

Web-Based Competency Management System for Technopark of ITMO University

Boris Gordeev, Olesya Baraniuc
ITMO University
St.Petersburg, Russia
gordeevbr@gmail.com, ob@itc.vuztc.ru

Alexey Kashevnik
SPIIRAS
St.Petersburg, Russia
alexey@iias.spb.su

Abstract—The paper presents a competency management system development and implementation for residents of ITMO University Technopark. Every resident is described by the profile. Profile consists of several competencies and evidences with skills levels characterized the competence and evidence degree of possession.

I. INTRODUCTION

Last years, competency management of employers and companies at all is a popular research and development topic (see [1-2]). At the moment, advantages in a global competitions are defined by the quickness of new knowledge acquisition and utilization.

In according with [3] the main aspects involved in competence management are related to: development of concepts, skills and attitudes (formation); work practices, ability to mobilize resources, which distinguishes it from others; combination of resources; search for better performances; permanent questioning; individual learning process in which the higher responsibility should be attributed to the individual him/herself; relationship to other people.

In the presented paper, the competency management is understood as a technology that aims at acquiring and describing people knowledge in a company or companies knowledge for collaboration.

Described in the paper competency management system has been implemented using Java programming language and Spring Framework technology stack. It uses MySQL Database Server to store the information. Application layer consisted of Spring Data JPA to extract data from database using Hibernate, web controllers and views were implemented using Spring MVC.

II. SYSTEM ARCHITECTURE

System was designed to be a three layer client-server web application. Its architecture has been split into three layers: data server layer, application server layer, thin web client. Data server layer consists of data model that determines how the information in competency management system is kept. Application server layer consists of logic that allows the user to implement actions. Thin web client provides the web interface for the user.

Data model has been designed to consist of following data objects:

- Profile. Object that represents information about its owner and references its competencies. This object is abstract and is used for inheritance. 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 [4].
- Competency. References a skill and gives information about level of its possession by competencies' owner. 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 [4].
- Evidence. Object used to confirm level of skill possession. It can be represented by experience or certificate. A competency's evidence is used to capture information to prove the existence, sufficiency, or level of a competency. An evidence might include test results, reports, performance appraisals, evaluations, certificates, licenses, or a record of direct observation, such as a report given by a former supervisor or other employment reference. An evidence shall improve the trust to be given into the description of a competency [5].
- Skill. An unique and measurable skill or knowledge. All skills are a part of a single taxonomy and can be represented as a tree of skills. The demonstration of expertise (e.g., programming, the ability to make effective presentations, or to negotiate successfully) [5].
- Skill (Competence) Level. Object that stores information about what requirements subject must suffice to in order to possess specified skill level. A competence level reflects the degree to which a person can possess a certain ability [4].
- Task. This object represents some real-world task that requires some qualifications to be performed. This object gives information about it and references this task's requirements. A competency of an individual human is usually defined as the required assets of an individual person to perform a certain task. This may be a programming task, a project management, a talk or simply a decision in a certain context. Thus, a competence is goal-oriented [5].

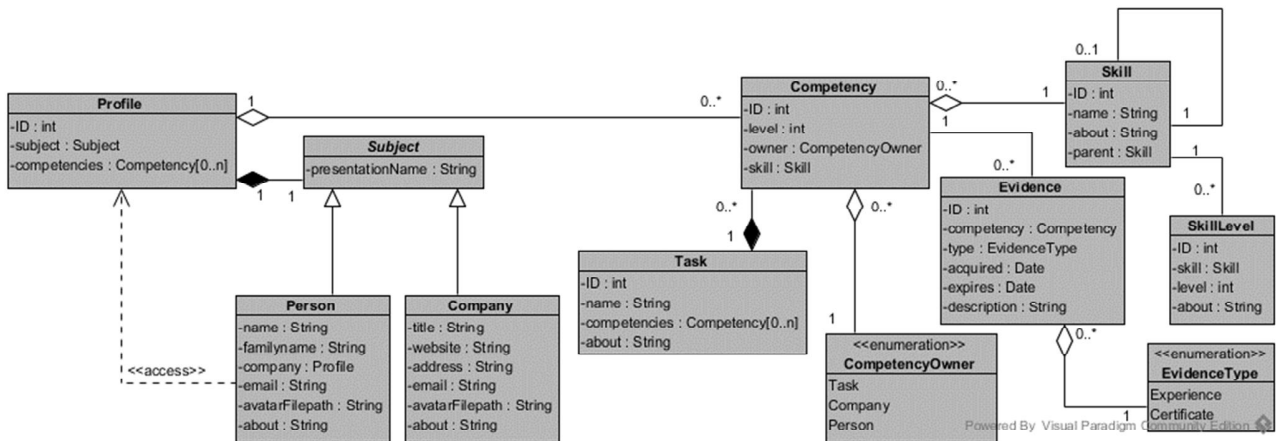


Fig. 1. Data model represented in UML notation

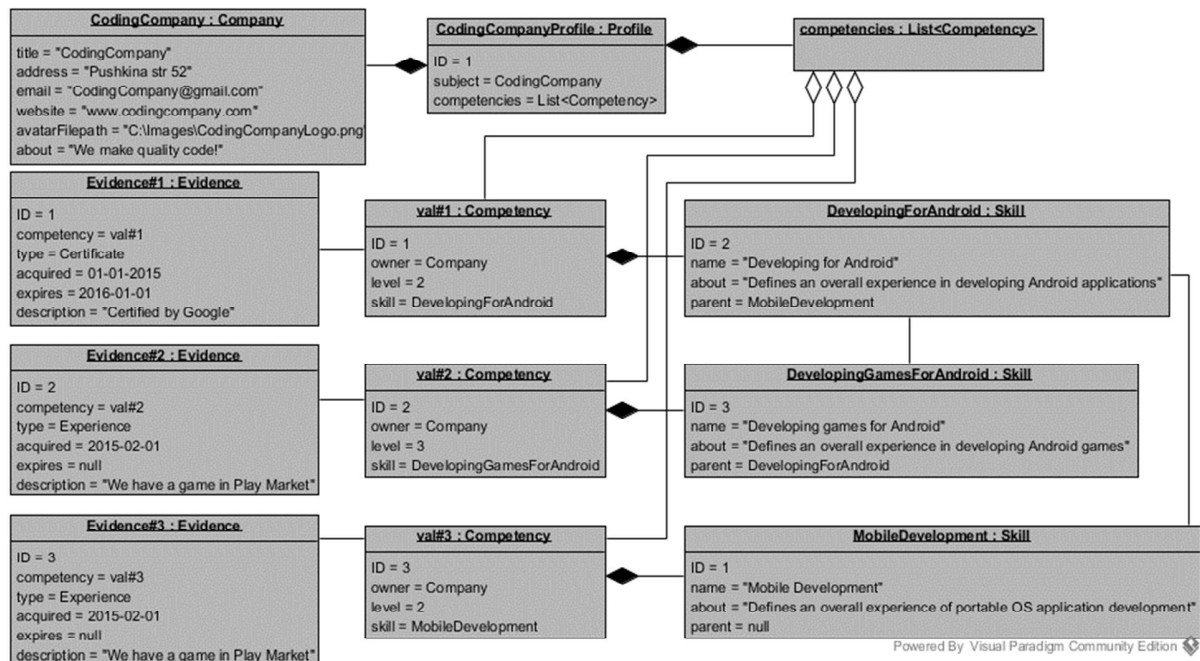


Fig. 1. Sample Profile

- Task Requirement. References a single skill and gives information about at which it level it should be possessed by subject in order to fulfill that requirement.

Data model represented in UML notation (class diagram) is presented in Fig. 1. Sample profile scenario represented in UML notation (object diagram) can be seen in Fig. 2.

III. IMPLEMENTATION

Application was implemented using Java programming language and Spring Framework technology stack. It used MySQL Database Server to store data. Application layer consisted of Spring Data JPA to extract data from database using Hibernate, web controllers and views were implemented using Spring MVC.

The following operations set is provided by the competency system.

- Compare profile with a task.
- Subprofile by skill set.
- Compare profile with a profile.
- Range profiles.
- Aggregate profile's possibilities.
- Competency search profiles.

Compare profile with a task operation can be used to determine if the selected profile can perform the selected task. Implementation has been done in the following way. For each task requirement within task, a competency implementing the same skill as the task requirement is extracted from the profile. If there is a skill, whose level for competency is less than the according skill level for task requirement, or if profile does not have competency with required skill, then selected profile cannot perform selected task. Otherwise it can.

Let's suggest that we have task "Frontend Development" ("t") which requires skill "Development" ("a") at level 3 and skill "Design" ("b" at level 4, Company Profile "x" with skill "a" at level 5 and "b" at level 4 and Company Profile "y" with skill "a" at level 3, "b" at level 3 and some other skill "c" at level 10. Then comparing "t" against "x" will return "true" while comparing "t" against "y" will return false ("y" has skill "b" at level 3 while it must be at least at level 4 to suffice).

Operation Subprofile a profile by skill set returns a copy of a profile being requested with only those competencies that implement skills from skill set requested. If profile does not have a competency for a skill in the set, then this skill's level for this profile is assumed to be 0. Implementation has been done in the following way. Implementation: for each skill within skill set, a competency with according skill is requested from profile and added to a new set of competencies. If such competency does not exist, a new competency with requested skill and skill level 0 is added. After all competencies are added, a new profile is created and information from source profile is copied into it. After this, resulting competency set is assigned to it.

Let's suggest that we have a Company Profile "p" with competencies: skill "a" at level 3, skill "b" at level 4 and "c" at level 2. Now, let's suggest that we have a set of skills such as: "b", "c" and "d". Then subprofiling "p" against such set will result in profile "p*" with competencies such as: skill "b" at level 4, skill "c" at level 2 and skill "d" at level 0.

Operation Compare profile with a profile that uses a metric to compare two profiles and returns a number. For the implementation the following formula is used:

$$R = 1 - \frac{\sum_{S \in (P1.C, P2.C)} \frac{|Level_{P1.C(S)} - Level_{P2.C(S)}|}{\max(|Level_{P1.C(S)} - Level_{P2.C(S)}|)}}{\text{Sum}(S)}$$

where: "R" is result – a number from 0 to 1, where 1 is perfectly equal and 0 is absolutely different; S is a skill; (P1.C, P2.C) are all skills implemented by all competencies of profile 1 and 2 (concatenation of set consisting of all skills that competencies of profile P1 implement with set consisting of all Skills that competencies of profile p2 implement); Level(Competency) is competencies' level; P.C(S) is a competency C of profile P implementing skill S; Summ(S) is sum of skills in (P1.C, P2.C).

Let's assume that we have two company profiles "x" and "y". The result of comparing them is result of comparing sets of competencies "x.C" and "y.C" that belong to these profiles. In case those competency sets are equal, the aforementioned equation will result in 1. Competency sets are equal when and only when for each competency "c" in "x.C" there is a competency in "y.C" with both the same skill and the same skill level. When both sets are empty, this is a special case when they are also equal, and the equation specified will return "1" in this case. Let's give a working example. If "x" has competency sets with skill "a" at level 2, skill "b" at level 3 and skill "c" at level 4, "y" has exactly the same set with "a" at level 2, "b" at level 3 and "c" at level. Then equation would look like:

$$1 - \frac{\frac{|2-2|}{2} + \frac{|3-3|}{3} + \frac{|4-4|}{4}}{3} = 1 - \frac{0+0+0}{3} = 1$$

In case competency sets "x.C" and "y.C" have no common skills, result will be "0". If there are skills "a", "b" and "c" from case "1", then having "x.C" with only one competency with skill "a" at level 2 and "y.C" with only one competency with skill "b" at level 3, then equation would look like:

$$1 - \frac{\frac{|2-0|}{2} + \frac{|0-3|}{3}}{2} = 1 - \frac{1+1}{2} = 0$$

In any other case, results will be between "0" and "1". If we add skill "c" with level "2" to "x.C" and with level "3" to "y.C" from case "2", then:

$$1 - \frac{\frac{|2-0|}{2} + \frac{|0-3|}{3} + \frac{|2-3|}{3}}{3} = 1 - \frac{1+1+\frac{1}{3}}{3} = \frac{2}{9}$$

So these sets are somewhere around 0.(2) equal.

Operation range profiles sorts a set of profiles based on comparison with a set of competencies. Implementation has been done in the following way. An empty profile is created and sets selected competencies as this profile's competencies. Then compares resulting profile with every profile selected and sorts them based on results.

Operation aggregate profile's possibilities returns all tasks that selected profile can perform. For the implementation the selected profile are compared with all tasks supplied in function call and returns set of tasks that this profile can perform.

Operation competency search profiles returns all profiles that match specified competency set. Implementation has been done in the following way. Compare all specified profiles one by one with specified competency set and returns only those that have result number different from perfect value (1) not more than acceptable (0.2 by default).

VII. CONCLUSION

Presented in the paper competency management system has been successfully implemented using Java programming language and MySQL Database engine. For the users a web interface has been developed. We are planning to accumulate information about ITMO University Technopark residents and use their competencies for partners finding.

REFERENCES

- [1] N. Lukyanova, Y. Daneykin, N. Daneikin, Communicative Competence Management Approaches in Higher Education, Procedia - Social and Behavioral Sciences, vol. 214, 2015, pp. 565–570.
- [2] J. Barbosa, M. Kicha, D. Barbosa, A. Kleina, S. Rigoa, DeCom: A model for context-aware competence management, Computers in Industry, vol. 72, 2015, pp. 27–35.
- [3] C. Bitencourt, Managerial Competence Management – the Organizational Learning Contribution, URL:

http://www2.warwick.ac.uk/fac/soc/wbs/conf/olkc/archive/oklc5/papers/i-3_bitencourt.pdf.

- [4] Vladimir Tarasov, Ontology-based Approach to Competence Profile Management, Journal of Universal Computer Science, vol. 18, no. 20 (2012), 2893-2919
- [5] Jürgen Dorn and Markus Pichlmair, A Competence Management System for Universities, Vienna University of Technology, Institute of Software Technology and Interactive Systems, Favoritenstr. 9-11, A-1040 Wien.