last case addresses the domain of establishing business cooperation. The project ICT-Support for Formation of Business Relationships with Developing Countries Based on Immigrant Competence (BR-IComp) aimed at supporting Swedish and Vietnamese companies searching for business partners. During business cooperation establishment companies-partners from different countries may encounter various obstacles that are caused by distinctions in how business is done in Sweden and Vietnam, as well as cultural differences that may lead to misunderstanding in communication. Members of the Vietnamese diaspora in Sweden could help to overcome these hurdles. The problem was to find a person having the needed product, business and cultural skills. The solution developed in this project was to match competence profiles of diaspora members against the competence profile of the company needing to find a person to help with business cooperation.

A competence profile in this case is the most complex amongst the presented project cases. A profile includes three major parts: general competences like ability to inform or plan, cultural competences, e.g. command of English or Swedish, and occupational competences, i.e. domain skills/knowledge. Each ability/skill is associated with a corresponding competence level. Competence profiles of each company were described in terms of economic activities, products and services produced, and skills of the company's personnel. All the competence profiles were created based on the ontological model of competences, which contained 429 classes totally [TSH06].

The profiles together with general company information were stored in a database available for search by Swedish and Vietnamese companies. When a potential partner company with the desirable profile was found, the second database was searched for a diaspora member having a competence profile matching the query defined as part of the found company's profile. The query represented a set of domain areas, cultural skills and other abilities needed to support business cooperation. If there were several competence profiles found, ranking was needed to order the found persons according to their abilities (and ability levels) to help with business relationships establishment. A competence profile may change when a person has acquired skills and knowledge through receiving training for a new job or improving command of a foreign language.

3.5 Requirements of Competence Profile Management and Operations on Competence Profiles

Each description of the project case explained how competence profiles were utilised in the cases. Based on this, we can formulate requirements of a competence profile management system, which are generalised and summarized in Table 1 (where 'MILOG' denotes 'Media-ILOG').

One can note that the first three requirements from each case are the same—they create a starting point for competence profile management. The requirement to represent a competence profile in a form that reflects complex semantics of the profile and allows for efficient data processing is intrinsic to every case. The last requirement from all the cases except for the first one is a precondition for integration of a competence profile management system into existing enterprise applications. Indeed, a competence profile management system cannot work in a standalone way in most cases. The rest of the requirements comprise the essence

Table 1: Requirements of a competence profile management system

N	Requirement	MILOG	BTG-QR	BTG-UR	BR-IComp
Functional: management of a single profile					
R1	Create a competence profile	√	√	√	√
R2	Add new competences to a profile	√	√	√	√
R3	Remove unneeded competences from a profile	√	√	√	√
R4	Extract a competence subprofile based on a constraint			√	
R5	Match a competence profile against a resource	√	√	√	
Functional: management of multiple profiles					
R6	Search for competence profiles based on a request				√
R7	Rank competence profiles based on a criteria				√
R8	Aggregate competences of a group of profiles	√			
R9	Monitor changes in competence profiles	√			
Non-functional					
R10	Represent competence profiles in a semantically rich form that allows for automated processing	√	√	√	√
R11	Interoperability with existing information systems on the data exchange level		√	√	√

of competence profile management—they are those ones that bring business value to an enterprise.

The elicited requirements imply necessity of construction of a competence profile as well as operations on competence profiles. The operations are intended to perform manipulations with competence profiles in order to support management of an enterprise. Hence, the required operations on competence profiles are explicitly identified in Table 2. The operations to be supported by a competence profile management system correspond to the functional requirement. Sect. 4.3 will provide detailed definitions of the operations.

4 Abstract Model of Competence Profile Management

This section constructs the abstract model of competence profile management based on the required operations (functional requirements) from Sect. 3.5. It starts with specification of enterprise context, continues with formal definitions of competence, competence level and competence profiles, and finishes with op-

Table 2: Required operations on competence profiles

N	Operation	Requirement
O1	Create a profile	R1
O2	Add competences to a profile	R2
O3	Remove competences from a profile	R3
O4	Extract a subprofile from an existing profile	R4
O5	Match profile against a resource, e.g. a data source	R5
O6	Search for profiles	R6
O7	Rank profiles based on criteria	R7
O8	Aggregate competences of a group of profiles	R8
O9	Monitor changes in aggregated profiles	R9

erations on competence profiles.

4.1 Enterprise Context

The context of an enterprise introduces initial concepts that underpin the definitions provided in subsections 4.2 and 4.3. These concepts originate from organisational behaviour studies and enterprise modelling languages (for example, see the organizational behavior classification and modelling framework [ASB10] and the Unified Enterprise Competence Modelling Language [PCFG07]). In our model we will distinguish between these elements of an *enterprise*:

- *Personnel* (workers) who are organized in a structured way. Each worker is assigned a *role(s)*.
- *Business processes*, which are composed of tasks. These *tasks* are carried out by the personnel in order to achieve some of the enterprise goals.
- *Resources*, which can be equipment, mechanical tools, data source and the like. Resources are used by the personnel to perform tasks.

Now let us consider the formal model of an enterprise context. Let $WK = \{wk_1, \ldots, wk_p\}$ be the set of workers (personnel) of an enterprise and $RL = \{rl_1, \ldots, rl_q\}$ be the set of roles from the organization structure of this enterprise. At any given moment a worker is assigned only one set of roles (*at least one role*) to act in. Let $rlf : WK \to \wp(RL)$ be the role function (where $\wp(RL)$ denotes the power set of RL), which maps a worker to all the roles assigned to this person, defined by:

$$rlf(wk_i) = \{rl_j \mid wk_i \text{ acts in } rl_j\} \tag{1}$$

Let $PR = \{Pr_1, \ldots, Pr_m\}$ be the set of all business processes of an enterprise. Let $TS^{Pr_i} = \{ts_1, \ldots, ts_n\}$ be the set of tasks belonging to one particular business process and $\preceq_{Pr_i}$ be a partial ordering on tasks that is specific to the process Pr_i. A business process consists of several tasks that are performed in a particular order. Hence a process can be defined as a poset[2]:

$$Pr_i = \left(TS^{Pr_i}, \preceq_{Pr_i}\right) \tag{2}$$

Then the set of all tasks for an enterprise can be defined as:

$$TS = \bigcup_{Pr_i \in PR} TS^{Pr_i} \tag{3}$$

We can note that this is a set cover of TS but not a set partition because one task can belong to several processes, e.g. blood test can be included in the processes of treatment for different kinds of patients.

At any given moment a role is assigned only one set of tasks (*at leas one task*) to carry out. Let $tsf : RL \rightarrow \wp(TS)$ be the task function, which maps a role to all the tasks assigned to this role, given by:

$$tsf(rl_i) = \{ts_j \mid rl_i \text{ performs } ts_j\} \tag{4}$$

Consequently, all the tasks being carried out by a worker can be given by the 'worker's tasks' function $wkt : WK \rightarrow \wp(TS)$:

$$wkt(wk_i) = \bigcup_{rl_j \in rlf(wk_i)} tsf(rl_j) \tag{5}$$

Let $RS = \{rs_1, \ldots, rs_t\}$ be the set of all resources of an enterprise. At any given moment a task is given only one set of resources (might be empty) to use[3]. Let $rsf : TS \rightarrow \wp(RS)$ be the resource function, which maps a task to all the resources assigned to this task, given by:

$$rsf(ts_i) = \{rs_j \mid ts_i \text{ uses } rs_j\} \tag{6}$$

Another function concerning resources that is useful is the 'resource's tasks' function $rst : RS \rightarrow \wp(TS)$, which maps a resource to the set of tasks that require this resource:

$$rst(rs_j) = \{ts_i \mid rs_j \in rsf(ts_i)\} \tag{7}$$

Let $\Sigma = \{rlf, tsf, wkt, rsf, rst\}$ be the set of all the functions associated with the context of an enterprise. Then the enterprise context can be specified as the following sextuple:

$$E = \langle WK, RL, PR, TS, RS, \Sigma \rangle \tag{8}$$

[2] Partially ordered set

[3] People are not considered resources in our model

This definition of an enterprise context can be applied to any of the project cases[4] from Sect. 3. For example, in BTG-UR there are roles (RL) of a urology specialist and urology nurse, a particular role being assigned to a healthcare worker (WK). The nurse role may imply several tasks (TS) like making urine tests or giving medicine. All the tasks belong to the process (PR) of patient treatment and some tasks may use equipment (RS), e.g for urine tests.

4.2 Competence, Competence Level and Competence Profile

Now that the initial concepts constituting an enterprise context have been defined, we can consider the notions of competence, competence level and competence profile. First we need to define competence level.

Definition 1. A *competence level* reflects the degree to which a person can possess a certain ability. Let $C = \{c_1, \ldots, c_u\}$ be the set of all competences and $L = \{l_1, \ldots, l_v\}$ be the set of all competence levels. Then, competence level can be associated with competence. Formally, the competence-level relation CL on $C \times L$ is given by:

$$CL = \{(c_i, l_j) \mid c_i \text{ is associated with } l_j\} \tag{9}$$

Some competences can have several levels associated to it, e.g. team coordinator role ability can be related to a general ability level as well as work experience level. One level can be used by several competences, e.g. average ability level. There are cases in which competence levels are not needed. If levels are not used, then any level of specific competence is enough to carry out the related task[5]. It should be noted that a zero level (or 'negative' one) is not considered at all because this means complete absence of competence.

Definition 2. A *competence* is an ability (or skill) at a certain *level* that is required to perform a task. Competence with associated levels is defined through its relation to tasks. Formally, the task-competence relation TC on $TS \times CL$ (where TS is specified in Eq. (3)) is given by:

$$TC = \{(ts_k, (c_i, l_j)) \mid ts_k \text{ requires } c_i \text{ at level } l_j\} \tag{10}$$

where l_j means the minimum level required to perform the task t_k.

For instance, to prepare a presentation for a meeting, a worker needs ability to inform (probably other abilities as well, the specification granularity depends on the domain). This definition is based on the one given by [BK98] (see Sect. 2.1

[4] Also see a note on the BR-IComp case in Sect. 6
[5] Levels that are insufficient are not considered in descriptions of roles and tasks.

for the details). The difference is that Def. 2 further specifies the notion of task through the enterprise context given by Eq. (8).

In general, carrying out one task may require many competences and the same competence may be required for several tasks. Let $tsc : TS \rightarrow \wp(CL)$ be the 'task's competences' function, which, maps a task to the set of competence with level required for this task, given by:

$$tsc(ts_k) = \{(c_i, l_j) \mid (ts_k, (c_i, l_j)) \in TC\} \tag{11}$$

Definition 3. A *competence profile* of a worker is the set of all competences with associated levels that are required to act in the role(s) assigned to this worker that is to perform all the tasks implied by these roles. Formally, a competence profile of the worker wk is the set given by:

$$Cp_{wk_i} = \bigcup_{ts_j \in wkt(wk_i)} tsc(ts_j) \tag{12}$$

where wkt is defined by Eq. (5).

This definition is close to the model of competence provided by [BC07], where competence is necessitated by an activity through business domains with related roles. However, Def. 3 constructs a set of competence differently—via decomposition of the assigned roles into tasks with required competences.

In this article we only consider competence profiles in an enterprise context, thus the notion of competence profile in other contexts may be different.[6] Additionally, a competence profile is constructed based on the description of roles assigned to the worker (not on assessment of the person), that is the competence profile represents required competences.

Let CP be the set of all competence profiles (one for each worker) given by:

$$CP = \bigcup_{wk_i \in WK} Cp_{wk_i} \tag{13}$$

Now we can introduce a convenience function mapping a competence profile of a worker to the set of tasks for which the required competences with corresponding levels are included in the profile. Formally, the 'competence profile's tasks' function $cpt : CP \rightarrow \wp(TS)$ is given by:

$$cpt(Cp_{wk_i}) = \{ts_j \mid tsc(ts_j) \subseteq Cp_{wk_i}\} \tag{14}$$

One more function will be useful—the one finding the competence profile of the given worker. Formally, the 'competence profile finder' function $cpf : WK \rightarrow CP$ (where WK is specified in Sect. 4.1) is given by:

$$cpf(wk_i) = Cp_{wk_i} \tag{15}$$

[6] For example, in general a competence profile may include skills needed for pursuing hobbies.

This function is both injective and surjective (bijective), which implies that it possesses an inverse function (cpf^{-1}).

An important constraint stemming from the domain is that the set of tasks for which the required competences are included in the competence profile of a worker should be the same as the set of tasks assigned to the worker through the roles, i.e. $(cpt \circ cpf)(wk_i) = wkt(wk_i)$. Now we can formally show that the abstract model satisfies this "tasks" constraint:

$$\begin{aligned}(cpt \circ cpf)(wk_i) &= cpt\left(\bigcup_{ts_j \in wkt(wk_i)} tsc(ts_j)\right) \\ &= \left\{ts_k \mid tsc(ts_k) \subseteq \bigcup_{ts_j \in wkt(wk_i)} tsc(ts_j)\right\} \\ &= \{ts_k \mid ts_k \in wkt(wk_i)\} \\ &= wkt(wk_i) \end{aligned} \tag{16}$$

4.3 Operations on Competence Profiles

Now we can define competence profile operations, which precisely describe the meaning of the required operations on competence profiles listed in Table 2. The numbering in this section corresponds to the numbering in Table 2.

Operation 1 Create competence profile. *Describe competences with associated levels that are needed to act in the roles assigned to a worker that is to perform all the tasks implied by the roles. Having definition of competence profile (see Def. 3), it is easy to define this operation. Formally, it is the function* $crt : WK \rightarrow CP$ *(where* WK *is defined in Sect. 4.1 and* CP *by Eq. (13)) given by:*

$$crt(wk_i) = \bigcup_{ts_j \in wkt(wk_i)} tsc(ts_j) \tag{17}$$

where wkt *is defined by Eq. (5) and* tsc *by Eq. (11).*

Operation 2 Add competences to a profile. *Add competences with associated levels to a profile needed to act in a role newly assigned to a worker that is to perform all the tasks implied by this role. Formally, the function* $add : CP \times RL \rightarrow CP$ *(where* RL *is defined in Set. 4.1), which maps a worker's competence profile and a role to a modified competence profile, is given by:*

$$add(Cp_{wk_i}, rl_k) = Cp_{wk_i} \cup \bigcup_{ts_j \in tsf(rl_k)} tsc(ts_j) \tag{18}$$

where tsf *is defined by Eq. (4).*

Operation 3 Remove competences from a profile. *Remove competences with associated levels from a profile required by the role that is now obsolete for a worker. When a role becomes obsolete for a certain worker, the role function* rlf *(given by Eq. (1)) is updated first. Then the function* $rem : CP \rightarrow CP$, *which maps a worker's competence profile to a modified competence profile, can be defined by:*

$$rem(Cp_{wk_i}) = Cp_{wk_i} \cap \bigcup_{ts_j \in wkt(wk_i)} tsc(ts_j) \tag{19}$$

Note that wkt returns here tasks for only the remaining roles because rlf is updated before carrying out this operation. Simple subtraction of the competences required to perform all the tasks implied by the obsolete role would not work in this case since the same competence can be required for both "obsolete" task and "active" task.

Operation 4 Extract competence subprofile. *Limit the profile of a worker to only those competences with associated levels that are needed to perform the given set of tasks. Formally, the function* $extr : CP \times \wp(TS) \rightarrow CP$ *(where* TS *is defined by Eq. (3)), which maps a worker's competence profile and a set of tasks to a subset of the worker's competence profile, is given by:*

$$extr(Cp_{wk_i}, TS') = Cp_{wk_i} \cap \bigcup_{ts_j \in TS'} tsc(ts_j) \tag{20}$$

Note that this operation does not guarantee that the extracted subprofile will contain all the competences needed to carry out the complete set TS. The result of $extr$ may even be $\emptyset$.

Operation 5 Match competence profile against a resource. *Compare the tasks, for performing which a resource is needed, to the tasks, for performing which competences (with associated levels) are included in the competence profile of a worker. Formally, the function* $match : CP \times RS \rightarrow [0, 1]$, *which maps the competence profile of a worker and a resource to a real number (similarity measure), is given by:*

$$match(Cp_{wk_i}, rs_j) = tsm(cpt(Cp_{wk_i}), rst(rs_j)) \tag{21}$$

where cpt is defined by Eq. (14) and tsm by Eq. (22) below.

The function tsm, which is used in $match$, is the 'tasks matching' function $tsm : \wp(TS) \times \wp(TS) \rightarrow [0, 1]$, which maps two sets of tasks to a real number (a score showing how similar the sets are):

$$tsm(T_1, T_2) = x \quad \text{where} \quad \begin{cases} x = 1 & \text{if} \quad T_1 \supseteq T_2 \\ x \in (0, 1) & \text{if} \quad T_1 \not\supseteq T_2 \ \wedge \ T_1 \cap T_2 \neq \emptyset \\ x = 0 & \text{if} \quad T_1 \cap T_2 = \emptyset \end{cases} \tag{22}$$

The exact definition of tsm depends on the domain and can be different. For some domains it will be enough just to test for full inclusion of the resources' tasks into the the competence profile's tasks that is tsm can return either 0 or 1. Other domains may require a finer-grained measure. Then tsm will return a value from $[0, 1]$. An example of a similarity measure based on semantic distance is given in [BBL07].

Operation 6 Search for competence profiles. *Find all competence profiles that include competences with associated levels needed for performing the tasks specified in a request that is find all competence profiles that "support" carrying out the specified tasks. Formally, the function* $srch : \wp(TS) \to \wp(CP)$, *which maps a set of tasks to a set of competence profiles that include competences required for the tasks, is given by:*

$$srch(TS') = \{Cp_i \mid tsm(cpt(Cp_i), TS') \geq z\} \tag{23}$$

where z *denotes a threshold, which is specific for a domain, and* $z \in [0, 1]$.

Operation 7 Rank competence profiles. *Order a set of competence profiles according to the given criterion. Let* $CR = \{cr_1, \ldots, cr_s\}$ *be the set of all ranking criteria. We can assume that a ranking criterion can be transformed into a partial ordering* $\preceq_{cr_i}$ *on a set of competence profiles. Then the function* $rank : \wp(CP) \times CR \to \wp(CP)$, *which maps a set of competence profiles and a ranking criteria to the corresponding poset, is formally given by:*

$$rank(CP', cr_i) = (CP', \preceq_{cr_i}) \tag{24}$$

A criterion used in $rank$ can be different, e.g. it can concern certain ability to perform tasks or it can reflect how well a competence profile matches a particular task or resource. The input set of profiles for the $rank$ operation can be, for example, the one found by $srch$.

Operation 8 Aggregate competences of a group of profiles. *Show all tasks that can be performed using competences (with associated levels) from the given competence profiles. Formally, this is the function* $aggr : \wp(CP) \to \wp(TS)$ *given by:*

$$aggr(CP') = \bigcup_{Cp_i \in CP'} cpt(Cp_i) \tag{25}$$

These aggregated tasks can show 'the collective competence' of an enterprise at the moment. They also reflect the focus of the current activities of an enterprise.

Operation 9 Monitor changes in competence profiles. *Show changes in the tasks that can be performed using competences (with associated levels) from all* *the competence profiles. The changes are to be shown at certain points in time. Let* $(CP_{t_0}, CP_{t_1}, \ldots, CP_{t_n})$ *be a sequence of sets of all competence profiles (for all workers from the enterprise context given by Eq. (8)) changing with time and* $\mathcal{CP}$ *be the set of all such possible sequences. Let* $(TS_{t_0}, TS_{t_1}, \ldots, TS_{t_n})$ *be a sequence of sets of all tasks (from the enterprise context given by Eq. (8)) changing with time and* $\mathcal{TS}$ *be the set of all such possible sequences. Then the function* $chng : \mathcal{CP} \rightarrow \mathcal{TS}$*, which maps a sequence of sets of competence profiles to a sequence of sets of tasks, is defined by:*

$$\begin{aligned} chng((CP_{t_0}, CP_{t_1}, \ldots, CP_{t_n})) &= (aggr(CP_{t_0}), aggr(CP_{t_1}), \ldots, aggr(CP_{t_n})) \\ &= (TS_{t_0}, TS_{t_1}, \ldots, TS_{t_n}) \end{aligned} \tag{26}$$

These changes can reflect the direction in which the enterprise activities are shifting.

5 Ontology-based Implementation of the Abstract Model of Competence Profile Management

The abstract model described in Sect. 4 provides formal definitions of competence and competence profile operations, which correspond to the required operations (functional requirements R1–R9) described in Sect. 3.5. In order to build a competence profile management system, the mathematical constructs of the abstract model need to be implemented. The choice of a means of implementation is based on the two non-functional requirements (see Table 1). Requirement R10 states that competence profiles should be represented in a semantically rich way that provides for automated processing. The Web Ontology Language (OWL) [OWL09] as an ontology language allows for description of the concept of competence and its relationships to other concepts because an ontology is "*an explicit specification of a conceptualization*" [Gru93]. The mathematical constructs of the abstract model map very well to the OWL constructs. Complex semantics of competence profiles can be expressed with OWL thanks to restrictions on classes, object property characteristics and reasoning capabilities. On the other hand, OWL allows for construction of ontologies (knowledge bases) that can be directly used in a software system. The second non-functional requirement (R11) calls for interoperable data exchange between different system. One of the most common formats for data exchange is XML nowadays. The OWL serialization is build upon XML and tools working with OWL ontologies provide for export/import of data via XML-OWL or OWL-XML transformations.

Fig. 1 depicts the layered architecture of the ontology-based implementation of the the abstract model. The cornerstone is an OWL DL ontology schema (classes with restrictions on them and object properties with their character-

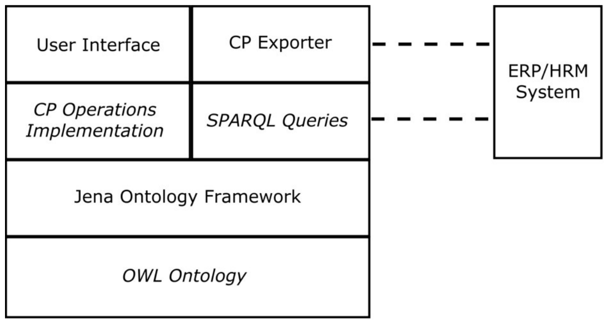

Figure 1: The architecture of the implementation of the abstract model

istics)[7]. The sets are implemented with classes, while the relations with both classes and object properties from the schema. The Jena ontology framework (`http://www.openjena.org`) provides an API for managing the OWL ontology and executing SPARQL (`http://www.w3.org/TR/rdf-sparql-query/`) queries. The functions require classes and object properties as well as SPARQL queries, which are executed using Jena. The queries retrieve data on organisational structure and business processes from the ERP/HRM system of the enterprise. The operations on competence profiles (CP) are implemented with Java code that relies on the Jena functionality to manipulate the ontology elements. The result of carrying out CP operations can be either directly displayed to the user or exported to the Enterprise Resource Planning (ERP) or Human Resource Management (HRM) system for further use. The interoperability with the latter can be achieved, e.g. by using HR-XML [HR-11]. The following subsections detail the OWL schema, SPARQL queries and operations implementation. The other details of the software part of the implementation are not presented because they are out of the scope of this paper.

5.1 Ontology Schema and SPARQL Queries

Overview of the ontology schema is shown in Fig. 2 in a graphical form. The full schema is specified in OWL DL but for brevity only the overview is presented here. Fig. 2 shows classes denoted by ellipses and object properties denoted by arcs. Arrows show the direction of relationships along the object properties. Each object property includes cardinality shown with either a number or the asterisk character (for "many"). The `hasCP` and `nextTask` object properties are functional. The latter is also reflexive. All the object properties except for `nextTask`

[7] The schema provides constructs that are applicable to competence profile management in any domain. For a particular domain a specialisation of the schema may be created to take into account the domain details, e.g. structuring of competences.

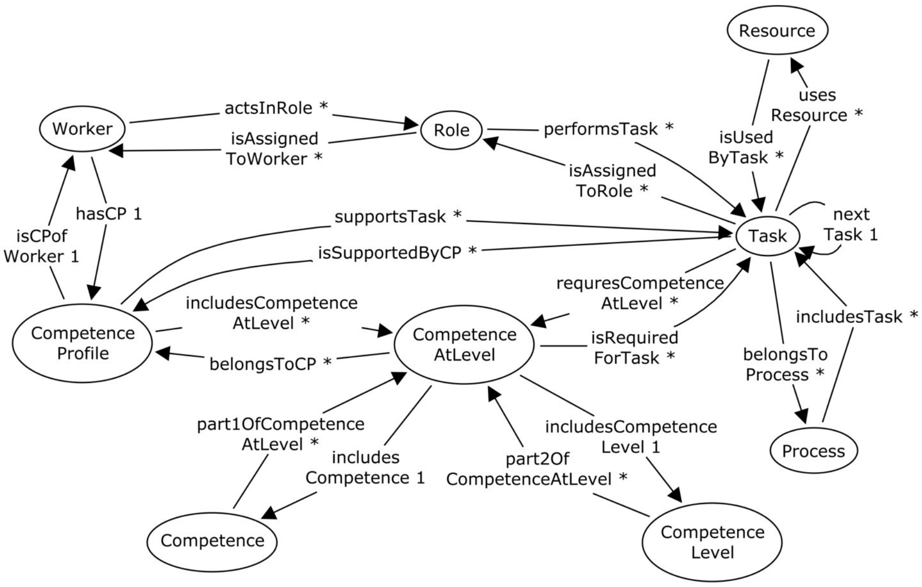

Figure 2: The classes and object properties included in the ontology schema

have inverses—this is depicted by arrows going into the opposite directions. When the OWL ontology is populated for a specific domain, individuals for each class are created and interlinked with object properties by either a knowledge engineer or the operations implementation code. At the same time, links between these individuals through the inverses are inferred by an OWL reasoner. Definition of each class except for `Resource` includes restrictions, e.g. the definition of `Task` includes the following restrictions (expressed in Turtle):

```
rdfs:subClassOf [ a owl:Restriction ;
  owl:onProperty :nextTask ; owl:someValuesFrom :Task ] ;
rdfs:subClassOf [ a owl:Restriction ;
  owl:onProperty :usesResource ; owl:someValuesFrom :Resorce ] ;
rdfs:subClassOf [ a owl:Restriction ; owl:onProperty
  :requiresCompetenceAtLevel ; owl:someValuesFrom :CompetenceAtLevel ] .
```

Now we will consider those classes that correspond to the sets from the enterprise context E (given by Eq. (8)). The `Worker` class implements the WK set, `Role` implements the RL set, `Process` implements the PR set, `Task` implements the TS set, and `Resource` implements the RS set. A business process Pr_i (given by Eq. (2)) is implemented as depicted in Fig. 3 (where a rectangle denotes an individual). Let us assume that the business process `Process1` (Pr_i), which is an individual of the `Process` class, consists of the tasks `Task1`, `Task2` and `Task3`, which are individuals of the `Process1Task` class (corresponding to

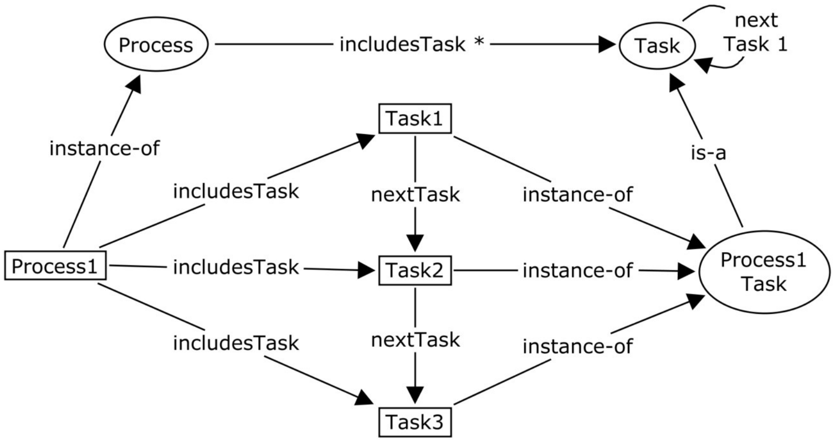

Figure 3: The fragment of the ontology showing processes and tasks

TS^{Pr_i}). Then the tasks are ordered using the `nextTask` object property, which corresponds to a partial ordering $\preceq_{Pr_i}$. The `Process1Task` is a subclass of `Task` class, the latter corresponding to the definition of the set TS (given by Eq. (3)).

The functions from the enterprise context E are implemented with SPARQL queries as detailed in Table 3. The queries retrieve data from the ERP/HRM system either directly as an RDF graph, e.g. using the DR2RQ Platform (`http://d2rq.org/`), or in off-line mode, e.g. through HR-XML to OWL transformation. Each function is realised with a SPARQL query, which is constructed using the classes and object properties from the ontology schema. The meaning of the use of SPARQL queries is that if an argument of a function is given, the query returns an element or a set of elements, which correspond to the value of the function at the given argument[8]. The `cpm` prefix used before the object properties denotes the URI of the namespace of the OWL ontology containing the ontology schema.

Now that we detailed the implementation of the enterprise context elements, we can show how the definitions of competence (Def. 2), competence level (Def. 1) and competence profile (Def. 3) are realised. The *competence-level* relation CL (given by Eq. (9)) is implemented with the `CompetenceAtLevel` class that links `Competence` and `CompetenceLevel` through the object properties `includesCompetence` and `includesCompetenceLevel`.[9] The *task-competence* relation TC

[8] A name that is preceded by the question mark denotes a variable. A name that is preceded by a semicolon denotes a particular individual and is supposed to be substituted by the individual ID from an ontology when executing the query.

[9] Note that `part1OfCompetenceAtLevel` is *not* the inverse of `includesCompetence` and `part2OfCompetenceAtLevel` is *not* the inverse of `includesCompetenceLevel`.

Table 3: The implementation of the enterprise context functions

Function	Classes used	Object properties used	SPARQL query
Role function rlf (given by Eq. (1))	Worker, Role	actsInRole	SELECT ?rl WHERE { :wk cpm:actsInRole ?rl }
Task function tsf (given by Eq. (4))	Role, Task	performsTask	SELECT ?ts WHERE { :rl cpm:performsTask ?ts }
'Worker's tasks' wkt (given by Eq. (5))	Worker, Role, Task	actsInRole, performsTask	SELECT ?ts WHERE { :wk cpm:actsInRole ?rl. ?rl cpm:performsTask ?ts }
Resource function rsf (given by Eq. (6))	Task, Resource	usesResource	SELECT ?rs WHERE { :ts cpm:usesResource ?rs }
'Resource's tasks' rst (given by Eq. (7))	Task, Resource	usesResource	SELECT ?ts WHERE { ?ts cpm:usesResource :rs }

(given by Eq. (10)) is implemented with the classes `Task` and `CompetenceAtLevel` and the object property `requiresCompetenceAtLevel`. The set of competence profiles CP (given by Eq. (13)) is realised with the `CompetenceProfile` class. The implementation of the *competence profile* definition Cp_{wk_i} (given by Eq. (12)) requires the most classes and object properties from Fig. 2 (except for the classes `Process` and `Resource`). A `Worker` individual is linked to his/her competence profile through the `hasCP` object property. The `CompetenceProfile` individual is linked to particular competences with levels through `includesCompetence-AtLevel` object property. The rest of the classes and object properties is used by the functions *wkt* and *tsc* during the initial construction of a competence profile (see Procedure 1).

The implementation of the functions related to the definitions of competence and competence profile is detailed in Table 4. Each function is again realised with a SPARQL query, which retrieves data from the OWL ontology (except for *tsc* that queries data in the ERP/HRM system). The implementation of the 'Worker finder' function cpf^{-1} uses an inverse object property, which requires the SPARQL query to be run against an inferred ontological model. The 'competence profile's tasks' function *cpt* is implemented using the `supportsTask` object property having the meaning that a task can be carried out using the competences from the given competence profile.[10]

[10] One can write a query using the object properties `requiresCompetenceAtLevel` and `belongsToCP` but it will just find all subgraphs connecting a task with the given profile through a competence. There will be no guarantee that all the competences required for a particular task are included in the profile.

Table 4: The implementation of the competence profile related functions

Function	Classes used	Object properties used	SPARQL query
'Task's competences' function *tsc* (given by Eq. (11))	Task, Competence	requires-Competence	SELECT ?cl WHERE { :ts cpm:requiresCompetenceAt-Level ?cl }
'Competence profile's tasks' function *cpt* (given by Eq. (14))	Competence-Profile, Task	supportsTask	SELECT ?ts WHERE { :Cp cpm:supportsTask ?ts }
'Competence profile finder' function *cpf* (given by Eq. (15))	Worker, Competence-Profile	hasCP	SELECT ?Cp WHERE { :wk cpm:hasCP ?Cp }
'Worker finder' function, i.e. cpf^{-1} (see page 13)	Competence-Profile, Worker	isCPof-Worker	SELECT ?wk WHERE { :Cp cpm:isCPofWorker ?wk }

Finally, we need to implement the "tasks" constraint (given by Eq. (16)), which can be used to periodically check the integrity of the OWL ontology. It is realised with two SPARQL queries as follows meaning that the result sets retuned by the queries have to be equal (which is checked by the Java code):

```
SELECT ?ts WHERE { :wk cpm:hasCP ?Cp.      ?Cp cpm:supportsTask ?ts } =
SELECT ?ts WHERE { :wk cpm:actsInRole ?rl. ?rl cpm:performsTask ?ts }
```

5.2 Implementation of the Operations on Competence Profiles

Now we can proceed with realisation of the operations on competence profiles from Sect. 4.3. Each operation is implemented as code in Java because we chose Jena for ontology management. The code uses the Jena functionality and the functions implementations described in Tables 3 and 4 to perform manipulations with the OWL ontology that result in adding/removing elements to/from the ontology or interrogating the data in it. Below an example is given of operation implementation. The remaining implementations are omitted for brevity.

Procedure 1 *This procedure implements the operation for creating competence profile using the assigned roles (see Oper. 1).*

```
public void createCP(String workerStr) {
    Individual cp = cpModel.createIndividual(workerStr + "_CP",
        CPMONTO.CompetenceProfile); // create an empty profile
    Individual worker = cpModel.getIndividual(workerStr);
    cpModel.add(worker, CPMONTO.hasCP, cp);
    List<RDFNode> tasks = func.workersTasks(workerStr); // wkt() func
    for (RDFNode ts : tasks) {
```

```
        List<RDFNode> competencesAtLevel = func.tasksCompetences(
            ts.toString()); // tsc() function
        for (RDFNode cl : competencesAtLevel) {
            cpModel.add(cp, CPMONTO.includesCompetenceAtLevel, cl);
        }
    }
    findSupportedTasks(cp); // add relations "supportsTask"
}
```

This procedure creates a competence profile for a worker and adds relevant competences to it. It also links the competence profile to the tasks, which can be carried out using the competences from the profile, through the `supportsTask` object property. This is needed for the implementation of the 'competence profile's tasks' function *cpt* (see Table 4). Although the ontology schema shown in Fig. 2 includes inverses for the object properties, we do not need to explicitly link the found individuals with the inverse. These facts will be automatically inferred by a reasoner.

6 Using the Ontology-based Implementation in a Case

This section presents a short example of using the ontology-based implementation in the BR-IComp case (see Sect. 3.4).[11] The first step was to import the OWL ontology schema (from Sect. 5.1) into a new BR-IComp ontology: `<owl:imports rdf:resource="cpm-cl_ontology"/>`. Then the ontology was specialized with subclasses and subproperties as well as populated with individuals representing the specifics of the domain. The enterprise context was not initially represented in the BR-IComp case so we introduced the main business process—establishment of business relationship, which is modelled as the `BusinessRelationshipProcess` individual of the `Process` class. This process was described in the BR-IComp project report [JJH07] and now it is presented in Table 5 in a more detailed manner. Diaspora members do not initially belong to the staff of a company. However, when a company needs to overcome cultural and language differences, a member of the Vietnamese diaspora in Sweden is employed as a worker. He/she plays the role of business relationship mediator modelled as the `BusinessMediatorRole` individual of the class `Role`. This role implies carrying out tasks 2, 3, 5, 7, and 10 (see Table 5) that are individuals of of the class `BusinessRelationshipProcessTask` (that is a subclass of `Task`).

As soon as competences with levels were already specified in OWL in this case, they were simply transferred to the new ontology. Each task assigned to the business mediator role requires several competences. An example of a task with required competences is given below in Turtle (for better readability):

[11] The BR-IComp ontology presented in [TSH06] was constructed before creation of the approach for competence profile management. Hence, the ontology was completely reworked according to the new approach.

Table 5: The process of establishment of business relationships

N	Task description
1	Search for a possible business partner in Vietnam
2	Make initial contact with the found Vietnamese company (potential partner)
3	Specify in detail the product demand
4	Test the pilot delivery that is sent to check the quality of the product
5	Report the results of testing the quality of the pilot delivery
6	Prepare a contract on regular deliveries of the product
7	Finalize the contract on regular deliveries of the product taking into account business practice in Vietnam
8	Prepare delivery documents in order for the deliveries to be received into the country according to the regulations and without complications
9	Check the quality of the delivered products
10	Handle complaints, which may include the return of unsatisfactory products
11	Make secure payment for the delivered products

```
:DeliveryTestReportTask
  a  :BusinessRelationshipProcessTask ;
  cpm:nextTask :ContractPreparationTask ;
  cpm:requiresCompetenceAtLevel
    :CommitmentAbility-AverageLevel ,   :InformingAbility-AverageLevel ,
    :PlanningAbility-AverageLevel ,     :TeamFormingAbility-AverageLevel,
    :InformalKnowledgeOfVietnam-Grown , :InformalKnowledgeOfSweden-Lived,
    :InterculturalSensitivity-Acceptance ,
    :VietnameseLang-NativeLevel ,       :SwedishLang-BeginnerLevel .
```

All the competences in the BR-IComp case are grouped into the following classes: `GeneralCompetence` (the first four ones in the example), `CulturalCompetence` (the rest in the example, the two last ones being language competence), and `OccupationalCompetence`. The last type is domain-dependent and modelled as part of describing separate business processes, e.g. a car repair process. The division of competences into these groups reflects some of the different perspectives on the competence described in Sect. 2.1.

After specializing and populating the ontology, the Java code was run. It loaded the OWL ontology, attached a Jena reasoner (optimized rule-based reasoner with OWL rules) and then executed Procedure 1 to create a competence profile for a diaspora worker. A fragment of the created profile is depicted in Fig. 4, where the `cpm` prefix in the object properties denotes the URI of the namespace of the OWL ontology schema. `DiasporaWorker_C` is an individual of the class `DiasporaWorker` that is a subclass of `Worker`. The profile includes the Vietnamese language competence at native level. The competence is an individual of the `LanguageCompetence` class, which is a subclass

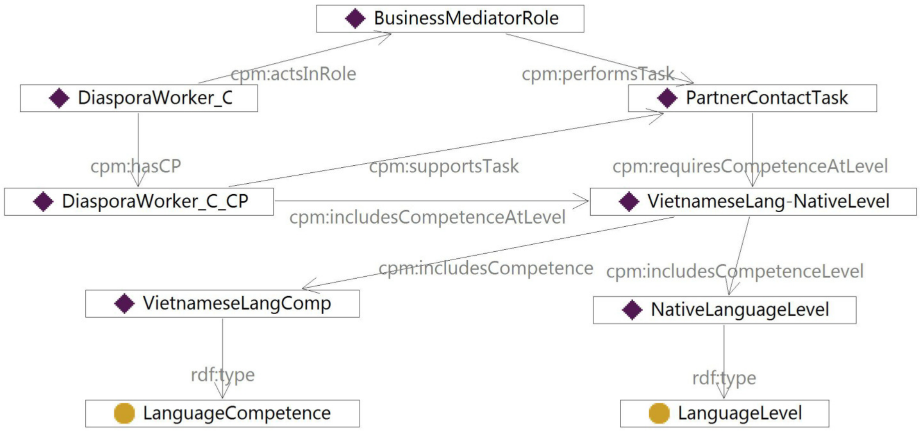

Figure 4: A fragment of the created competence profile

of `CulturalCompetence` (that is in turn a subclass of `Competence`), while the level is an individual of the `LanguageLevel` class, which is a subclass of `CompetenceLevel`. The competence profile `DiasporaWorker_C_CP` is also linked to `PartnerContactTask` through the `supportsTask` object property.

Finally, Oper. 6 was carried out to search for profiles able to perform partner search (task 1 rom Table 5). As a result the `DiasporaWorker_C_CP` profile was found (it should be noted that task 1 is not assigned to `BusinessMediatorRole`).

7 Conclusions

This article has proposed formalisation and implementation of competence profile operations in an enterprise context. Four project cases were described and the use of competence profiles in these cases was analysed leading to establishment of requirements for competence profile management and operations on competence profiles. To formally define competence profile and operations on it, an abstract model was created with the help of discrete structures, namely sets, relations and functions. The ontology-based implementation of these definitions was provided consisting of an OWL ontology schema, SPARQL queries and Java code. An architecture was given demonstrating different layers of the implementation and interconnection with EPR/HRM systems of the enterprise. As a result, the elicited functional and non-functional requirements were satisfied. Finally, an example of using the implementation was given for one of the project cases.

The main contribution of this work is twofold: the abstract model that formalises the operations on competence profiles and the ontology-based implementation of it. Firstly, the abstract model provides exact specification of the semantics of the definition of competence profile and the identified operations on profiles. This formal specification addresses the lack of research into structures and utilisation of competence profiles in a competence management system that is how profiles can be managed as separate entities. The proposed definitions and operations are based on the analysis of real-world project cases and grounded in an enterprise context, that is in business processes and roles. Connecting processes, roles or tasks to competences has been inspected in many works (e.g. [BC07], [Gra10], [PCFG07]), however operations on competence profiles have not been explicitly addressed. The discrete structures used for the formalisation in the abstract model can be directly transformed into corresponding computer representations. The abstract model allows for different implementations. Although the ontology-based implementation presented in this article reflects the frequent use of ontologies in competence management, other implementations are possible, e.g. the use of ER-model and SQL queries. Secondly, the ontology-based implementation is carried out in OWL DL, which makes it possible to directly use the implementation in a software system. The implementation architecture can support development of competence management system able to perform the proposed competence profile operations. This differentiates this approach from other ones that mainly focus on formal specification and implementation of competence models and profiles (e.g. [BH07], [InL12], [PCFG07]). As soon as the utilized Semantic Web tools support transformation from OWL to other XML-based data formats and vice versa, data interoperability is increased with existing enterprise information systems, e.g. ERP/HRM systems. The importance of interoperability is reflected in the InLOC project [InL12].

However, this study has limitations. The specification of competence profile definition and operations on competence profiles is based on four project cases only. Hence, no statement of applicability of this approach to other cases can be made. More domains should be studied to verify and complete the set of competence profiles operations. Although the abstract model and its implementation were validated in one case (BR-IComp), the experimentation was limited. Additional experimentation in all the project cases is needed to validate the proposed approach in practical situations. Moreover, the proposed approach focuses on required competences only, while other approaches consider competences acquired by a worker as well (e.g. [BH07], [Gra10], [InL12]). Team competence is not examined either, though it can be important for cases like creation of project teams (e.g. [CNS+07], [JV11]).

Future work can be organised along both theoretical and practical lines. It is interesting to investigate how the abstract model can be extended to a multi-domain algebra on competence profiles. Another interesting question is to study how competence profiles can be ranked or mapped against resources.

Profile ranking can be implemented with inference rules, e.g. written in SWRL (Semantic Web Rule Language). This is particularly important for practical applications of competence profiles management. Competence gap analysis is another area where the presented approach to competence profiles management can be applied. Finally, the approach should be tested in other cases and used in full-fledged computer applications.

Acknowledgments

The author would like to thank Prof Kurt Sandkuhl for his support and fruitful discussions during the work on this article, and Dr Ulf Seigerroth for his encouragement. The author is also grateful to Dr Evgeny Ivashko, Dr Andrew Krizhanovsky and Dr Christer Thörn for helpful comments on different versions of the text, as well as to the reviewers for suggesting many improvements.

Some parts of the presented research were financed by Swedish International Development Agency (SIDA) in the project "ICT-support for formation of business relationships with developing countries", by Hamrin Foundation (Carl-Olof och Jenz Hamrins Stiftelse) in the project "Media Information Logistics" and by the Vinnvård research program financed by Vinnova and the Vårdal Foundation together with the Swedish Association of Local Authorities and Regions in the project "Bridging the Gaps".

References

[AJ09] Ayub, M. and Jawad, M.: Structuring and modelling competences in the healthcare area with the help of ontologies; Master's thesis, School of Engineering of Jönköping University, 2009.

[ASB10] Aitken, C., Stephenson, C., and Brinkworth, R.: Process classification frameworks; In vom Brocke, J. and Rosemann, M., editors, *Handbook on Business Process Management 2*, International Handbooks on Information Systems, pages 73–92. Springer Berlin Heidelberg, 2010.

[BBL07] Billig, A., Blomqvist, E., and Lin, F.: Semantic matching based on enterprise ontologies; In Meersman, R. and Tari, Z., editors, *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, volume 4803 of *Lecture Notes in Computer Science*, pages 1161–1168. Springer Berlin / Heidelberg, 2007.

[BC07] Bennour, M. and Crestani, D.: Using competencies in performance estimation: From the activity to the process; *Computers in Industry*, 58(2):151–163, 2007.

[BH07] Berio, G. and Harzallah, M.: Towards an integrating architecture for competence management; *Computers in Industry*, 58(2):199–209, 2007.

[BK98] Bjurklo, M. and Kardemark, G.: Social tests in a model for controlling the enhancement of competence; *Journal of Human Resource Costing & Accounting*, 3(2):51–64, 1998.

[CEN10] CEN Workshop on ICT Skills: *European e-Competence Framework 2.0*, 2010 `http://www.ecompetences.eu`. Accessed 23 October 2012.

[CNS+07] Colucci, S., Noia, T. D., Sciascio, E. D., Donini, F., and Ragone, A.: Semantic-based skill management for automated task assignment and courseware composition; *Journal of Universal Computer Science*, 13(9):1184–1212, 2007.

[GLT10] Gäre, K., Lindholm, E., and Tarasov, V.: Efficient tools for design of information supply in microsystems; Presented at the Microsystems in Healthcare Seminar, 2-4 March 2010, Qulturum, Jönköping, March 2010.

[Gra10] Grant, S.: The logic of competence;, 2010 `http://blogs.cetis.ac.uk/asimong/2010/11/19/the-logic-of-competence/`. Accessed 23 October 2012.

[Gru93] Gruber, T. R.: A translation approach to portable ontology specifications; *Knowledge Acquisition*, 5(2):199–220, 1993.

[GS11] Grant, S. and Sgouropoulou, C.: What is a level of competence?; In Stracke, C. M., editor, *Proceedings of the 5th European Conference on Competence Modelling for European HR and Policies: Bridging Business, Education, and Training (COME-HR)*, 2011.

[HBW03] Hammer, M. R., Bennett, M. J., and Wiseman, R.: Measuring intercultural sensitivity: The intercultural development inventory; *International Journal of Intercultural Relations*, 27(4):421–443, 2003.

[HR-11] HR-XML Consortium: HR-XML 3.2 release;, 2011 `http://www.hr-xml.org/?32StandardsInfo`. Accessed 23 October 2012.

[IJ12] Ismail, M. and Jan, A.: Context-based supply of documents in a healthcare process; Master's thesis, School of Engineering of Jönköping University, 2012.

[InL12] The InLOC information model;, 2012 `http://wiki.teria.no/display/inloc/Information+Model`. Accessed 23 October 2012.

[JJH07] JTH, JIBS, and HTU: ICT-support for formation of business relationships with developing countries based on immigrant competence. Phase 2; Technical report, Jönköping University, March 2007.

[JMP07] Jussupova-Mariethoz, Y. and Probst, A.-R.: Business concepts ontology for an enterprise performance and competences monitoring; *Computers in Industry*, 58(2):118–129, 2007.

[JV11] Janev, V. and Vraneš, S.: Ontology-based competency management: the case study of the Mihajlo Pupin Institute; *Journal of Universal Computer Science*, 17(7):1089–1108, apr 2011.

[LCD05] Liu, P., Curson, J., and Dew, P.: Use of RDF for expertise matching within academia; *Knowledge and Information Systems*, 8(1):103–130, 2005.

[OWL09] OWL Working Group: *OWL 2 Web Ontology Language* W3C, 2009 `http://www.w3.org/TR/owl2-overview/`. Accessed 23 October 2012.

[Paq07] Paquette, G.: An ontology and a software framework for competency modeling and management; *Educational Technology & Society*, 10(3):1–21, 2007.

[PCFG07] Pépiot, G., Cheikhrouhou, N., Fürbringer, J.-M., and Glardon, R.: UECML: Unified enterprise competence modelling language; *Computers in Industry*, 58(2):130–142, 2007.

[RM09] Różewski, P. and Małachowski, B.: Competence management in knowledge-based organisation: Case study based on higher education organisation; In Karagiannis, D. and Jin, Z., editors, *Knowledge Science, Engineering and Management*, volume 5914 of *Lecture Notes in Computer Science*, pages 358–369. Springer Berlin / Heidelberg, 2009.

[SÖS+07] Sandkuhl, K., Öhgren, A., Smirnov, A., Shilov, N., and Kashevnik, A.: Ontology construction in practice: Experiences and recommendations from industrial cases; In Cardoso, J., Cordeiro, J., and Filipe, J., editors, *Proceedings of the 9th International Conference on Enterprise Information Systems*, pages 250–256, 2007.

[TSH06] Tarasov, V., Sandkuhl, K., and Henoch, B.: Using ontologies for representation of individual and enterprise competence models; In Bellot, P., Duong, V., and Bui, M., editors, *Proceedings of the 4th IEEE International Conference on Computer Sciences Research, Innovation and Vision for the Future*, pages 205–212. IEEE, February 2006.