DEVELOPMENT OF A WEB-BASED STUDENT ENQUIRY SYSTEM: INTEGRATION OF ONTOLOGY AND CBR

Seung Hwan Kang\textsuperscript{1}, Sim Kim Lau\textsuperscript{1}

Abstract
This paper discusses the development of a web-based student enquiry system using techniques in case-based reasoning combined with advances in ontology. The system functions as an online enquiry front-desk to allow prospective students to find out if they are eligible for admission to a post-graduate course in an Australian University. Ontology is applied in the system to ensure explicit specifications of concept and definition of terminology can be achieved in the web-based system. The system applies case-based reasoning technique to handle the query.

1. Introduction

Advances in the Internet and the World Wide Web (WWW) have resulted in proliferation of web-based systems in various business functions in organisations. Examples of e-commerce solutions that have been deployed in the business organisations include online-ordering such as shopping cart system, e-auction, e-marketplace and supply chain integration. As time and space factors no longer become an impediment in the Internet age, more and more organisations are embarking to develop web-based systems to either supplement or complement their business functions. The development of the WWW has also resulted in faster information dissemination.

As with any knowledge-based model, it is not easy to formalise knowledge, especially in the distributed networked environment such as the WWW. The driving factor in the proliferation of the WWW is ‘freedom from centralised control’ \cite[p.4]{6}. However this very nature of freedom also means that it presents difficulty in representing and formalising the structure of the information and knowledge in the Web. The inconsistency in knowledge representation can result in community of practice using different vocabularies and terminologies, thus problems such as synonym and homonym can arise. The ability to associate proper meaning to the content of the knowledge becomes increasingly important in the WWW environment. One of the increasingly important areas of research conducted in recent years to overcome this problem can be found in the area of the Semantic Web and ontology.

This research also investigates the application of case-based reasoning (CBR) in a web-based system. CBR is an Artificial Intelligence technique that allows new problem to be solved based on

\textsuperscript{1}School of Economics and Information Systems, University of Wollongong, Australia
similar problems in the past [9, 13]. It is a technique that is similar to the way humans solve problems based on one’s past experiences by applying and sometimes adapting the solution from old problems to new ones. The application of the CBR enables the problem solver to become more proficient in dealing with a wide range of problems over time.

The remainder of the paper is organised as follows. Section 2 provides overview of the ontology and the CBR. Section 3 discusses the prototype development issues. Section 4 gives three scenarios to demonstrate the application of the CBR to the system. Finally section 5 concludes the paper.

2. Background

2.1. Ontology

Ontology is an emerging research area that attracts research interest amongst researchers in Information Systems (IS), particularly its applications on the Semantic Web. Gruber, defines ontology as “a formal explicit specification of a shared conceptualisation” [5]. It can be used to define machine readable terms of the relevant concepts in the domain knowledge, and to capture agreed concepts that exist in the community of practice [3, 4, 7, 11]. It is also applied as a form of meta-knowledge and to allow conceptualisation to be structured in systems such as knowledge-based or knowledge management systems.

Generally, ontology can take several forms and structures - from simple to complex. A simple ontology can take the form of a simple hierarchical taxonomy, whereas a complex ontology can involve metadata schema and logic theory [11]. When formal ontologies are designed, they must satisfy the criteria of clarity, coherence, extendibility, minimal encoding bias and minimal ontological commitment [5]. The first criterion of clarity means that the ontology must be effective in defining and communicating the intending meaning of the term or concept it represents. Coherence and extendibility respectively refer to the inferences must be consistent and have the ability to infer new term or concept from existing definitions. Minimal encoding bias refers to the concepts be specified at the knowledge level without referencing to a particular implementation platform. Finally minimal ontological commitment means the ontology should be specified based on weak theory; it should specify as many possible models and not become too specific about the domain it intends to describe or define, thus allowing freedom for ontology commitment.

The concept of ontology can be applied as an interoperable interpretation approach. The objective is to enable reusable knowledge, meta-knowledge and sharing of knowledge to be achieved. The ability to share common understanding of the structure of information among people or software agents has been cited as one of the common goals of using ontology. This is particularly true in situations where interoperability over distributed environments require agreement on the definitions of terms, so that people and software agents share the same definitions of terms with the same meanings. It consequently minimises ambiguities amongst the use of various terms.
Case-based reasoning refers to a problem-solving paradigm that relies on case representation. It is defined as a conceptualised part of knowledge representing past experience [9, 13]. Case representation in a CBR system includes a detailed problem description and a detailed solution description. Within a case representation, most types of data can be stored in a case. These include stored data in a relational database, photographs, sound, and video. When planning for which information should be in a case, it is important to consider the functionality of the information and the ease of acquisition of the information. In fact, the CBR is dependent on the structure and the collected case in a case repository. Therefore it is important to have a mechanism that organises information that can be retrieved when it needs. Case representation also should have a standardised mechanism that is supportable, suitable and appropriate for case retrieval.

There are four phases in the CBR cycle: retrieve, reuse, review and retain [1, 9, 13]. Figure 1 shows the four phases based on Watson [13, p.17]. The retrieval phase is used to decide which case in the case repository is similar to the target case, which is the current problem to be solved. When the case that is the most similar to the target case is found, then the CBR system retrieves the matched case that can provide a detailed solved problem description to the problem. The two most widely used techniques of case retrieval are: nearest-neighbour retrieval and inductive retrieval. Nearest-neighbour retrieval is a technique used to measure how similar the target case is to a source case. It processes retrieval of case using the comparison approach of a collection of weighted attributes in the target case to the source case in the CBR case repository. If there is no matched case it will return the nearest-matched source case that may provide a detailed solved problem description to a new problem.

Inductive retrieval is sometimes used in the retrieval phase. It is a technique used to extract rules or construct decision trees from past cases. This technique processes a target case based on indexed source cases, which are normally indexed by keywords and stored into a set of cases. Then the set of cases are divided into a decision tree structure. If the target case is not found in the decision tree at runtime, then the CBR system may not retrieve a source case. Literature suggests the use of a combination of these two techniques in which inductive retrieval is used to retrieve a set of matching cases and then the nearest neighbour retrieval technique is used to rank the cases in the set according to their similarity to the target case [1, 13].
The reuse phase is to use the solution from the retrieved case to solve the problem. The solution from the matched case can be used without modification, or adaptation may be applied to revise the solution, so that it can be applied to match the new problem. This is the revise phase which verifies and evaluates the solution to match the correctness of the solution. Adaptation is a technique to alter the retrieved case to produce a new solution to a new problem. It implies that the solution of the retrieved case can be changed so that it can be presented to suit the new use. The purpose of case adaptation is to improve the CBR system’s overall problem solving ability using newly introduced cases for future use. The two most widely used techniques of case adaptation are: structural adaptation and derivational adaptation. Structural adaptation is a technique to apply adaptation rules or formulas directly to the stored solution. Once a case has been applied by the adaptation rules or formulas, the CBR system adapts the solution as a match with the new problem. On the other hand, derivational adaptation is a technique used to reuse the rules or formulas that generated the original solution to produce a new solution to the current problem. Once the verification is completed, the target case with its solution will be retained in the case memory. The revised solution is stored in the new case in the case repository so that it can be used for future reference. This is the retain phase of the CBR cycle.

3. The Development of the Web-based Enquiry System

A prototype web-based enquiry system has been developed that deals with admission query to a postgraduate IS course. The course is a very popular course for overseas students. The admission officer often receives the enquiry via telephone, email, or mail from prospective students from different countries about the course and they want to find out whether they are eligible to apply. The requirement of the system is to allow anyone to enter their details on the system and it will advise whether they are qualified for admission to the course. If they satisfy the admission criteria, then they are advised to submit a formal application for formal assessment. As a large majority of the applicants are from overseas, a web-based system that allows the applicants to access the enquiry system from the WWW is required.
The general admission criteria for the course is the completion of an equivalent undergraduate degree and completion of a programming subject at tertiary level. The decision making process for admission under general criteria is easy and straight-forward. However these are not the only criteria used to admit students to the course. There are other criteria that can be considered under special case admission. The special case admission is evaluated under the category of recognition of prior learning and relevant Information Technology (IT) working experience. For example, an applicant who has limited academic qualifications such as not possessing a Bachelor degree qualification but has extensive relevant IT working experience can apply under the category of special case admission. In another instance, if an applicant does not have an equivalent undergraduate degree, but possesses extensive relevant IT-related work experience and relevant IT professional certificates such as Microsoft Certified System Engineer (MCSE), then the applicant can apply for admission under the special case consideration too.

Very often the decision making process can be traced back to past experiences. The admission officer often looks at past cases or similar past cases to determine whether an applicant can be accepted under the special case admission. Thus the CBR can be deployed in the decision making process. Past successful application details are stored in the case repository. Each enquiry from prospective students is treated as a target case. The new case will be matched with the cases in the case repository and an outcome can be determined. Similarly any new case can be stored in the case repository for future references. This way corporate memory based on past cases can be retained.

3.1. Development of Ontology

In defining the admission criteria, there exists ambiguity in the definitions of some terminologies such as equivalent undergraduate qualification and related IT-working experience. We propose to use ontology as a means to capture agreed concepts and define the terms and their meanings accordingly. For example how does one define IT-related working experience? When an applicant says s/he has IT-related working experience, does the job title reflect the job function? Currently, the ontology in the system includes sixty-eight IT-related job titles. These job titles are derived from the career descriptions listed in the Australian Computer Society (ACS) website [2]. These job titles are well defined and are commonly used in the industries, Universities and government organisations. It enables the system to identify and define a well-agreed description of the IT jobs. If a user were to select a job title as systems analyst, then the job functions must match with that identified by the ACS as “a systems analyst will require at least five years experience in software development and programming, with at least two or more years in systems analysis” [2]. In this way, confusion regarding various IT job title as well ambiguity in the IT job description can be resolved.

We have used RDF (Resource Description Framework) to implement the concepts in the ontology. The RDF schema is an extension of describing vocabulary in a semantic way that allows groups of related resources and the relationships between these resources to be described [12]. This design approach aims to improve the functionality of the Web by providing more flexible and adaptable information identification. Figure 2 shows an example of using containers to refer to a collection of resources. There are three types of container objects: bag, sequence and alternative [10]. A bag is an unordered list of resources, a sequence is an ordered list of resources and alternative is a list of
resources that represent alternatives for a single value. Figure 2 shows an instance of a container object of bag. The property “type” in the diagram means that it is declared as an instance of the bag container object type [10]. The membership relation between this bag container resource and other resources are named as “_1”, “_2”, “_3” and “_4”. Here it shows the resource is job titles described by the ACS, and is identified by the URI: “http://.../JobTitles/ACS”. That resource has a rdf:type property whose value is rdf:Bag and a subclass of one of rdf:Bag. The first member of the container is the value of the container's rdf:_1 property; the second member of the container is the value of the container's rdf:_2 property and so on. In the prototype system, there are sixty-eight members of the bag container. The notation that were used here are: IT for IT-related jobs, NONIT for NON IT-related job, JT for Job Title, JD for Job Description, DUTY for job Duty, and CP for Career Path. We have also used similar representation to show IT professional certificates in Figure 3.

Figure 2 A partial sample RDF representation for the IT job titles

Figure 3 A sample RDF representation of IT professional certificates
3.2. Application of the CBR

There are two types of cases in the case repository: the original case and the adapted case. The original case refers to the case that was given by the admission officer before the system is developed. In this prototype, the admission officer has provided fifteen original cases that represent successful admission applications to the course. On the other hand, the adapted case is one that has been adapted by the system as a result of the revise and retain phase in the CBR cycle. In the retrieval process, the original case has precedence over the adapted case. The system will first attempt to find an existing original case that matches with the target case. If a matched original case is found, then the system will return the outcome based on the solution component of the original case. Otherwise, it will attempt to find a matched adapted case. If one is found, then the solution of the matched adapted case will be returned. If none of the matched original and adapted case is found, it will process similarity assessment and the case that is found to be closely matched to the target case will be retrieved from the case repository. Any new case may be stored in the case repository for future reference.

4. Illustration of the System

The prototype system is developed in the Semantic Web environment using XML and RDF. Each enquiry is presented as a target case. Each application is stored as problem description in the case and the outcome of the application (i.e. whether s/he is admitted to the course) is the solution to the case. As discussed previously, there are two types of cases in the case repository: the original and the adapted cases. These cases are stored in the MySQL database management system. Each user enquiry is stored in XML format. Then a software agent is used to transform the XML file to the RDF file using the eXtensible Stylesheet Language Transformations (XSLT). We do this so that only relevant information that is referenced to the ontology is extracted. The ontology in the system is developed in RDF format. Thus, it is necessary to transform the user query from XML to RDF format so that reference can be made to the ontology. To implement the XML conversion, Xerces Java 2 parser is used. The reason for using Java 2 Platform is that it provides support of XML features such as Document Object Model (DOM), Simple API for XML Parsing (SAX) and XSLT. RDQL (RDF data Query Language) of Jena Semantic Web Toolkit is also integrated to query the RDF files because it includes built-in support for RDF containers, integrated RDQL, and support for storing ontology in a memory model [8]. We will use the following three scenarios to demonstrate the application of the CBR to the system.

4.1. A Matched Original Case

This scenario is used to demonstrate the retrieval process for the original case. When an original case is found to match with the target case of the user query, then the solution found in the original case will be applied and a recommendation is made. A sample screen output for this scenario is shown in Figure 4. For the purpose of presentation in this paper, we have produced the output as shown in Figure 4 to demonstrate the result. In the actual prototype, the user will only see a message
that advises the applicant whether s/he is eligible or ineligible to apply. A sample of the message is as follows: "You have enquired about the Master of Information Systems course. From the information you have provided, we advise that you are eligible to apply for the course, please submit a formal application to the University". Note that a formal application is required because the applicants need to provide documented proof of academic qualifications, IT professional certificates or supported documents for working experiences. These documents need to be verified and checked manually.

A sample query for this scenario is: "I have a qualification of a three-year Bachelor of IT and I have professional certificates that include Microsoft Certificate Professional Systems Engineer. Will I be eligible to admit to the Master's course in Information Systems?" Based on the original cases that were stored in the case repository, in this instance it is found that case 15 matches with this user query and the solution that was stored in this case will be retrieved and apply to the query. The expected output is the applicant satisfies the admission criteria and thus is eligible to apply for admission to the course. This example demonstrates the use of existing knowledge (original case) to solve new problem (target case).

![Project Prototype](image)

**Figure 4** A matched original case

### 4.2. A Matched Adapted Case

This scenario is used to evaluate the case retrieval process when an original case is not found to match the user query. In this instance, the system will attempt to find a matched adapted case. A sample user query for this scenario is as follows: "I have a qualification of a three-year Diploma
majoring in IT and I have a professional certificate of Microsoft Certificate Professional Systems Engineer. I have worked as a programmer for two years. Will I be eligible to admit to the Graduate Diploma course in Information Systems?" In this example, there is no original case that is found to match with the target case. For demonstration purpose in this paper, we have included the result “no original case is found” in Figure 5. Then the system will try to look for a matched adapted case stored in the case repository. In this example, an adapted case is found to match with the target case. So the solution from the adapted case is returned. A sample output of this scenario is shown in Figure 5. This example demonstrates re-use of knowledge that has been retained in the case repository to solve new problem.

![Image](image.png)

**Figure 5** A sample output of matched adapted case

### 4.3. A Best-Matched Case

The third scenario is when none of the case in the case repository is found to match with the user query. In this situation, the system performs similarity assessment, and the case that is found to be the nearest to the target case is retrieved. This process is determined by the lowest value calculated using the following equation [13, p.28]:

\[
\text{Similarity}(T, S) = \sum_{i=1}^{n} f(T_i, S_i) \sum W_i; \text{ where } T \text{ is the target case, } S \text{ is the source case, } n \text{ is the number of attributes in each case, } i \text{ is an individual attribute from 1 to } n, f \text{ is a similarity function for attribute } i \text{ in cases } T \text{ and } S, \text{ and } W \text{ is the importance weighting of attribute } i. \text{ The equation calculates the sum of similarity of the target case to the source case for all attributes multiplied by the importance weighting of individual attributes. It is worth noting that the weightings of each attribute show the importance of the attribute in decision-making. In the system, the admission officer determines the weightings in which the criteria of qualifications of the}
\]

213
applicants and the number of years of the qualifications are considered to be twice as important as other attributes. Table 1 shows a sample list of attributes used in the prototype.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Domain (all possible values for the attributes)</th>
<th>Weighting</th>
<th>Integer Value (assigned to the attribute)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Graduate Diploma</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Qualification</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Diploma</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Bachelor</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Master’s</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1-4</td>
<td>Use the same value as the input values</td>
<td>Use the same value as the input values</td>
<td></td>
</tr>
<tr>
<td>IT related</td>
<td>Yes</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Professional</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Certificate</td>
<td>Nil</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCJP, MCP+I, MCSE, MCSE+I</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Occupation</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Nil</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-IT related career</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IT-related career</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Experience</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0-10</td>
<td>Use the same value as the input values</td>
<td>Use the same value as the input values</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 A sample of tabulated attributes of prototype

A sample user query for this scenario is as follows: “I have a qualification of a three-year Bachelor degree majoring in IT and I have a Microsoft Certificate Professional Systems Engineer certificate. I also have five years working experience as an IT manager. Am I eligible to admit to the Master of Information Systems course?” Figure 6 shows a sample output of the value computed using the above equation of similarity assessment. For the purpose of demonstration in this paper, we have included the result to show no original case or adapted case is found. In addition, the calculated similarity assessment values for all original cases have been included too. It is worth pointing out that cases numbered 5, 6, 8 and 9 should not be included in the calculation as these cases are past cases that are related to the Graduate Diploma course and thus are not relevant to this query. In this
example, only cases numbered 11 to 14 are found to be equally close to the target case. Thus, the solution from any of these cases will be revised and verified to return the outcome for the target case. Once the solution for the target case is verified and evaluated to match the correctness of the admission criteria, then the target case with its case description and solution will be retained and stored in the case repository as the new adapted case.

5. Conclusion

The flexibility of the CBR technique allows the system to learn from past experiences and gain new knowledge in the form of adapted cases retained in the case repository. This way the system is able to evolve through the adaptation of new cases and solve new problems by making use of previous similar situations and reusing information and knowledge gained through the process. We have applied the concept of ontology as a means to provide an agreed set of definitions and terms in the domain. It provides a very useful way for the community of practice to define agreed concepts and terminologies in the system.
6. References


