

Using Ontology for Personalised Course Recommendation Applications

Mohammed Essmat Ibrahim¹, Yanyan Yang², David Ndzi³

^{1,2}School of Engineering
University of Portsmouth, Portsmouth, UK
{mohammd.ibrahim,Linda.Yang}@port.ac.uk

³School of Engineering and Computing
University of the West of Scotland, Scotland
david.ndzi@uws.ac.uk

Abstract. The primary data source for universities and courses for students is increasingly becoming the web, and with a vast amount of information about thousands of courses on different websites, it is quite a task to find one that matches a student's needs. That is why we are proposing the "Course Recommendation System", a system that suggests the course best suited for prospective students. As there has been a huge increase in course content on the Internet, finding the course you really need has become time-consuming, so we are proposing to use an ontology-based approach to semantic content recommendation. The aim is to enhance the efficiency and effectiveness of providing students with suitable recommendations. The recommender takes into consideration knowledge about the user (the student's profile) and course content, as well as knowledge about the domain that is being learned. Ontology is used to both models and represent such forms of knowledge. There are four steps to this: extracting information from multiple sources, applying ontologies by using Protégé tools, semantic relevance calculation and refining the recommendation. A personalised, complete and augmented course is then suggested for the student, based on these steps.

Keywords: Recommendation systems, Semantic Web, Ontology, Course Selection, Semantic Similarity.

1 Introduction

When choosing a suitable university course, students need information from many external sources in order to improve their decision-making processes, including the web. The process of choosing a course can be extremely tedious and very complex. As students are required to choose from a wide range of courses, based on a series of decisions and recommendations [1], they frequently find it difficult to find a course that is suitable for them. It is possible to find courses that cover almost every domain of knowledge [2] and each university publishes information about this on their websites.

Such abundant information means that students need to find, organise and use resources that can match their individual goals and interests, as well as their current knowledge. This can be a slow task as it involves accessing each and every platform, searching for available courses, reading each of the course syllabuses carefully, and then choosing the appropriate one.

There are many online systems that are currently available to find and search for courses [3]. These tools are based on either previous users' knowledge of courses or keyword-based queries. Just because more course information is now offered on university websites, it does not automatically mean that students possess the cognitive ability to evaluate them all. Instead, they are confronted with a problem that is generally termed as "information overloading"[4]. Studies also show that course choice decision is influenced by the student's background, as well as their personal or career interests [1].

By identifying the needs of the students and their areas of interest, it is possible to recommend an appropriate course. It is possible to help them to choose a course by developing methods that will both integrate data from multiple heterogeneous data sources and allow them to rapidly set valuable course-related information. This is based on their own preferences, such as electronic engineering [5].

In order to represent an area of knowledge, an ontology is used that formally describes a list of terms, each representing an important concept, such as classes of objects and the relationships that exist between them [6]. Ontologies provide formal semantics which can be used to process and integrate a range of information on the Internet. Ontology is described by Gruber [7] as an explicit specification of a conceptualisation.

Recommendation systems have recently offered personalised and more relevant recommendations. This is achieved by using information on the basis of situations, such as studying various objects, context and areas of interest, location and careers. For example, courses that are recommended to a student who wishes to work in IT, and is searching for "Computer Networking", will be different to those that are recommended to a student who aims to become an academic member of staff in the same area. This is because both their requirements and the level of education is different. These are treated as contextual data, which has been measured as a major source of the correctness of recommendations [8-9].

This paper's proposed approach overcomes the overloading problem by using personalised search results. It extracts and integrates information about courses from many different sources, builds ontology mapping of the information and sorts it in the database. As designing ontology is the creation process of a lot of classes and relationships, the user will be able to gain clear knowledge about the course [3]. In this paper, we build a relationship between relevant information on the Internet, including course modules, job opportunities and users' interests. Ontology provides a vocabulary of classes and properties to describe a domain and emphasises the sharing of knowledge [6]. The use of semantic descriptions of the course and learner profiles allows for both qualitative and quantitative reasoning about the matching that is available, as well as the required courses and student interests that are needed to refine the process of deciding which course to take.

This present paper is structured as follows. In Section 2, we discuss the previous work that is relevant to this study. Section 3 presents the ontology model in order to express knowledge about the student profile, course content, job content and the domain that is being learned. Section 4 describes the ontology-based semantic recommendation in detail, and Section 5 describes the prototype implementation and preliminary results. Finally, Section 6 concludes the paper by pointing out the direction of future work.

2 Related Work

Recommendation systems are a promising way to effectively filter out an information overload. These are “software tools and techniques that provide the suggestions for items to be of use to a user” [11]. A variety of techniques have been used to perform a recommendation, such as content-based, collaborative, hybrid and other techniques [10-12]. The attention needed to develop the various recommender systems is still high because there is an abundance of practical applications that can help users to deal with the overload of information and provide a personalised service [13]. The objects that are influenced by recommender systems include a wide spectrum of artefacts, such as books, documents CDs, television programs and movies. Compared with these fields, and the emergence of the education field, course content recommendation is a new topic, which has only been investigated by several systems over the past few years. Many kinds of research into course recommendation systems that aim to help students to find courses that are suitable for them have been carried out [14-16]. Current course recommendation systems collect information from a single data source, including students, university databases, users’ course ratings, course histories, past behaviour of students, historical enrolment data and previous students’ work histories. The students, however, need to gain a clear knowledge of the relevant course that will meet both their personal needs and career interests.

Recently, a recommendation system and expert system was established that was generally based on domain knowledge and problem-solving methods, such as shared and reused knowledge. The recommendation system and expert system utilised an ontology in order to solve classification, annotation, rendering and to arrange different interpretations that make knowledge representation work efficiently.

We, therefore, proposed an approach that uses the knowledge-based semantic approach to making recommendations to students. We also support recommendation refining. We mainly consider the user’s profile context for content recommendation, as we did in their learning goal and prior knowledge. This system extracts information from a number of sources about the content and then discovers semantic matching between the course information and user/student profile.

There are several techniques that have been employed to perform data matching in different applications. Two common measurements used to calculate the similarity of data records for matching are TF-IDF based methods [17] and String edit distance [18]. A support vector machine (SVM) [19] classifier, which has been trained with these similarity measures, is then used to identify instances that refer to the same real entities. This enables us to create semantic relations between different data sources.

3 Ontology Model

We use ontologies in the proposed approach to model knowledge about the course content (course profile), knowledge about the user (student profile) and the domain knowledge (the taxonomy of the domain being learned). Within the domain of knowledge representation, the term ontology refers to both the formal and explicit descriptions of domain concepts. These are frequently conceived as a set of entities, relations, functions, instances and axioms [7]. By enabling the users or contents to share a common understanding of the knowledge structure, ontologies give applications the ability to interpret the context of student profiles and course content features, based on their semantics. In addition, the hierarchical structure of the ontologies allows developers to reuse domain ontologies (for example, in computer science and programming language) to describe learning fields and build a practical model, without starting from scratch.

We constructed three ontologies in the proposed system. These are course ontology, student ontology and job ontology. To test our system, it was decided the domain ontology would be computer network courses. Knowledge, represented by ontologies, can be combined into one single ontology, as shown in this paper.

In addition, knowledge from different ontologies can be combined by merging ontologies. We have shown the merger of two ontologies in this paper: the first ontology is the course and the second is the student profile ontology. The latter contains details of the student developed in the educational domain. Protégé 5.1.0 tools were used to develop and merge by using the ontologies [20]. The course content ontology depicts various contexts about course information, including the course topic, type, duration, level and modules, as shown in Fig. 1.

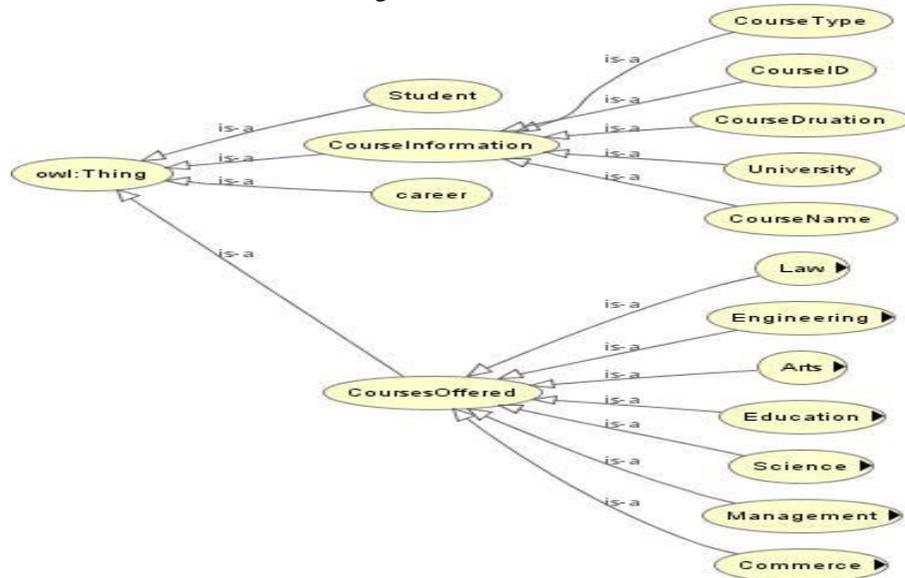


Fig.1. Course ontology

The user/student ontology includes information about the student, such as personal information, academic information and general information. Refer to more details in [22], as shown in Fig. 2.

The job ontology includes information, such as the job topic, job requirements and location.

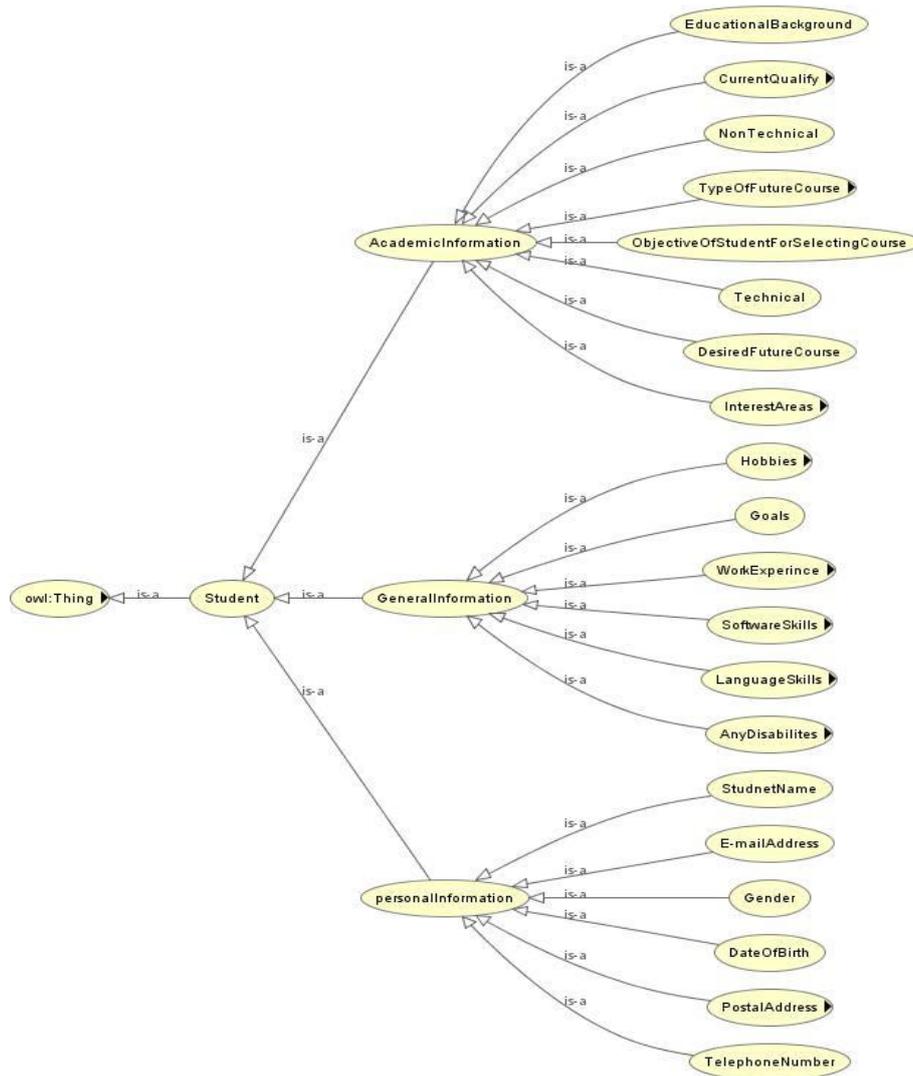


Fig. 2. Student profile ontology

After we have created ontologies for the university course and user/student profile, we will need to carry out the following steps, so as to provide uniform knowledge about the course information:

1. The extraction of similar concepts between two ontologies, such as “computer department” with “computer science department”, “Faculty” with “Academic staff”, and “Staff” with “Technical staff”, which are similar to each other.

2. The measurement and determination of the type of similarity relations between terms. Each approach and algorithm could consider different types of similarity relations between the terms (such as “Equivalent”, “Less general”, “More general” or “Overlapping”. For example, “computer department” has an equivalent relationship with “computer science department” and “Master Program” has a less-general relationship with “Graduate Program”). This study considered using the String edit distance and TF-IDF methods to measure the similarity relation between the terms.

3. Representation of similar relations between terms. Similarities between terms are formalised in this step. For example, we should represent the similarity relation between “course” and “program” by one formal language. These formal descriptions are from similarity relationships and are called semantic mapping information.

4. Execution of semantic mapping between similar concepts. The concepts, which are similar to each other at this stage, are mapped together. For example, “computer department” is mapped to “computer science department”.

Therefore, ontology semantic mapping is a difficult, complex process that requires the execution of an algorithm (for the detection and measurement of similarities), scripting language (for representing mapping information) and tools (for the execution of semantic mapping).

3.1 Ontology mapping

The proposed system database consists of tables of course information, student profiles and job information. Each table consists of a set of attribute values. The attributes of tables are an RDF node. We define a semantics mapping as a process from a database to an RDF graph, in a final ontology. For example, let C1 be a course in the T1 and C1 be entity courseID, which is a primary key. All of the other attributes, such as course_modules, will be related to courseID if a student S1 has an attribute, such as Main_area_interest. The ontology mapping will be linked to the concepts in the course_title and course_modules, with student_main_area. The relationship between the concepts are based on the subjects’ properties; for example, the domain of (has_select) property will be the person or student and the range will be the course. For the (leadTo) property, the domain will be the course and the range will be a job. The ontology representation of the database tables is shown in Fig. 3.

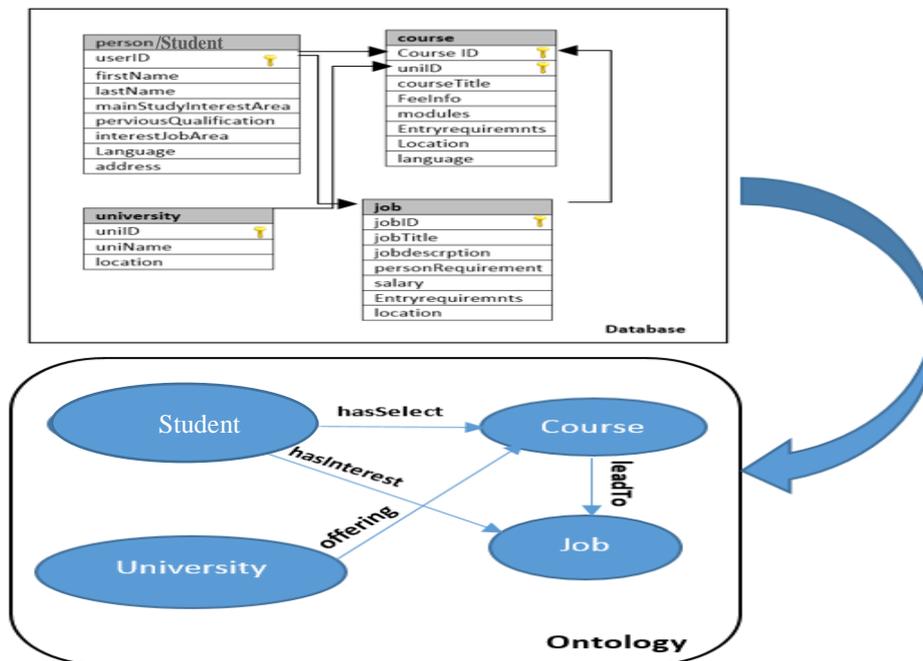


Fig. 3. Ontology represent of database tables

3.2 Ontology Features Matching

The use of ontology allows us to improve the methods that only compute string similarities between ontology instances.

Two ontology features are utilised in the proposed extended method. The first feature is the ontology hierarchy. With an ontology schema, we can compute the subsumption relations between concepts in the ontology schema by using a specific reasoner. A hierarchy of the concepts can then be constructed, which allows us to explore the “concept-level similarity” of instances. The quality and completeness of ontology data varies because different data sources contribute to it separately. There is no guarantee that the instances that refer to the same real-world entity are identified with exactly the same concept by different data sources.

For example, any search for the course base on a “computer network”, as a topic on the UCAS website [21], will give over 120 alternative courses that are similar in their topic concepts (including computer network security, Computer Network Technology, Computer Network Administration and Management, Network Computing, Network and Computer Systems Security, among offers). They provide similarity as an area of study, but when each topic was analysed, we found they had different modules or units that influenced the student’s job fields. These modules or units were present as instances in ontology, as shown in Fig. 4. We need to share these concepts and define the relations for the modules with similar topics by the ontology.

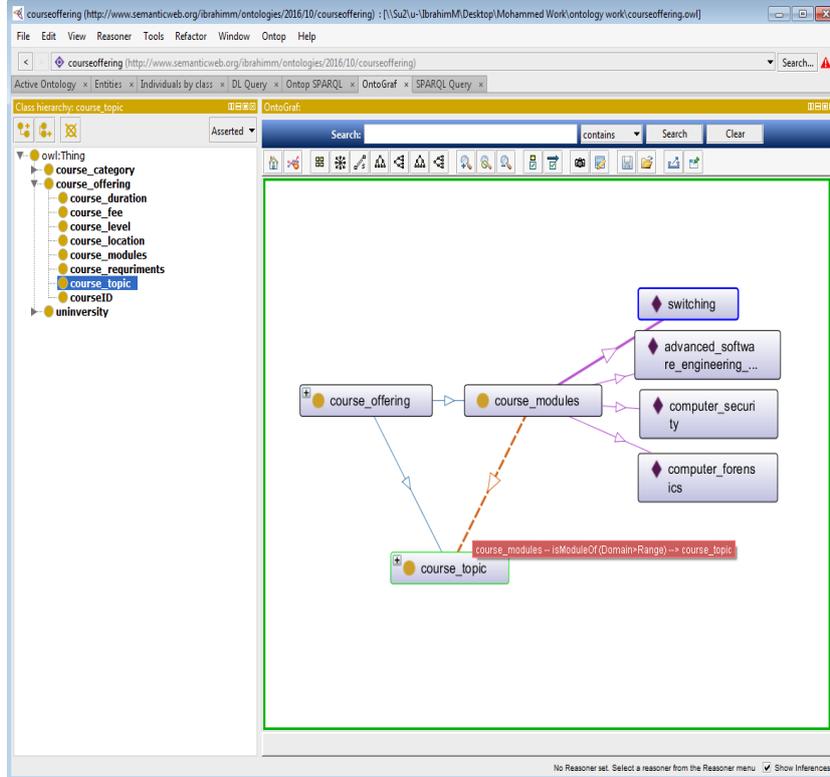


Fig. 4. Ontology represent of course topic and course modules

We define “concept distance” in order to measure concept-level similarity. Suppose that two instances x , y are concept A and B , respectively. This can be referred to as $A(x)$, $B(y)$. The concept distance between x and y , referred to by $\text{ConceptDistance}(x, y)$, is defined as follows:

$$\text{ConceptDistance}(x, y) = \begin{cases} 0 & A \equiv B \\ P(A, B) & A \sqsubseteq B \text{ or } B \sqsubseteq A, \\ P(A, B) + k & A \not\sqsubseteq B, B \not\sqsubseteq A \\ \infty & A \sqcap B = \perp. \end{cases} \quad (1)$$

While $P(A, B)$ means that the length of the path between concept A and B , according to the computed concept hierarchical tree, k is a penalty item and so is always given a positive number. If the concept distance of two instances is bigger, then naturally, the likelihood of it being the same would be less.

We also test the object properties of instances so as to compute “context similarity”. Object properties enable users to create specific relations between instances. Before an object property is used to link instances with semantic relations, it generally has to be

defined between concepts, with an option to specify its cardinality constraints. Moreover, an object property can be an inverse object property, as this allows the use of more flexible ways to describe ontology data. Inverse object properties are often very common among different data sources. For example, we tend to describe course modules that use a property as “has_modules”, in order to relate them to their course instances. By reasoning on the inverse properties, and checking the cardinalities on them, we can compute the context similarity between instances.

By reasoning on the inverse properties and checking their cardinalities, we can compute the context similarity between instances.

4 Semantic Course Recommendation Design

In this section, we give an overview of the Course Recommendation System, which provides the user/student with relevant course recommendations. The proposed system contains two main parts – client side and server side, as shown in Fig. 5. The components will be handled sequentially, but iterations are planned, so each component can better accommodate the needs of the next.

The first part, which is the client side, will be implemented with a web interface module. This is responsible for taking users’ queries, user information and user interactions (feedback) to the server side. A system interface will be available on a web-based user interface (WUI), which is transmitted via the Internet and viewed by a web browser program. The system data will take input from different information sources. Automated information extraction techniques will be applied and the results will consist of a list of features for each course, as well as relevant careers. All the information about the courses and users will be stored in a users’ and course profiles’ storage in the system database.

The second part of the system is the server side, which includes the following components:

- Data extraction API: we built this tool from scratch to extract specific information from UCAS website about postgraduate courses in the United Kingdom universities. At the other hand, this tool extract information about the jobs from job website in the United Kingdom.
- The ontology-based data integration component will gather course information, utilizing the web, through extraction of meta-information about the courses’ attributes and will discover how these correlate to specific users’ needs. In addition, it includes examining the ability of the new approach to data integration to translate the user’ input to specific needs, and find the relationship between course information and different career goals. Along with this it exploit contextual and social data to create a meaningful profile. The data integration will be analyzed to create more information about the whole course category, such as the discovery of the most important features, average or common feature values, and feature value to career relationship.

- Recommendation engine component. We developed a hybrid recommender filtering approach, which combines content based filtering and collaborative based filtering to increase the recommendation system's efficiency performance. We presented a personalised recommendation course, a system that makes use of representations of courses and student profiles, based on ontologies, in order to provide semantic applications with personalised services. The recommender uses domain ontologies to enhance the personalisation. On the one hand, the user's interests are modelled in a more effective and accurate way by applying a domain-based inference method. On the other hand, the matching algorithm used by our content-based filtering approach, which provides a measure of the affinity between an item and a student, is enhanced by applying a semantic similarity method.

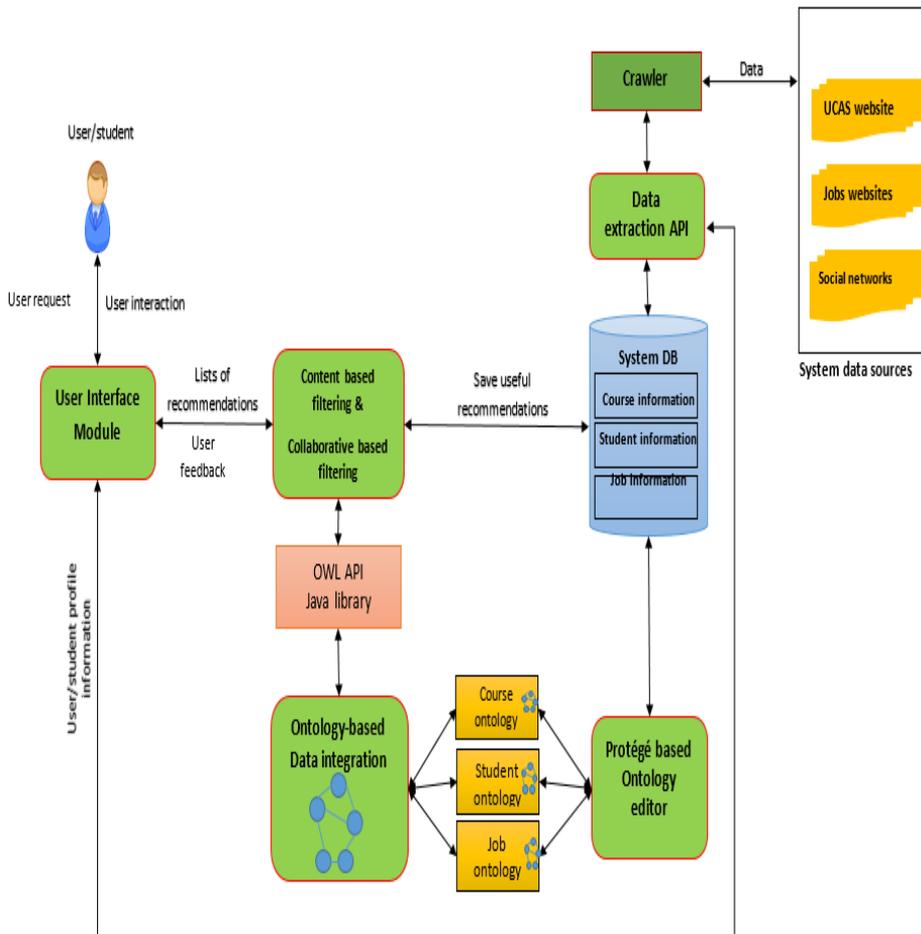


Fig. 5. Course Recommendation System Design

5 Prototype Implementation and Experiment

With the proposed recommendation approach, we built a semantic learning content recommender system. It was developed with Java (JDK1.8) and is used for compiling and executing java code. Eclipse IDE is used to edit the code, while the protégé tool 5.1.0 is used to create, edit and combine ontologies. We tested the overhead of the semantic content recommendation in terms of the response time. The experiment was deployed on a PC with 3.20GHz i5-4460 CPU and 8GB memory running Windows 7.

We have extracted data by using data extraction API from the UCAS website to implement the proposed system. This is a popular web application that provides course data in the UK. UCAS provides information on more than 78,000 under- and postgraduate courses in different fields. The information about courses includes courseID, universityID, course title, study mode, qualification, course fee, course modules, entry requirement and university location. The ontology will describe the relationship between the courses in the domain, and the ontology Protégé tool will be used to represent this.

The user/student profile will build through explicitly by asking the user to create an account on the system to build his/her profile. In the next phase, we plan to get information about the user through implicitly approach by gathering user information from a social network, such as Facebook, LinkedIn, to mention but a few. The course recommendation will be based on user profile, as depicted in Figs. 6a and b.

The screenshot shows the Eclipse IDE on the left with a project named 'courses'. The project structure includes a 'src' folder with subfolders 'controller' and 'dao'. The 'controller' folder contains several Java files, including 'CourseServlet.java', 'IndexServlet.java', 'LoginServlet.java', 'LogoutServlet.java', 'ProfileServlet.java', 'RecommenderServlet.java', and 'RegisterServlet.java'. The 'dao' folder contains 'Course.java', 'CourseDriver.java', 'CourseParser.java', 'DataIndex.java', 'DBConnect.java', 'InsertDB.java', 'Job.java', 'JobDriver.java', 'JobRecommenderIndex.java', 'JobScaper.java', 'Main.java', 'User.java', 'UserCourse.java', 'UserCourseDAO.java', and 'UserDAO.java'. The right side of the screenshot shows a web browser displaying a web application. The browser address bar shows 'http://localhost:8085/uni_scraper/indexServlet'. The page has a search bar with the text 'Find Courses/Job' and buttons for 'SIGN IN' and 'CREATE ACCOUNT'. Below the search bar is a 'Recommended Courses' section with a search input and a dropdown menu set to '10'. The table below lists recommended courses:

University	Course Title	Course Duration	Study Mode	Qualification	Fee Info	Location	Summary	Direct Link
Aberystwyth University	Computer Science	36 months	Full-time	PhD	£4,121	Penglais Campus	View Summary	Direct Link
Aberystwyth University	Computer Science	12 months	Full-time	MPhil	£3,996		View Summary	Direct Link
Aberystwyth University	Computer Science (Software Engineering)	12 months	Full-time	MSc	£6,000	Penglais Campus	View Summary	Direct Link
Aberystwyth University	Computer Science (Software)	24 months	Part-time	MSc	£3,000	Penglais Campus	View Summary	Direct Link

At the bottom of the browser window, the console shows the following output:

```
Tomcat v8.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk1.8.0_101\bin\java.exe (15 Jan 2017, 20:38:06)
Doc Score:1.0923616
*****409 : Start Record*****
Id :: 14778
Title :: Computer Science
Duration :: 6 years
*****End Record*****
Doc Score:1.0423425
*****418 : Start Record*****
Id :: 875
Title :: Advanced Computer Science (Computer Security)
Duration :: 12 months
*****End Record*****
Doc Score:0.901369516
```

Fig. 6a. Initial results for Course recommendation

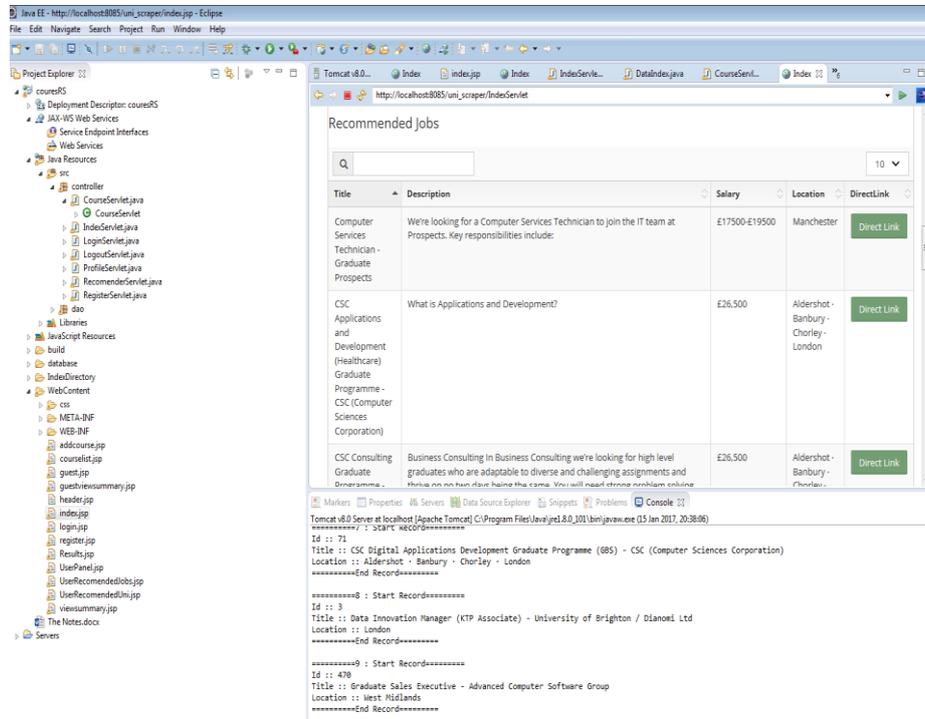


Fig. 6b. Initial results for relevant job for recommended course

6 Conclusion and Future Work

In this paper, we have proposed an ontology-based personalised course recommendation system. The use of ontology can effectively improve the quality of service of a personalised recommendation, and we have also modelled a domain ontology to support semantic interoperation between the student's profile ontology and course ontology. Our experimentation has proved that this ontology-based recommendation approach can improve the recommendation's accuracy. This approach enables e-learning systems to easily reuse and share learning objects that have been published by various systems. It uses specific ontology to infer what course a student should study and what course content a system should look for automatically

In future, we will enrich our repository by absorbing more course and user information and heterogeneous data sources. We will also further evaluate our approach and compare it with other related methods through simulation experiments that use more perspectives.

References

1. Le Roux F, Ranjeet E, Ghai V, Gao Y, Lu J.: A course recommender system using multiple criteria decision making method. In: International Conference on Intelligent Systems and Knowledge Engineering. Atlantis Press, Chengdu, (2007) . doi:10.2991/iske.2007.238
2. Apaza RG, Cervantes EV, Quispe LC, Luna JO.: Online Courses Recommendation based on LDA. In: SIMBig. pp 42-48. Peru,(2014)
3. Wang SC, Tanaka Y.: Topic-oriented query expansion for web search. In: Proceedings of the 15th international conference on World Wide Web, pp. 1029-1030. ACM, New York (2006). doi: 10.1145/1135777.1135999
4. Zhang ZK, Zhou T, Zhang YC.: Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs. J. Physica. A: Statistical Mechanics and its Applications 1; 389(1):179-86 (2010)
5. Huang CY, Chen RC, Chen LS.: Course-recommendation system based on ontology. In: International Conference Machine Learning and Cybernetics (ICMLC). pp 1168-1173. IEEE, Tianjin (2013). doi: 10.1109/ICMLC.2013.6890767
6. Yang H, Cui Z, O'Brien P.: Extracting ontologies from legacy systems for understanding and re-engineering. In: Computer Software and Applications Conference, pp. 21-26. IEEE, Phoenix (1999). doi: 10.1109/CMPSAC.1999.812512
7. Gruber TR.: A translation approach to portable ontology specifications. Technical report, Knowledge acquisition (1993)
8. Meehan K, Lunney T, Curran K, McCaughey A.: Context-aware intelligent recommendation system for tourism. In: Pervasive Computing and Communications IEEE International Conference. pp. 328-331. IEEE, San Diego (2013). doi: 10.1109/PerComW.2013.6529508
9. Asabere NY.: Towards a viewpoint of Context-Aware Recommender Systems (CARS) and services. International Journal of Computer Science and Telecommunications. pp 10-29 (2013)
10. Burke R.: Hybrid recommender systems: Survey and experiments. J. User modelling and user-adapted interaction. pp. 331-370. Springer, (2002) doi:10.1023/A:1021240730564
11. Ricci F, Rokach L, Shapira B.: Introduction to recommender systems handbook. Springer US (2011)
12. Balabanović M, Shoham Y.: Fab: content-based, collaborative recommendation. J Communications of the ACM. pp. 66-72. New York (1997). doi: 10.1145/245108.245124
13. Adomavicius G, Tuzhilin A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. J. IEEE transactions on knowledge and data engineering, IEEE, pp. 734-49. (2005). doi: 10.1109/TKDE.2005.99
14. Yu Z, Nakamura Y, Jang S, Kajita S, Mase K.: Ontology-based semantic recommendation for context-aware e-learning. In: International Conference on Ubiquitous Intelligence and Computing. UIC 2007. LNCS, vol 4611, pp 898-907. Springer, Berlin Heidelberg, (2007). doi: 10.1007/978-3-540-73549-6_88

15. Tsai KH, Chiu TK, Lee MC, Wang TI (2006) A learning objects recommendation model based on the preference and ontological approaches. In: *Advanced Learning Technologies Sixth International Conference*. pp. 36-40. IEEE, Kerkrade (2006). doi: 10.1109/ICALT.2006.1652359
16. Farzan R, Brusilovsky P.: Social navigation support in a course recommendation system. In: *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. AH 2006. LNCS, vol.4018, pp. 91-100. Springer, Berlin Heidelberg (2006). doi: 10.1007/11768012_11
17. Salton G, Buckley C.: Term-weighting approaches in automatic text retrieval. *J Information processing & management*, Elsevier, pp 513-23 (1988). doi: 10.1016/0306-4573(88)90021-0
18. Gusfield D.: *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge university press, (1997)
19. Joachims T.: Text categorization with support vector machines: Learning with many relevant features. In: *European conference on machine learning*. ECML-98. ECML 1998. LNCS, vol.1398, pp. 137-142. Springer, Berlin Heidelberg(1998).doi: 10.1007/BFb0026683
20. Cardoso J.: The semantic web vision: Where are we?. *J. IEEE Intelligent systems*. (2007). doi: 10.1109/MIS.2007.4338499
21. The Universities and Colleges Admissions Service in United Kingdom. <https://www.ucas.com/>
22. Ameen A, Khan K.U.R., Rani B.P.: Ontological student profile. In *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, pp. 466-471. ACM (2012). doi: 10.1145/2393216.2393294