Towards FCA-facilitated Ontology-supported Recruitment Systems

Hui Ma[*,a], Sven Hartmann[b], Panrawee Vechsamutvaree[a]

[a] Victoria University of Wellington, New Zealand
[b] Clausthal University of Technology, Germany

Abstract. *Human resources (HR) recruitment is still a major challenge for many organizations since HR recruitment officers need to spend lots of time and effort to find the best candidate from a large number of applicants for a job position. In this paper, we propose a new approach for ontology-supported web-based HR recruitment systems. Our approach is facilitated by Formal Concept Analysis (FCA) for constructing domain-specific ontologies to model position requirements and applicants' competences. To evaluate our approach, we implement a prototype and conduct a case study. The case study demonstrates that the proposed approach has the potential to improve the effectiveness and efficiency of the HR recruitment process.*

Keywords. Ontology • FCA • Human Resources Management

1 Introduction

The human resources (HR) recruitment process is often time consuming and involves high operational costs because domain experts are often required to be involved to make decisions (Carroll et al. 1999). Carroll et al. (1999) states that the number of applicants can be up to 1000 to 2000 for one or two vacancies. In this case, HR officers and domain experts have to spend tremendous amounts of time on finding the best candidate for a position from the long list of applicants. To facilitate the access of HR managers and job applicants web-based recruitment systems are used to advertise job positions and collect job applications. After job applications being collected, HR officers and domain experts have to manually go through all the application to select candidates with competencies that best match the job position requirements. One example is tutor recruiting at the School of Engineering and Computer Science (ECS) at Victoria University of Wellington (VUW). In each trimester, a senior tutor (serve as a HR recruitment officer) has to spend a couple of weeks time to manually check the candidate competencies and requirements of tutors and select tutors from a large number of applicants. This process is not only time consuming but also not effective, i. e. may not be able to look at all the possible candidates for each course. Also, domain knowledge about competence and courses are required from course lecturers. It is important to capture the domain requirements so that quality solutions can be produced (Kop and Mayr 2008).

To improve efficiency and performance of the recruitment process, the use of a domain-specific ontology has been proposed for recruitment systems and competency management (Draganidis et al. 2006; Fazel-Zarandi and Yu 2008). Ontologies have been used for skill management, expertise finding and competency management (Woelk 2002), The benefits of ontology-supported

* Corresponding author.
E-mail. hui.ma@ecs.vuw.ac.nz

recruitment systems are reported in (Baader et al. 2003; Bizer et al. 2005; Maedche and Staab 2001; Mochol et al. 2006; Trichet and Leclere 2003; Woelk 2002; Zhang 2007).

An ontology identifies the relevant concepts and relationships among the concepts that exist in a domain (Gruber 1995). A well designed ontology can provide good support for machine processing and be understandable for human users (Baader et al. 2003; Bachmann et al. 2007). Building an ontology is often time consuming and requires the involvement of many domain experts. Also, different organizations have different business contexts and need organization-specific ontologies for their system. Many organizations, however, face the challenge that there is neither an existing ontology to be used nor sufficient expertise available to build an ontology. Formal Concept Analysis (FCA) has been proposed for building ontologies without the involvement of domain experts. Hwang et al. (2005) uses FCA to express an ontology in a lattice. The lattice is easy to understand for human users and can serve as a basis for ontology building. However, there is a lack of research on the use of FCA for building ontologies for HR recruitment systems.

The aim of this paper is to propose an ontology-supported approach for web-based recruitment systems which exploits FCA for building ontologies so that the HR recruitment process can be improved. In particular, we will present an ontology design approach that uses FCA to construct a domain-specific ontology. The constructed ontology is then used to automatically match applicants to job vacancies to reduce the involvement of domain experts. To demonstrate the effectiveness and efficiency of our approach, we develop a prototype system and conduct a case study for tutor matching.

This paper is organized as follows. In Section 2 we give a motivating example. Section 3 provides preliminaries that we will need later on. In Section 4 we present our proposed approach for ontology-supported HR recruitment where FCA is used for building domain-specific ontology. Section 5 reports on a case study that we have conducted to gain experience with the proposed approach. Section 6 concludes the paper and makes suggestions for future work.

2 A Motivating Example

HR recruitment often involves the processing of a large number of applications for a job vacancy. The School of Engineering and Computer Science (ECS) at VUW recruits about 150 tutor positions each trimester for about 50 courses at different levels.

When students want to apply for a tutor position at the school, they have to use an online portal where they provide personal information, such as their student ID, name, email, phone, degree completed, degree enrolled, year of study, major, tutoring experience, and preferred courses. Further information about the courses that the applicants have completed at VUW can be retrieved from the course results database using the applicant's student ID.

Different courses require different competences for their tutors. Based on the requirements for a particular course, a senior tutor manually selects a list of candidates which is then provided to the respective coordinator of the course to make the final decision. The matching of tutors to courses is based on a combination of several competence factors. For example, an ideal tutor of a course is someone who has the experience of tutoring the same course before. If she/he has not tutored the course before, then she/he should have studied the course at VUW before with a good grade, say grade B or better. Moreover, the education level of the candidate should not be lower than the course level.

If no candidate satisfies all the requirements of a course, then the senior tutor and the course coordinator need to work together to find someone who matches most of the requirements. For example, a candidate can be considered for a course if she/he has not done the course at VUW before, but has done some post-course of the course. To measure the degree of matching of the requirements, ranking methods can be used. In the literature, the matching between job positions and job candidates could be done by measuring the distance between respective concepts in an ontology hierarchy. One can use a standard competence dictionary such as the IEEE Reusable Competency Definition (IEEE RCD 2008) or DISCO (Müller-Riedlhuber 2009). However, for this case there is no exiting domain-specific ontology that can be used.

3 Preliminaries

In this section we will recall relevant background information about ontologies and FCA.

3.1 Description Logics

Description logics (DL) are a family of logic languages for representing knowledge and reasoning about it (Baader et al. 2003). The knowledge of interest for some application is represented in terms of individuals, concepts and roles, and is stored in a knowledge base. Concepts stand for sets of individuals, while roles stand for relationships to other individuals. A *DL knowledge base* is a set of axioms, and usually consists of two parts: a terminological knowledge layer (called TBox) and an assertional knowledge layer (called ABox). The TBox describes the terminology in use for the application, that is, defines the concepts and captures additional constraints on their interpretation. The ABox describes the individuals, that is, contains assertions that relate individuals to concepts.

We briefly review syntax and semantics of description logics. Let N_C and N_R be fixed sets of concept names and of roles names, respectively. One can then build complex concept expressions out of them by using the concept constructors provided by the particular description logic being used (see Looser et al. 2013). Let C denote the set of complex concept expressions that can be obtained by finitely many applications of these constructors. The members of N_C are called atomic concepts. We further use the empty concept $\bot$ as a shortcut for $\neg\top$, and $(\leq m).R$ as a shortcut for $\neg(\geq m+1).R$.

Different description logics vary by the concept constructors that they permit. The choice of a particular description logic is usually done by balancing expressiveness against the complexity of the associated decision problems. In this paper, we use $\mathcal{ALN}$ which is among the most popular description logics (Liao et al. 1999). However, the ideas discussed in this paper are general in the sense that they can be easily tailored to other reasonably expressive description logics. Note that $\mathcal{ALN}$ is included in $\mathcal{SHOIN}^{(\mathcal{D})}$, the description logic underlying OWL-DL.

A *subsumption axiom* is a statement of the form $C_1 \sqsubseteq C_2$ with concepts C_1, C_2 in C. A *terminology* (or TBox) $\mathcal{T}$ is a finite set of subsumption axioms. We use the shortcut $C_1 \equiv C_2$ to denote both $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$, and call C_1, C_2 *equivalent*.

Interpretations are used to assign meaning to syntactic constructs. An interpretation $\mathcal{I}$ consists of a non-empty interpretation domain O and an interpretation function $\mathcal{I}(.)$, which assigns to each atomic concept a subset of O, and to each role a binary relation on O. The interpretation $\mathcal{I}$ can be easily extended to concept expressions in N_C. An interpretation $\mathcal{I}$ is a *model* of $\mathcal{T}$ if $\mathcal{I}(C_1) \subseteq \mathcal{I}(C_2)$ holds for every subsumption axiom $C_1 \sqsubseteq C_2$ in $\mathcal{T}$. A model is finite if the interpretation domain O is finite. In this case, the model is also said to be an *instance* (or ABox) of $\mathcal{T}$.

A concept C_1 *subsumes* a concept C_2 if $\mathcal{I}(C_1) \subseteq \mathcal{I}(C_2)$ holds for every instance $\mathcal{I}$ of $\mathcal{T}$. We also write $\mathcal{T} \models C_1 \sqsubseteq C_2$. We use $\mathcal{T} \models C_1 \equiv C_2$ as a shortcut for $\mathcal{T} \models C_1 \sqsubseteq C_2$ and $\mathcal{T} \models C_2 \sqsubseteq C_1$.

Example 1 For our HR recruitment system we may use the subsumption axiom $\texttt{SWEN301} \sqsubseteq \exists\texttt{hasTask.EssayMarking}$, $\texttt{SWEN301} \sqsubseteq \exists\texttt{isPostcourse.SWEN222}$ to state that course SWEN301 has a task of essay marking and is a post-course of SWEN222.

The subsumption problem asks whether $\mathcal{T} \models C_1 \sqsubseteq C_2$ holds for concepts C_1, C_2 and a TBox $\mathcal{T}$. The satisfiability problem asks whether $\mathcal{T} \models C \equiv \bot$ holds for a concept C and a TBox $\mathcal{T}$. Both problems are decidable for description logics such as $\mathcal{ALN}$. For details, we refer to Baader et al. 2003.

3.2 Formal Concept Analysis

It has been shown that FCA can help to structure and build an ontology for particular application of interest (Cimiano et al. 2005; Hwang et al. 2005). FCA provides a formal framework for recognizing groups of elements that exhibit common properties. It is a theory of data analysis which identifies conceptual structures among data sets. These structures can be graphically represented as conceptual lattices which allow the analysis of complex structures and the discovery of dependencies within the data. FCA can be used for visualizing the ontology in the form of a lattice, in order to support navigation and analysis tasks. Representing an ontology as a lattice makes it easy to understand for human users.

Next, we review basic notions from FCA (Ganter and Wille 1999). A *formal context* is a triple $A \times B$ $K := (O, C, R)$, where O is a set of individuals, C is a set of properties (or attributes), and R is a relation $R \subseteq O \times C$ that links each individual $o \in O$ to the properties satisfied by o. That is, $(o, b) \in O$ states that O has property b. A *formal concept* (or FCA concept) is a pair (A, B) such that A and B are maximal with $A \times B \subseteq I$. The set $A \subseteq O$ is called the *extent*, while the set $B \subseteq C$ is called the *intent* of the concept. A formal concept (A_1, B_1) is a *subconcept* of a formal concept (A_2, B_2) if $A_1 \subseteq A_2$ (or equivalently $B_1 \supseteq B_2$). The concept lattice $\mathfrak{B}(\mathbb{K})$ of $\mathbb{K}$ is the set of all its FCA concepts together with the subconcept/superconcept relation.

As usual, we use lattice diagrams to illustrate the ordering relation among the formal concepts in a concept lattice. The nodes of the lattice are labelled by the respective FCA concepts. For an example, see Figure 4. For the sake of clarity, however, node labels do not show the full extent and intent of an FCA concept, but show an individual o only in the most specific FCA concept it belongs to.

Contexts can be represented as tables, with columns headed by the properties b and rows headed by individuals a. The cells of the table are marked if and only if the relations holds for the corresponding pair of individual and property. For an example see Figure 3.

4 Ontolgy-supported HR Recruitment using FCA

In this section, we present our approach for an ontology-supported web-based HR recruitment system where FCA is used to build a domain-specific ontology.

4.1 Framework of an Ontology-supported Web-based HR Recruitment System

Figure 1 shows our framework of an ontology-supported web-based HR recruitment system.

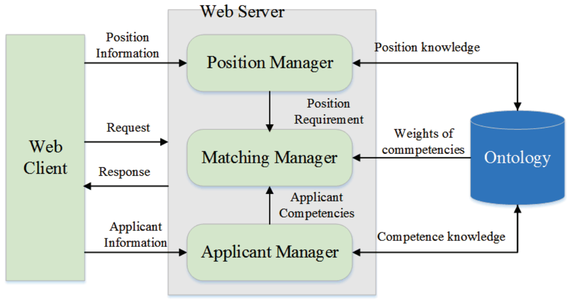

Figure 1: Framework of a web-based HR recruitment system

In the front-end, we have Web Client which connects to Web Server using HTTP request and HTTP response. In the Web Server, there are three modules which are Position Manager, Applicant Manager and Matching Manager. The three modules in Web Server connect to the Ontology for storing and retrieving the data. Position Manager is used for managing job positions, such as adding a new position or editing position requirements. Applicant Manager is used for managing applicants' profiles, such as adding a new applicant, editing the profile of an applicant, or searching applicants. Matching Manager is used for managing the position-candidate matching. For matching applicants to positions, the system compares the position requirements retrieved by Position Manager and the applicant's competencies retrieved by Applicant Manager. Matching Manager is also used for managing the weights of each competency.

Often, applicants do not match all the requirements. To rank candidates for a position, we need a ranking method. We will assign weights to relevant competence factors. The matching of different competencies have different weights since the importance of each competency to a position is different. For example, for tutor-course matching, tutoring experience is weighted higher than the education level of a candidate.

4.2 FCA-based Ontology Building

To collect the domain knowledge that can be used for selecting candidates we propose to use FCA. Figure 2 shows the steps of our FCA-based method of building domain-specific ontologies from candidate application information. Note that one can take advantage from candidate application instances collected from previous years to build the ontologies.

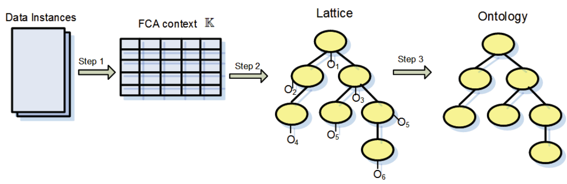

Figure 2: FCA-based ontology building process

We first create a formal context matrix. Then, we use this formal context matrix to build the concept lattice for an ontology. In one organization, we many need to build several contexts. For example, for the tutor recruitment system, we use FCA to build a prerequisite course context and an education level context.

Figure 3 shows an example of a context $K(O, C, R)$ of course prerequisites. The set of objects O is a collection of candidates, denoted by Alex, Bob, etc. The set of attributes C is the set of courses. If a candidate has already passed a course, then the relationship R holds and we mark it by "X" in the table. The concept lattice in Figure 4 is derived from the context in Figure 3.

Each node in this lattice, illustrated by a circle, is a formal concept. For example, one of the formal concepts of the context described in Figure 3 is {Alex, Eric} × {COMP103, SWEN223}, where the set {Alex, Eric} is the extent of the concept, while the set {COMP103, SWEN223} is its intent. According to Figure 4, the formal concept {Alex, Bob} × {COMP103, SWEN221, SWEN222} is a sub-concept of the concept {Alex, Bob, Candy} × {COMP103, SWEN221}.

	SWEN222	SWEN223	SWEN301	SWEN221	COMP103	SWEN423
Alex	X	X	X	X	X	
Bob	X			X	X	
Candy				X	X	
Daniel					X	
Eric		X			X	
Fiona	X	X	X	X	X	X

Figure 3: Formal context of course pre-requisites

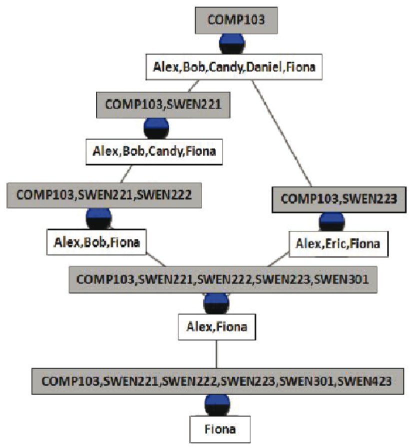

Figure 4: Concept lattice of course pre-requisites

To build an ontology, we can use the reduction procedure proposed in (Haav 2004). The reduction procedure has two steps. The first step is to eliminate the redundant elements. For example, a pair (A, B) and B (intent) appear in every descendant. We can eliminate the inherited elements. The second step is to eliminate the lattice of extents. Figure shows an example of a resulting ontology.

5 A Case Study

To demonstrate the effectiveness and efficiency of our proposed approach, we have implemented a

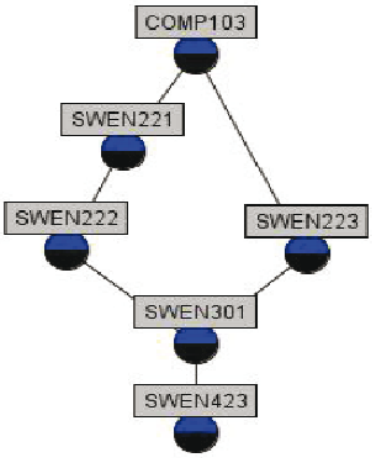

Figure 5: Ontology of course pre-requisites

prototype of a web-based tutor recruitment system using our proposed framework and used the proposed FCA-based ontology building method to build ontologies. We have then designed a test case, based on real data at ECS. We used the prototype to select a shortlist of candidates for each tutor position. In the mean time we invited the senior tutor to select a list of candidates for each course. For this case study the senior tutor was asked to list the top 3 suitable candidates. We have then compared the quality of the results and the time used by the two different approaches.

5.1 Prototype of an Ontology-supported Recruitment System

We have developed a prototype to demonstrate the performance of our proposed framework. We used Apache Jena^{TM} to store and manipulate our ontologies, so that SPARQL could be used for querying. We used our FCA-based approach to build ontologies needed by the tutor recruitment system. We collected information on tutor applicants and courses, and constructed ontologies to capture the relationships between courses, degrees, and course requirements.

For each course, the prototype system retrieves the course requirements and the applicants' competency information from the ontologies. Then the Matching manager compares the course requirements to the applicants' competencies and calculates the fitness value of each qualified applicant, and displays a shortlist with the required number of ranked candidates, starting with the highest fitness. Often, the candidates do not match all the requirements. The fitness value permitted the ranking of candidates.

In this case study, we considered the fitness of a candidate as a weighted sum of different competencies, e. g., education level, experience:

$$Fitness = w_{EC} + w_L + w_{EP} + w_C$$

Herein, w_{EC}, w_L, w_{EP}, w_C denote the weights assigned for having experience of tutoring the course, education level, having experience of tutoring a post-course, having done the course. We have normalized the weights.

For each course, the weights of competencies are different. For example, for matching tutors to courses, experiences of being a tutor of the course has higher weight than the education level of the candidate. In this way, the system ranks a candidate who has not completed a Bachelor degree but has tutored this particular course before higher than a candidate who has a Master degree but has not tutored the course before.

5.2 Results

In this section we compare the results obtained with our system to the results obtained by the senior tutor using the existing approach. Due to the page limit we only discuss some example courses in this article.

Firstly, we compare the time used by the two approaches. Recall that there are 150 applicants a tutor position per trimester. Using the current approach the senior tutor needs to retrieve the information of the candidates and courses from different resources and then manually selects suitable candidates by using her domain knowledge of courses or by consulting the respective course lecturers. Using the current manual system the senior tutor needs to spend 75 hours in total to check first tutors for each courses, assuming 30 minutes for each candidate. Using our prototype she only needs to spend a few minutes.

Secondly, we compare the lists selected by the senior tutor and the ones produced by our prototype, see Table 1 and Table 2, respectively. For most courses, the results obtained by the two different approaches are very similar, except for COMP307. One candidates has been selected by the senior tutor because this candidate got good grades in other courses. However he had neither studied nor tutored COMP307 and its post-courses before. Candidates who studied COMP307 before got grades lower than B and, hence, are not selected. On the other hand, our system listed 300198173 and 300198193 as the most suitable candidates since they both got A+ for COMP422 which is a post-course of COMP307. The results obtained by our system are better because the two candidates achieved good grades in COMP422, and thus should have knowledge of COMP307 which is a prerequisite of COMP307. From these results, we can see that without using our system, potential candidates can be missed out.

	1st	2nd	3rd
SWEN221	300198187	300198201	300198202
SWEN222	300198201	300198180	300198187
SWEN307	300198201	-	-
SWEN304	300198175	300198176	300198182

Table 1: Matching results from the senior tutor

	1st	2nd	3rd
SWEN221	300198201	300198187	300198202
SWEN222	300198201	300198180	300198187
SWEN307	300198173	300198176	300198178
SWEN304	300198176	300198182	300198175

Table 2: Matching results from our prototype

From the above results, we can see that our prototype has the potential to find suitable candidates due to the usage of ontologies that model the relationship between courses. Some of the candidates might be missed out using the existing approach. The propotyope system has huge potential to assist the senior tutor by reducing the time and effort needed for matching the applicants to courses.

6 Conclusion

In this paper, we have proposed an approach of ontology-supported web-based HR recruitment that exploits FCA for building domain-specific ontologies. We developed a prototype of our proposed approach and conducted a case study for tutor-course matching at ECS of VUW. The case study shows that our proposed approach has indeed the potential to improve the efficiency and effectiveness of the HR recruitment process.

References

Baader F., Calvanese D., McGuinness D. L., Nardi D., Patel-Schneider P. F. (eds.) The description logic handbook: theory, implementation, and applications. Cambridge University Press

Bachmann A., Hesse W., Ruß A., Kop C., Mayr H. C., Vöhringer J. (2007) A Practical Approach to Ontology-based Software Engineering. In: Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'07), pp. 129–142

Bizer C., Heese R., Mochol M., Oldakowski R., Tolksdorf R., Berlin F. U., Eckstein R. (2005) The Impact of Semantic Web Technologies on Job Recruitment Processes. In: In Proceedings of the 7th International Conference Wirtschaftsinformatik, pp. 1367–1383

Carroll M., Marchington M., Earnshaw J., Taylor S. (1999) Recruitment in small firms: Processes, methods and problems. In: Employee Relations 21(3), pp. 236–250

Cimiano P., Hotho A., Staab S. (2005) Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis. In: Journal of Artificial Intelligence research 24, pp. 305–339

Draganidis F., Chamopoulou P., Mentzas G. (2006) An ontology based tool for competency management and learning paths. In: 6th International Conference on Knowledge Management (I-KNOW 06), pp. 1–10

Fazel-Zarandi M., Yu E. (2008) Ontology-Based Expertise Finding In: Practical Aspects of Knowledge Management: 7th International Conference (PAKM 2008), Proceedings Yamaguchi T. (ed.) Springer, pp. 232–243

Ganter B., Wille R. (1999) Formal Concept Analysis. Springer

Gruber T. R. (1995) Toward principles for the design of ontologies used for knowledge sharing? In: International journal of human-computer studies 43(5-6), pp. 907–928

Haav H.-M. (2004) A Semi-automatic Method to Ontology Design by Using FCA. In: Proceedings of the CLA 2004 International Workshop on Concept Lattices and their Applications

Hwang S.-H., Kim H.-G., Yang H.-S. (2005) A FCA-based ontology construction for the design of class hierarchy. In: Computational Science and Its Applications–ICCSA 2005, pp. 307–320

IEEE RCD (2008) IEEE Standard for Learning Technology - Data Model for Reusable Competency Definitions

Kop C., Mayr H. C. (2008) Templates in Domain Modeling - A Survey. In: The Evolution of Conceptual Modeling - From a Historical Perspective towards the Future of Conceptual Modeling [outcome of a Dagstuhl seminar held 2008]., pp. 21–41

Liao M., Hinkelmann K., Abecker A., Sintek M. (1999) A competence knowledge base system as part of the organizational memory. In: German Conference on Knowledge-Based Systems. Springer, pp. 125–137

Looser D., Ma H., Schewe K. (2013) Using Formal Concept Analysis for Ontology Maintenance in Human Resource Recruitment. In: Ninth Asia-Pacific Conference on Conceptual Modelling (APCCM), pp. 61–68

Maedche A., Staab S. (2001) Ontology learning for the semantic web. In: IEEE Intelligent systems 16(2), pp. 72–79

Mochol M., Nixon L. J. B., Wache H. (2006) Improving the Recruitment Process through Ontology-based Querying.. In: SEBIZ. CEUR Workshop Proceedings Vol. 226

Müller-Riedlhuber H. (2009) The European Dictionary of Skills and Competences (DISCO): an example of usage scenarios for ontologies. In: Proc. 5th International Conference on Semantic Systems, ISEMANTICS. JUCS Conference Proceedings Series, pp. 467–479

Trichet F., Leclere M. (2003) A Framework for Building Competency-Based Systems Dedicated to Human Resource Management In: Foundations of Intelligent Systems: 14th International Symposium (ISMIS 2003), Proceedings Springer, pp. 633–639

Woelk D. (2002) E-Learning, Semantic Web Services and Competency Ontologies. In: Proceedings of EdMedia: World Conference on Educational Media and Technology 2002. Association for the Advancement of Computing in Education (AACE), pp. 2077–2078

Zhang J. (2007) Ontology and the semantic web. In: Ontology and the Semantic Web 2007. dLIST, pp. 9–20