A semantic web architecture for integrating competence management and learning paths

Fotis Draganidis, Paraskevi Chamopoulou and Gregoris Mentzas

Abstract

Purpose – The purpose of this paper is to present a prototype ontology-based application that has been developed for competency management and learning paths.

Design/methodology/approach – The paper provides an overview of competency management and related work in this area, a description of the competency ontology, and a functional and architectural analysis.

Findings – The paper provides information on work related to ontology-based competency management systems, indicating an enhanced approach with a detailed analysis of system architecture and functional analysis.

Research limitations/implications – The proposed application will be implemented through a .NET deployment, in Microsoft Hellas, the Greek subsidiary of the multinational IT company.

Originality/value – Ontologies have already been created in different scientific areas, including knowledge and competency management. However, only a few ontology-based applications are available today within the domain of competency management. In this paper an ontology-based application is presented has been developed for competency management and learning paths. Specifically, the paper provides an overview of competency management and related work in this area, a description of the competency ontology, and a functional and architectural analysis.

Keywords Competences, Human resource management

Paper type Research paper

1. Introduction

Competency-based management has become a very crucial element in the effective operation of an enterprise or an organization, due to the increased need of the latter to be agile enough to adapt to quick market changes and reorientation of their business plans. In this situation, competency management systems (CMS) become the core human resource tool that enables the enterprise to manage and develop the skills of their employees, recruit the most appropriate candidates, and make effective succession planning and employee development plans.

Apart from enterprise competency management systems, research is being conducted on the development of ontology-based CMS which can provide possibilities such as the easy integration and mapping of different competency ontologies. Moreover, research efforts have been realized in the development of ontological e-learning systems. However, very few – if any – systems exist which integrate e-learning functionality with an ontological CMS. The mapping of employee or departmental/organizational skill gap analysis with the appropriate learning objects is crucial in order to develop the correct learning paths and consequently the appropriate competencies of employees or organizations.

This paper focuses on the description of an ontology based competency management system, which also integrates e-learning functionality in order to address this issue. The
interested reader can find an overview of the key concepts in competency-based management, as well as a description of the competency ontology, the functional and technical architecture of the system. The practical experience of the authors derives from the deployment of a Microsoft.NET version of the described system, in Microsoft Hellas, the Greek subsidiary of the leading IT enterprise Microsoft Corporation.

In the next section, we provide a brief history of competency management, a definition of the term and a description of its core elements. In section 3, we describe the research efforts conducted in ontological CMS and ontological e-learning systems. In section 4, we describe the competency ontology, while in the next two sections we analyze the functional and technical architecture of the system. Finally, conclusions and possible topics for further research are presented.

2. Competency-based management

The competency approach to human resources management is not new. The early Romans practiced a form of competency profiling in attempts to detail the attributes of a “good Roman soldier”. The introduction of competency-based approaches came in around 1970 and their development since then has been rapid. The distinguished Harvard psychologist David McClelland is credited with introducing the idea of “competency” into the human resource literature, in his efforts to assist the United States Information Agency improve its selection procedures. The latter argued that traditional intelligence tests, as well as proxies such as scholastic grades, failed to predict job performance. McClelland’s counter argument to the growing dissatisfaction with intelligence testing and the traditional job analytic approaches to personnel selection was the proposal to test for competency. As a case study he proposed the selection of Foreign Service Information Officers (McClelland, 2003). In his research, McClelland found that competencies such as interpersonal sensitivity, cross-cultural positive regards and management skills differentiated superior from average Information Officers.

Throughout the years competency based approaches have proved to be a critical tool in many organizational functions, such as workforce and succession planning, performance appraisal etc. The main reasons for selecting these approaches are the following:

- they can provide identification of the skills, knowledge, behaviours and capabilities needed to meet current and future personnel selection needs, in alignment with the differentiations in strategies and organizational priorities; and
- they can focus the individual and group development plans to eliminate the gap between the competencies requested by a project, job role, or enterprise strategy and those available.

According to the HR-XML Consortium Competencies Schema (see http://ns.hr-xml.org/2_0/HR-XML-2_0/CPO/Competencies.pdf), a competency can be defined as “A specific, identifiable, definable, and measurable knowledge, skill, ability and/or other deployment-related characteristic (e.g. attitude, behaviour, physical ability) which a human resource may possess and which is necessary for, or material to, the performance of an activity within a specific business context”.

“Throughout the years competency based approaches have proved to be a critical tool in many organizational functions, such as workforce and succession planning, and performance appraisal.”
A further analysis of the concept of “competency” brought us to the conclusion that, typically, a competency is defined in terms of:

- **category** – a group that homogeneous/similar competencies belong to;
- **competency** – a descriptive name for the specific competency;
- **definition** – statement(s) that explains the basic concept of this competency; and
- **demonstrated behaviour** – behaviour indicators an individual should demonstrate if the specified competency is possessed.

Table I depicts an example of a competency’s definition in terms of category, competency, definition and demonstrated behaviour. The general category of the competency is “people management competencies”, which amongst others can include the competencies of “building a team’s spirit” and “developing people”. Each competency has a definition and corresponding demonstrated behaviours, which are equal to the “proficiency levels” that are presented later in this paper.

### Table I  
An example of a competency’s definition  

<table>
<thead>
<tr>
<th>Category</th>
<th>Competency</th>
<th>Definition</th>
<th>Demonstrated Behaviour</th>
</tr>
</thead>
</table>
| People management competencies  | Building team spirit  | Provide team members with the excitement and desire to cooperate with each other, contributing to common goals | 1. Encourages help and respect to other team members  
2. Creates a common mission and a feeling of belonging to a team which aims at that Provide mentoring and experience transfer |
|                                 | Developing people     | Help team members to reach their potential in personal development         | 2. Provide feedback on strengths and weaknesses of the team members                      |

3. Related work

The use of ontologies in a competency-driven e-learning system is a research area that has been explored during recent years. Some ontology-based competency based tools or prototypes have been introduced, such as “CommOn” (Trichet and Leclere, 2003), a framework for building competency based systems. CommOn is based on two models (implemented with specific tools) which guide firstly the building of competency reference systems related to particular domains such as healthcare or information and telecommunication, secondly the identification and the formal representation of competency profiles, and thirdly the matching of competency profiles. The CommOn framework allows one to build shareable ontologies and knowledge bases represented with semantic web languages and to develop competency-based web services dedicated to human resource management. Also, other systems include “SMS – Skills Matching System” (Colucci et al., 2003) which is a prototype that is not linked with e-learning systems, “GMS” (Vasconcelos de Braga et al., 2003), which is an ontology-based competency management system for managing group competencies and which is not integrated with e-learning functionality or system. Moreover, an architectural proposal of a prototype system for ontology based competency management is presented by Reich et al., “Ontology based competency management in Swiss Life” (Reich et al., 2002), as a further step to XML-based competency management systems, such as “MaSel” (Garro and Palopoli, 2003). However, these prototypes do not integrate e-learning functions or links with learning objects and resources.

Research work is also conducted in the usage of ontologies in learning objects, in order to facilitate the discovery and reuse of learning objects stored in local and global repositories (e.g. Sicilia et al., 2002; Urban and Barriocanal, 2003). A learning object is defined as “any entity, digital or non-digital, that may be used for learning, education or training” (IEEE
Learning Technology Standards Committee, 2001). Initiatives such as the IEEE Learning Object Metadata (LOM), Dublin Core and IMS Global Consortium (see www.imsglobal.org) are developing standards, specifications and reference models for learning objects in order to facilitate the online retrieval and reusability of the latter. In the future developments of the system discussed lies the automatic discovery of learning objects from global repositories and their mapping with competency gap reports, in order to facilitate the user to access both local and global repositories of learning objects.

In Europe, there are three main projects that focus on ontology based competency development:

1. “TenCompetence” (see www.tencompetence.org), which integrates models and tools in the creation, storage and exchange of knowledge resources, learning activities, competence development programmes and network data for lifelong competence development;

2. “Knowledge on Demand” (Sampson et al., 2002a, b; see http://kod.iti.gr/), which aimed to design, develop and test a learning environment as a dynamic and adaptable online environment which allows the individual learner to acquire knowledge according to his/her personal learning needs, without however taking into account the organisational aspect of competency development; and

3. “Learning in Process” (see www.fzi.de/ipe/eng/projekte.php?id = 226), which addresses both the organisational and personal aspects and enables user-context aware delivery of e-learning material (Schmidt and Winterhalter).

Moreover, the use of ontologies in e-learning applications has been theoretically researched, for example with the architectural proposal of a prototype system for e-learning using ontologies (Stojanovic et al., 2003; Schmidt and Winterhalter, 2004). Some research efforts in integrating competency ontologies with e-learning have taken place in theoretical background (Woelk, 2002) and have been partially implemented (Hirate et al., 2001).

Additionally, many learning management systems integrate competency management features, without ontological support. For a detailed analysis of the main competency management features included in some popular learning management systems (LMS), one can refer to the paper by Draganidis and Mentzas (2006).

Evidently, there is a gap in the integration of ontology based competency-driven e-learning systems, apart from the “Learning in Process” project, with which we have a parallel architecture with the inclusion of the human resources aspect (e.g. succession planning, training needs analysis, expert finding, etc.). Our system's target is to develop and deploy an ontology-based competency management system that will provide the possibility for further enhancements, such as succession planning and training analysis, always using the developed ontology as a reference. Moreover, through the use of web services the system can be integrated with other human resource management or e-learning systems.

4. Competency ontology

The correct design of the competency ontology is a crucial step in the development of an effective competency management system, which possibly has to collaborate with other similar systems or e-learning and human resources applications. There has been significant effort in analyzing the competencies, their definitions and the corresponding proficiency levels through available dictionaries or after careful consideration. Moreover, ontology design included the correlations between all the entities of the ontology, namely competencies, employees, jobs, learning objects, etc.

Figure 1 presents the developed competency ontology, which consists of 17 classes in total, six of which are main classes and are used to describe employees, projects, jobs, competencies and organizational departments, while the other 11 classes are auxiliary classes used to describe the main classes and the relationships among them. A more detailed analysis of the main classes of this ontology is as follows.
4.1 Skill class

The “Skill” class is used for defining the organization’s set of skills as these stem out from the competency model that the organization has developed. This class has four proficiency levels, 1-4, which are described by “SkillDescr_Lvl”. Each proficiency level corresponds to a different expertise of the competency it refers to, with proficiency level 1 corresponding to novice possessors and proficiency level 4 meaning that the employee is fully equipped for this competency. This data is also used from the system to identify experts in each competency who could also act as mentors to employees who possess the specified competency at a novice level. Another interesting element is the definition of the proximity among different skills, by using the “SkillRelated” class. This class rates the relevance of two skills on a scale from 1 to 10, with 1 meaning that the two skills are completely unrelated. This is exploited by the system in many ways, such as suggesting learning objects in similar competencies or proposing employees, for example for a project, who possess a relevant skill to the one required by the project in the first place, in case this is not possessed by anyone available. Furthermore, by using the class SkillLearnObj_Lvl an association between a learning object and a skill can be defined, with this association referring to a specific proficiency level.

4.2 The LearningObject class

The “LearningObject” class defines learning objects, which as presented in Figure 1 belong to a category. A learning object is basically an online, offline, tacit or intacit source which
facilitates the process of learning, such as a book, a seminar, an e-learning course, a mentorship, etc.

4.3 The Department and Jobs classes

The “Department” class is used for defining the organization’s departments, while the “Jobs” class is used to assign job roles to each department. Each job requires a set of skills, which are defined by the instances of class SkillAtLevel and a job is fulfilled only by one employee.

4.4 The Employee class

This class defines the employees of the organization, basically with their personal data and their skills. Personal data are instances of classes such as “Degree”, “Studies”, “Dates”, “Experience” and employees’ skills are instances of the SkillAtLevelYear class. This class defines the skill, the corresponding proficiency level and the corresponding time data validity. This means that the skills of an employee are stored chronologically from their first day in the organization, which provides the opportunity for a chronological view of an employee’s skills development and useful reflections on the results of an organization’s talent management, training or other human resource development programs’ effectiveness.

4.5 The Project class

The “Project” class is used for creating the projects that are running in an organization. Apart from the name and an optional description of the project, each instance of this class includes a number of required skills and – as in the case of “Jobs” – these are instances of the SkillAtLevel class.

5. Functional architecture

Our research efforts focus on developing a prototype ontology-based system which will integrate competency management with e-learning and other human resource functions, such as succession and career planning, training needs analysis and organizational planning; The functions that the systems offers can be divided into the main categories of Core System Functions and Reporting Functions, as depicted in Figure 2. Human-computer interaction analysis was also conducted to design a user interface that can effectively meet the need of the two user categories – namely human resource staff and employees – for simple and quick access to all of the system’s functions.

Core system functions include functions that are responsible for inserting, updating and deleting ontology data. Apart from functions such as Insert Learning Object, Create Job Profile, Update Project, Delete Learning Object’s Category, etc., it also utilizes functions for creating, updating and deleting a relationship between two competencies, a job assignment and an association between a learning object and a competency.

Reporting functions includes provide the system user with a number of view functions, such as “View Competency Model” and also some more complex views, such as “View Employee’s Competencies” and “View Jobs’ Infos”, which produces a table with all the organization’s jobs, the corresponding department and the corresponding employee. Additionally, various reports on skill gap analysis, succession planning, experts and projects are provided. Below, more details about two reporting functions for projects –

“‘The correct design of the competency ontology is a crucial step in the development of an effective competency management system.’”
“Find Best Fit Employees for Project” and “Find Learning Objects for Project” – are provided.

The function “Find Best Fit Employees for Project” has as input a specific project and – provided that the required competencies for this project have been defined – returns the most suitable employees to participate in this project, according to the competencies they possess. The first criterion that an employee has to fulfill, is that he/she possesses at least half of the required competencies at a proficiency level at least equal to the required level. This function also ranks the employees who are qualified for the project. To define the sorting algorithm, let us consider two employees E₁ and E₂. For each of them we define three values:
1. \( N \), the number of the required competencies that the employee possesses in a proficiency level at least equal to the required level;

2. \( T \), the number of required competencies that the employee possesses regardless of the possessed level of proficiency (\( T \) has to be at least 1); and

3. \( S \), the sum of the differences between the level at which the employee possesses a competency and the required level, namely \( S = \sum P_i - R_i \), where \( P_i \) refers to the possessed level of proficiency and \( R_i \) to the required level.

We accept that \( E_1 \) is more suitable for a specific project than \( E_2 \) if:

- \( (N_1 > N_2) \); or
- \( (N_1 = N_2) \) and \( (T_1 > T_2) \); or
- \( (N_1 = N_2) \) and \( (T_1 = T_2) \) and \( (S_1 > S_2) \).

If there is still a draw result amongst some employees, then these are randomly ordered.

The function “Find Learning Objects for Project” has as its input a specific project and returns relevant learning objects, also indicating the corresponding proficiency level. In case there is no learning object related to a competency \( A \), the system searches the ontology in order to find learning objects related to other competencies \( B_i \), supposing that a relation with rate greater or equal to 8 has been defined between the competencies \( A \) and \( B_i \).

Below we present an analysis of the reporting functions:

- “Skill Gap Analysis” between the existing and the desirable employee’s competencies according to the position that he/she occupies. This function also binds the results to the corresponding learning objects and experts for every competency that the employee possesses at a level lower than the required, in order to help him/her to decrease or completely eliminate the existing gap.

- “Skill Gap Analysis for a Future Position (Succession Planning)”, which is similar to the previous function, with the difference that the employee’s competencies are compared to the required competencies of a job other than the one that he/she currently executes.

- “Average Department’s Proficiency Level”: this function calculates the average level of proficiency possessed by the employees for all competencies in the competency model for a specific department.

- “Average Organization’s Proficiency Level”: this is similar to the previous function but refers to the whole organization and not to a specific department.

- “Average Department’s Proficiency Level based on Jobs”: this function is similar to “Average Department’s Proficiency Level”, but in this case we take under consideration only the competencies of each employee that are required for his/her job. This means that if an employee possesses a competency \( A \) at a proficiency level \( L \), but competency \( A \) does not belong to the required competencies of the job that he/she executes, then when the sum of proficiency level for this specific competency is calculated, \( L \) will not be added and when the average for the specific competency is calculated (i.e. \( \text{average} = \frac{\text{sum of proficiency level}}{\text{number of employees}} \)) the employee will not be added to the number of employees. Respectively, if an employee ought to possess a competency \( B \) but he/she does not, then when calculating the sum of a proficiency level for competency \( B \), zero will be added and when calculating the average for competency \( B \) the employee will be...
added to the number of employees. The gap between the required and the possessed
level of proficiency for every competency is also calculated.

- “Average Organization’s Proficiency Level based on Jobs” is a function similar to
  “Average Department’s Proficiency Level based on Jobs”, but refers to the whole
  organization and not to a specific department.

- “Find Expert”, is a function that finds an employee who is an expert for a given
  competency A. This means that he/she possesses the competency A at a proficiency
  level 4 or – if there is no such employee – at proficiency level 3. If there is still no
  employee, then the system searches for an expert for other competencies B, supposing
  that a relation with a rate greater or equal to 8 has been defined between the
  competencies A and B.

- “Competencies Development”: this function shows to what extent an employee has
  developed his/her competencies; namely, it gives his/her possessed proficiency level
  and the corresponding time duration for every competency since he/she was hired.

6. Technical architecture

In this section we provide an overview of the technical architecture of the system, as
depicted in Figure 3. The front-end has been designed as jsp pages and through them the
users can access the various functions of the system, while some of the jsp pages include
JavaScript functions. Apache Tomcat has been used as the servlet container.

Figure 5  Functional performance of “Find Best Fit Employees for A Project” user scenario
The back-end is implemented in Java and access to the ontology is provided through the Jena API and RDQL. Jena is an open source Java API for RDF that is available on the internet (see http://jena.sourceforge.net/) and RDQL is a query language for RDF in Jena models. The idea is to provide a data-oriented query model so that there is a more declarative approach to complement the fine-grained, procedural Jena API. It is “data-oriented” in the sense that it only queries the information held in the models; there is no inference being done at this stage but the Jena model can provide some smart functionality such as creating on-demand certain triples. The RDQL system accepts as input a description of the application’s requirements, formed as a query, and returns that information in the form of a set of bindings. The back-end consists of a number of java classes and there are six main classes:

1. SkillManager;
2. DepartmentManager;
3. EmployeeManager;
4. JobManager;
5. LearnObjManager; and
6. ProjectManager.

Each of them is responsible for the management of the corresponding ontology’s class. Finally, all the data are stored in an RDF ontology.

6.1 RDF Cascade

During the implementation phase we encountered the following hindrance: RDF does not support referentially triggered action clauses (i.e. Cascade, etc.), in contrast to database management systems (DBMS). This made the implementation of update and mainly delete functions more complex than what had been initially estimated, since when deleting a certain object from the ontology, the system should take into consideration its relations to other objects in the ontology. Let us consider a specific example: the function of deleting a competency. Deleting just its name, description and proficiency level description is not...
enough, as it is highly possible that a learning object has been associated with this competency. Apart from that the competency could belong to the required competencies for a specific job or project and also to the competencies that one or more employees possess. Finally the definition of a relationship between this and another competency cannot be excluded. The correct path of deleting a competency requires the deletion of all its possible relations to other objects as described above. Our system has been designed with self-developed RDF cascading functionality but the complexity of such a task is not always predictable. In case of large ontologies it can prove to be very demanding, time-consuming or even impossible. Consequently, this issue calls for further research.

7. User scenarios

We will now consider three user scenarios:

1. Find Best Fit Employees for a Project;
2. Skill Gap Report; and

In each we will provide the reader with relevant screenshots and a description of the functional performance of the system.

Figure 7 Functional performance for “Employee Skill Gap Report”
7.1 User scenario: Find Best Fit Employees for a Project
The user wants to find the most suitable employees to execute a specific project (e.g. the development of the company website). The user selects a project, and supposing that the required competencies for it have been determined, the system generates the report shown in Figure 4, where the employees have been sorted according to the algorithm discussed above.

In Figure 5 we show in detail the way that the system performs this function. After the user has selected a project, the required competencies are identified together with the proficiency level that each employee possesses for every one of the required competencies. If an employee has at least half of the required competencies at a level equal to or greater than the requested, the employee is considered suitable. Finally a ranking of all the suitable employees takes place.

7.2 User scenario: Skill Gap Report
In this user scenario the user wants to produce the Skill Gap Report for an employee. After selecting the employee, a report is produced as shown in Figure 6. The system compares the possessed versus the required competencies and in case of a negative skill gap result, it provides the employee with a personalized learning path, with all the learning objects and experts needed, in order to help the employee improve his/her proficiency level. In this specific example the system does not contain a learning object for the competency “Java”.

Figure 8 “Find Learning Objects” and “Find Experts” functions
thus it informs the user that there is a learning object for the competency C#, which is related to the competency “Java”, and their relationship rate is 9.

Figures 7 and 8 provide a description of the way that the system performs this function. The actions “Find Learning Objects” and “Find Experts” (Figure 7) are examined in detail in Figure 8. As shown in Figure 7, once the user has selected the employee, the system finds his/her competencies and compares them to the required competencies. If there is a gap (i.e. employee possesses a proficiency level lower than the required) the systems searches for corresponding learning objects and experts. As shown in Figure 8, in case that there is no learning object or expert the search does not end. The system searches for related competencies and then for the corresponding experts and learning objects.

7.3 User scenario: Future Position Skill Gap Report for An Employee

Finally we discuss the scenario according to which the user wants to identify the Future Position Skill Gap Report for an Employee (Figure 9). This scenario is similar to the previous one, with the only difference that employee’s competencies are compared to the required competencies of a job different to the one he/she is currently executing. We only provide the equivalent analysis for Figure 7 (in Figure 10), since the way that the systems performs the Find Learning Objects and Find Experts functions is identical to that shown in Figure 8.

8. Conclusions and further research

In this paper we have provided an introduction to the competency management area and a practical approach to the integration of competency management, e-learning and ontologies, presenting a prototype ontology-based system. The competency ontology and the functional and technical architectures of the system have been analyzed, together with some user scenarios and issues that need further research in ontological systems, such as RDF cascading.

Our next steps include the deployment of a competency management system in Microsoft Hellas and the extension of the system with semantic search capability and inference engine, as well as the evaluation of the effectiveness of ontological systems in real-life environments. Moreover, further research lies in the area of expanding the system with
semantic attributes, such as adding semantic annotation to the web services exported, in order to enable it with broader integration capabilities with other ontology-based human resources systems. Moreover, by adding semantic web services capabilities to the system, we will enable semantic web matchmakers to accept the descriptions of our available services and match them against requirements from different requestors, in an automatic way. This process can currently be deployed manually, through registering the web services produced to the Universal Description, Discovery and Integration directory (UDDI).

References


