

2003

A framework for case-based reasoning integration on knowledge management systems

Seung Hwan Kang
Payap University, seung_h@payap.ac.th

Sim K. Lau
University of Wollongong, simlau@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/commpapers>



Part of the [Business Commons](#), and the [Social and Behavioral Sciences Commons](#)

Recommended Citation

Kang, Seung Hwan and Lau, Sim K.: A framework for case-based reasoning integration on knowledge management systems 2003, 1327-1343.
<https://ro.uow.edu.au/commpapers/2896>

A framework for case-based reasoning integration on knowledge management systems

Abstract

To support the sharing and reusing of well-defined knowledge among knowledge management systems, it is useful to use standardised formalisation. It is also common effort to difficulty of knowledge acquisition known as knowledge acquisition bottleneck. In this paper investigates the feasibility of using techniques in case-based reasoning of artificial intelligence for the knowledge acquisition phase in knowledge management systems. The need of an ontological approach of the semantic web for well-defined set of domain knowledge is proposed in order to avoid knowledge acquisition bottleneck. Our viewpoint of this approach is that the ontology-driven mechanism allows us to provide standardised structured vocabularies and conceptualisation of knowledge domain. Over the standardised platform, we see an alternative to share and reuse homogenous information and knowledge in the knowledge management systems.

Keywords

knowledge, systems, framework, reasoning, management, integration, case

Disciplines

Business | Social and Behavioral Sciences

Publication Details

Kang, S. & Lau, S. K. (2003). A framework for case-based reasoning integration on knowledge management systems. In J. Hanisch, D. Falconer, S. Horrocks & M. Hillier (Eds.), 7th Pacific Asia Conference on Information Systems (pp. 1327-1343). Adelaide, SA: University of South Australia.

A Framework for Case-based Reasoning Integration on Knowledge Management Systems

Seung Hwan Kang and Sim Kim Lau

School of Economics and Information Systems, University of Wollongong
Northfields Avenue, Wollongong, NSW, Australia 2500
sk33@uow.edu.au

School of Economics and Information Systems, University of Wollongong
Northfields Avenue, Wollongong, NSW, Australia 2500
simlau@uow.edu.au

Abstract

To support the sharing and reusing of well-defined knowledge among knowledge management systems, it is useful to use standardised formalisation. It is also common effort to difficulty of knowledge acquisition known as knowledge acquisition bottleneck. In this paper investigates the feasibility of using techniques in case-based reasoning of artificial intelligence for the knowledge acquisition phase in knowledge management systems. The need of an ontological approach of the semantic web for well-defined set of domain knowledge is proposed in order to avoid knowledge acquisition bottleneck. Our viewpoint of this approach is that the ontology-driven mechanism allows us to provide standardised structured vocabularies and conceptualisation of knowledge domain. Over the standardised platform, we see an alternative to share and reuse homogenous information and knowledge in the knowledge management systems.

Keywords

Knowledge management systems, knowledge acquisition, case-based reasoning, ontology, the semantic web

1. Introduction

Knowledge management systems (KMS) are gaining interest among organisations, and the KMS can provide support for knowledge-sharing within communities-of-practice (Walsham 2001). In general, KMS is a system that designed specifically to provide the sharing and integration of knowledge (Alavi and Leidner 1999). Due to downsizing and outsourcing in the recent years, a lot of organisations have come to realise that they have lost core competency knowledge within the organisation as people leave their jobs (Hildreth et al. 1999). Globalisation of organisations has also accelerated the needs of knowledge sharing in different geographical locations.

To support Knowledge Management (KM), the process of knowledge acquisition is considered as an important phase in the development of KMS. Brule and Blount (1989) define knowledge acquisition as the process of translating implicit knowledge into explicit form. There are two types of knowledge: explicit and tacit knowledge (White 2000). Explicit knowledge can be easily written down in procedural form and taught to others, whereas tacit knowledge is the knowledge that cannot be easily described in procedural form. In particular tacit knowledge is often resides with the expert, and the expert may or may not be able to expressively describe his or her knowledge explicitly. Very often, tacit knowledge is carried out as heuristic or rule-of-thumb. In general, transferring of knowledge, in particular tacit knowledge, to the corporate knowledge repository is not an easy process.

This paper explores the possibility of applying Case-Based Reasoning (CBR) techniques using an ontology-driven framework to allow knowledge sharing and re-using in KMS in a semantic web environment. The basic idea of adapting CBR in KMS is to store knowledge in the case repository and to retrieve knowledge from past cases and apply it in a new set of problem that is similar to the existing ones. CBR reduce knowledge acquisition effort through the reuse and adaptation of past cases, it allows knowledge sharing and reuse to be made possible through the adaptation of new knowledge using new case behaviour and reusing of knowledge from existing cases. In our project, we use agents to achieve this. We use ontology to provide a framework for conceptualised knowledge representation in the form of standardised structured vocabularies and structured conceptualisation of knowledge domain. This allows homogenous information and knowledge to be stored in the KMS. Wache et al. (2001) point out that using ontology can prevent generation of heterogeneous information and it allows well defined, agreed, and organised information to be used in knowledge representation. By representing knowledge in a case, it provides an opportunity whereby the retrieval process can identify the case with the most similar problem description in the past cases when new problem arises. This way, newly adapted knowledge can be shared and reused in the knowledge repository without inconsistency.

2. Domain

The domain of our research is a student admission enquiry system that allows students to find out if they are eligible for admission to the Graduate Diploma of Information Systems or Masters of Information Systems courses in a local Australian university. The Graduate Diploma course is specifically designed for those who hold tertiary qualifications in areas not related to the discipline of Information Systems (IS) and for students who wish to gain initial essential education in IS. General admission requirement for this course include: a university degree or equivalent and completion of at least the equivalent of one introductory computer or programming subject at tertiary level. Currently, the Admission Officer in the Department assesses the eligibility of each applicant to the course manually. Other than the head of the Department, the Admission Officer is the only one in the Department who has the knowledge about the admission criteria. A majority of the applicants who seek to enrol in this course are with overseas qualifications from China, Indonesia, Thailand, India, Pakistan and other countries. One of the problems encounter is how does one define a qualification from country X is equivalent to the local Australian university degree. Very often, the academic qualification from each country is different. For example, is a 4-year tertiary degree from country X and a 3-year tertiary qualification from country Y equivalent to a 3-year

undergraduate degree in Australia? One of the methods to determine the degree from a country is equivalent to the local tertiary qualification is that the student must have completed twelve years of schooling before they start their degree or diploma in their country. This is deemed necessary because not all overseas applicants who have a degree qualification from their country have completed twelve years of schooling prior to the commencement of their university courses. Currently the Admission Officer has some form of tacit knowledge about educational qualifications from different countries because the Admission Officer has more than ten years of experience in this aspect.

To complicate the matter, the general admission criteria are not the only criteria used to admit students to the course. There are other criteria that can be used to determine the applicants' eligibility for admission to the course. For example, applicant who do not possess the equivalent university degree, but have qualifications such as Diploma in Information Technology (IT) or Diploma in Computer Science (CS) with relevant working experience in IT can be admitted to the course. Similarly, students who have professional IT certificate such as Microsoft Professional Certificate can be admitted to the course as well. When this occurs, the Admission Officer will consult the head of the Department and based on past experiences and previous cases of students' admission, the Admission Officer will determine the case individually based on individual's academic qualifications, professional certificate and employment history. Figure 1 shows a list of special cases in which the students can gain admission to the course.

1. 3-years Diploma (after 12 years schooling) + relevant work experiences of at least 2 years from non-IT related courses.
2. 3-years Diploma (after 12 years schooling) + professional certificate such as: Microsoft Certificate Professional Systems Engineer; Microsoft Certificate + Internet, Etc. from non-IT related courses.
3. 3-years Diploma (after 12 years schooling) majoring in Computer Science, or IT or IS from IT related courses
4. 3-years Diploma (after 12 years schooling) + 1 year studies in other courses such MIB, or MBA with good results.
5. Degree with a major in IT, IS or CS from IT related courses
6. 2-years bachelor/diploma (after 12 years schooling) will not be accepted to postgraduate courses.

Figure 1. Admission to Graduate Diploma in Information Systems – Special Cases

The Masters of Information Systems is a postgraduate coursework program offers by the Department. This provides graduates with the opportunity to study some advanced topics in IS and to undertake a research project. To be eligible for the course, the applicants must have: (i) a degree in computing and or IS; or (ii) a degree with a major study in computing and/or IS; or (iii) a graduate diploma in computing and/or IS. As with the Graduate Diploma course, these are not the only criteria used to assess the eligibility of the students who wish to apply for the course. Students with other relevant academic qualifications and working experience as well as professional certificates can be admitted to the course. Figure 2 shows a list of special cases in which the students can gain admission to the master course.

1. Have at least two 3-years Diploma majoring in IS, Computer or Telecommunication plus relevant working experiences of more than 2 years.

2. A 3-years bachelor of honours plus professional certificate such as: Microsoft Certificate Systems Engineer; Microsoft Certificate Professional; Microsoft Certificate Professional + Internet, etc.

Figure 2: Admission to Masters of Information Systems - Special Cases

The Admission Officer often receives enquiry (via telephone, email, or mail) from prospective students from different countries to enquire about the course and to find out if they are eligible for the course. A student admission enquiry system is to be developed to assist the Admission Officer in handling the enquiries. The requirement of the system is to allow anyone to enter their details to the system and the system will advise if they are eligible for the course. If they are eligible, then they are advised to submit formal application to the University for assessment. As a large majority of the applicants are from overseas, so a system that allows the applicants to access the enquiry system from the WWW is required. It is also essential that the system can learn from experience and be able to adapt new knowledge in terms of the criteria that were used to admit a student to the course.

3. Theoretical Background

This section presents brief theoretical background on case-based reasoning, agents and ontology.

3.1 Case-Based Reasoning

In CBR, a case is defined as “a conceptualised piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoned” (Kolodner 1993 p.13). Thus, a case can be viewed as a conceptualised part of knowledge representing past experience. It is an area of AI that allows us to deal with situations that are similar to the current ones, and use them to solve new problems. The CBR cycle consists of four phases: retrieve, reuse, revise and retain (Kolodner 1993, Aamodt & Plaza 1994, Watson 1997).

The retrieval phase is to decide which case in the case repository is similar to a target case (that is current problem). When the case that is the most similar is found to the target case, 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-neighbor retrieval and inductive retrieval.

In the reuse phase, the solution from the retrieved case is used to solve the target case. Reusable case is more user-acceptable because its solution has already been accepted and convinced by the previous user. At the reuse phase, the solution from the matched case can be used without modification or adaptation may be applied to adapt the solution to match the new problem. Adaptation is a technique to alter the retrieved case to reproduce a new solution to a new problem. The retrieved case can be changed so that it can be presented to suit 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.

In the revise phase, the solution needs to be verified and evaluated to match the correctness of the solution. Once the verification is completed, the target case with its solution will be retained in the case memory. This is the retain phase of the CBR cycle. Indexing is commonly used in the case retainment phase in CBR. It allows retrieval of cases to be optimised.

In our research, we represent knowledge using cases. The use of CBR technique in this project provides an opportunity when new problem arises, the retrieval process can identify the cases with the most similar problem description from the knowledge repository. The project also aims to allow newly adapted knowledge to be shared and reused in the knowledge repository.

3.2 Agents

There are a number of definitions for agents in the literature. Agents can be described as: “special purpose” (Smith et al. 1994), “perceives and acts in its environment through preceptors” (Russell & Norvig 1995), “autonomously” (Maes 1995), “behaving its dynamic property of functions such as social ability, reactivity, pro-activeness” (Wooldridge & Jennings 1995, Hayes-Roth 1995). Agent can be classified as a software, intelligent, or learning agent. Software agent is a piece of software application that acts like human. On the other hand, intelligent agent has human-like characteristics such as autonomy, temporal continuity, reactivity, and is goal driven (Wooldridge & Jennings 1995, Maes 1995). A learning agent has an inherent characteristic of the human beings to adapt its behavior in order to improve its performance (Konar 2000). With learning capability, the learning agent adapts the abstract patterns of relationship in the domain autonomously.

Software agent can be used in KMS, and it can be developed to replace repetitive tasks such as searching databases, retrieving and filtering information, and delivering it back to the end user. For example, agent can play the role of a search engine when the user wants to search for knowledge in the KMS. When software agents are used alone without referring to the context in which they are operating, they may be of limited use. Thus the knowledge about the domain in which the agent is operating is important. To ensure that knowledge sharing and reuse can be achieved in KMS, better support for standardising conceptualisation of domains is required. To achieve this, ontology can be applied to explicitly formalise the specification of a shared conceptualisation. We aim to use ontology to capture domain knowledge in a generic way so that it provides a commonly agreed definition of vocabulary and concepts of the domain. In our research, ontology is applied to the KMS as a form of meta-knowledge. In our research, ontology is applied to the KMS as a form of meta-knowledge. The use of ontology allows consistent conceptualization to be referenced in KMS.

3.3 Ontology

Standardised conceptualisation is achievable with the use of ontology. Ontology is “a science or study of being” (Hornby 1995). Recently ontology has been widely used in AI, particularly in knowledge representation. Knowledge representation is “an explicit specification of a conceptualization” (Gruber 1993a). In general, ontology is used to conceptualise representation of a community of interest for human end-users or software agents, and it is

used to explicit formal specification of a shared conceptualization (Gruber 1993b, Gomez-Perez 1999, Sowa 2000, Hendler 2001, Berners-Lee et al. 2001). Ontology also aims at capturing domain knowledge in a generic way so that it provides a commonly agreed understanding of a domain.

Semantic web allows information to reuse and share over the Web (Berners-Lee et al. 2001, W3C 2002a, W3C 2002b). Semantic web also enables software agents to communicate with other software agents. The agents share terms of mapped or merged ontologies, which are usually defined using a language of Extensible Markup Language (XML) (W3C 2000) and the Resource Description Framework (RDF) (Goldfarb 2001, Lassila & Swick 1999). XML is a subset of Standard Generalised Markup Language (SGML) developed to response to large scale of digital publishing, and information exchange on the web (W3C 2000). In 1999, RDF model and syntax specification is to provide “machine-understandable” metadata (Lassila & Swick 1999). With RDF model, it is possible to offer a structure to assist with interoperability between software agents and web applications for the exchange of web-based human and machine-readable as well as machine-understandable information.

4. Prototype Development

4.1 Knowledge Representation

In our research, we present knowledge using cases. The applicant's details are stored as problem description in the case and the outcome of whether s/he is admitted to the course (Graduate Diploma and Masters courses) is the solution in the cases. The knowledge repository consists of two types of cases, the original cases and the adapted cases. The original cases are cases that were provided by the domain expert (the Admission Officer) and the adapted cases are cases that were retained in the knowledge repository as a result of knowledge adaptation (to be described in the later section). There are fifteen original cases in the knowledge repository (see Appendix A). The cases are stored in MySQL database management system.

Each of the user input will become a target case in the system. This target case may become an adapted case and can be stored in the knowledge repository after they have been adapted in the revise phase of the CBR cycle. Each user input (target case) is stored in XML format. Then a software agent is used to transform the XML file to RDF file using the eXtensible Stylesheet Language Transformations (XSLT). The user input needs to be transform to the RDF format because only relevant information will be extracted and used by the software agent in the retrieval phase of the CBR cycle. Secondly, the ontology is developed in RDF format, thus it is necessary to transform the user query from XML format to the RDF format so that reference can be made to the ontology.

4.2 Ontology Design

The ontology that we use in this project is simple in concept. We use ontology to define the concepts of course, qualifications, professional certificates and work experience together with their possible values as shown in Figure 6. The use of ontology allows standardized concepts

to be referenced in the system and also allowed easy maintenance of the system. For example, additional information can be added to the system by adding new entry to the ontology; such as adding new IT related certificates or adding new courses. We can also add rules or logics to the concept if required. For example, we will not accept a qualification to be considered if the student has not completed twelve years of schooling prior to the commencement of their university courses. This rule can be built into the ontology. In another instance, we can include a list of jobs as IT or non-IT related. This way when the applicant enters the job description, the system will reference to the ontology and thus determine if it is IT or non-IT related.

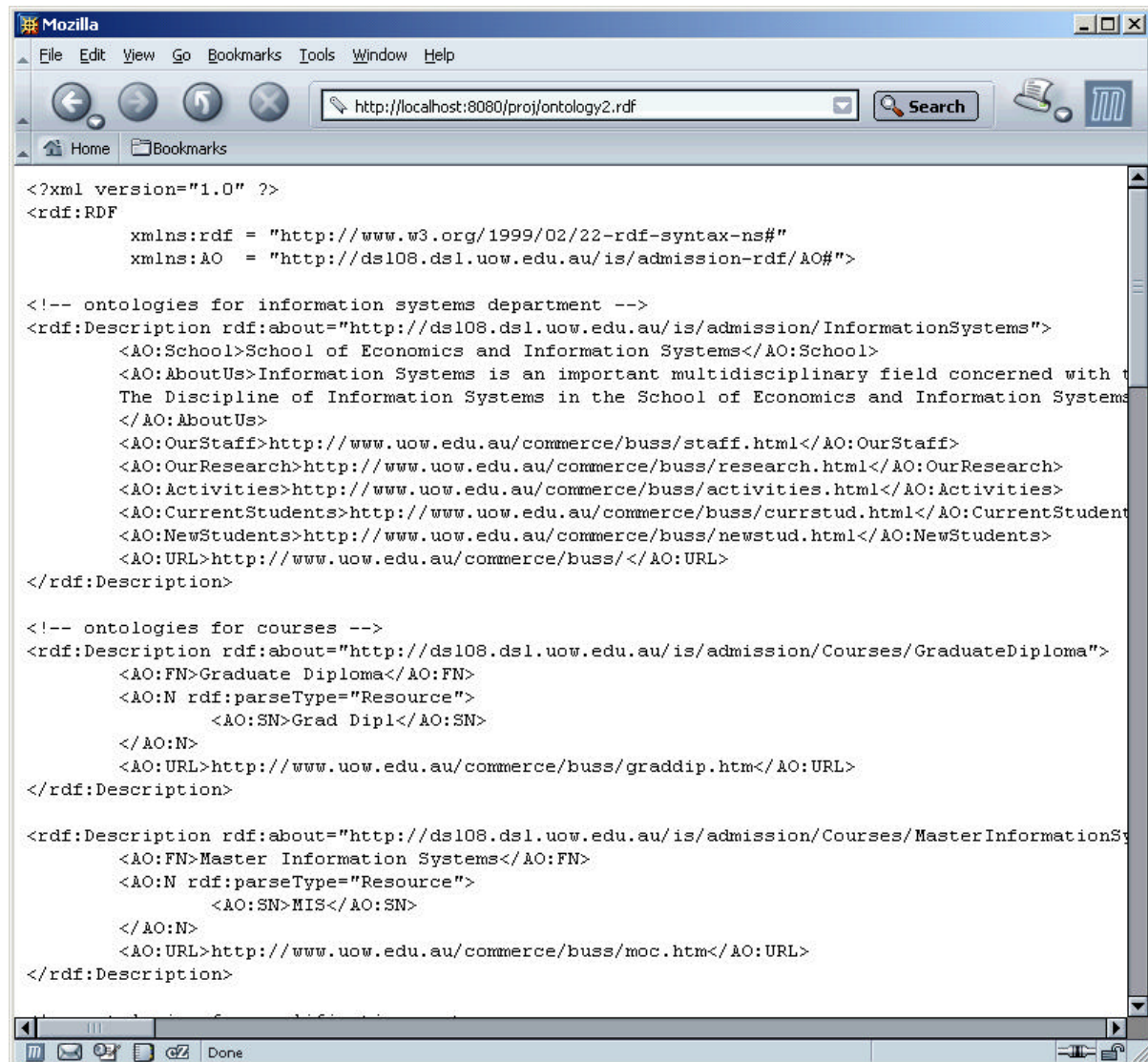


Figure 3. A Sample Output of Ontology

The ontology will also include a list of abbreviation for each of the concept or attribute. This way, the system is able to detect any abbreviation or partial names for each of the concept such as “mis” refers to “Masters of Information Systems”, “dipl” refers to “Diploma” and “mcse” refers to “Microsoft Certified Systems Engineer” (see Figure 6 & 7).

4.3 Agents

We use agents in the following tasks: retrieval, reuse and revise. Figure 4 shows a graphical view of the processes in which agents are used.

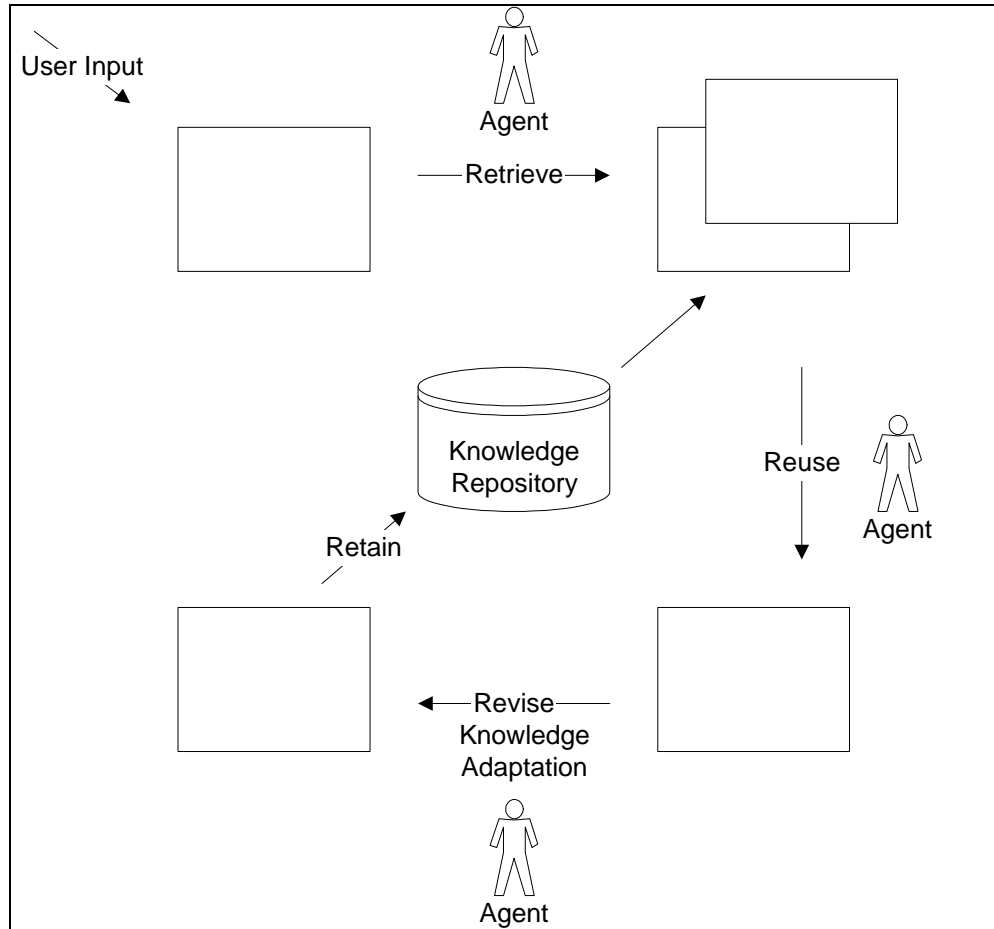


Figure 4. A Graphical View of Prototype Processes

The retrieval agent performs similarity assessment using RDQL (RDF data Query Language) to find the best or closest matched cases in the knowledge repository. The RDQL which we use in this project is derived from SquishQL by Hewlett-Packard Labs (HPL), and it is an implementation of an SQL-like query language for RDF (HPL 2002). The system will first retrieve the original case if there is one available, otherwise it will attempt to find if there is any adapted case stored in the knowledge repository that matches with the user query. If there is, then the adapted case will be used; otherwise similarity assessment will be performed. Figure 5 shows the flowchart of this process.

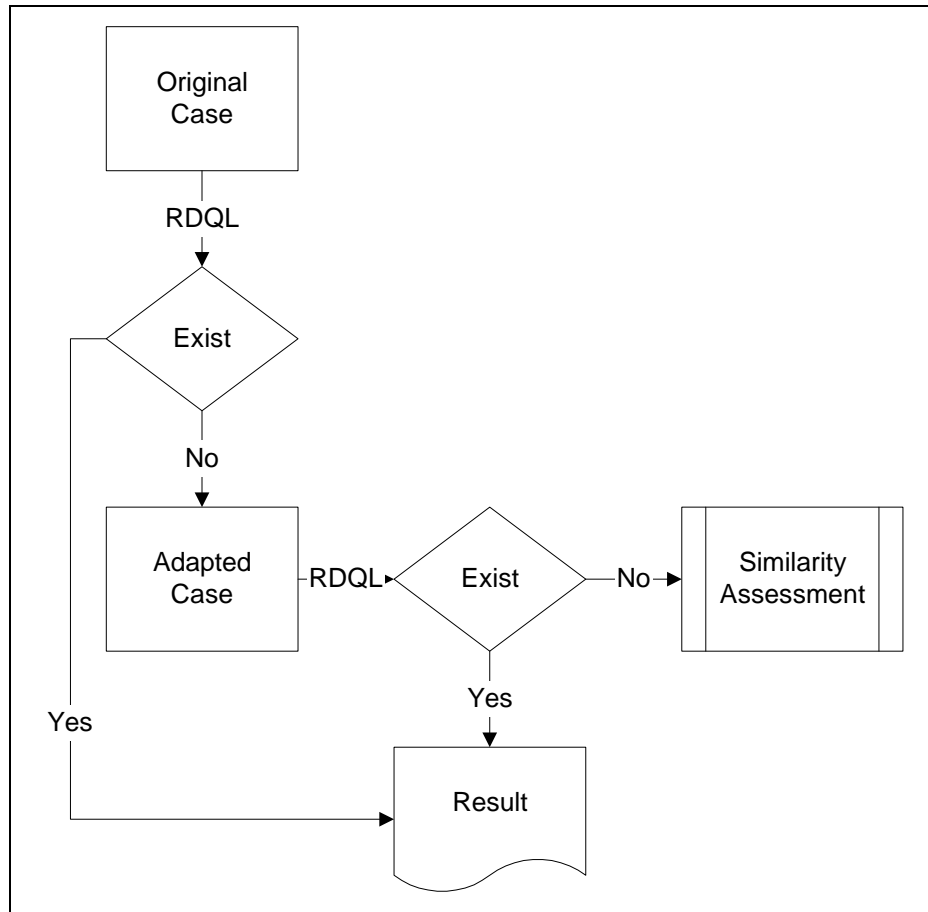


Figure 5. Decision Tree for Case Retrieval

The similarity assessment is performed using the nearest-neighbour technique. It processes retrieval of cases by comparing a list of weighted attributes in the target case to source cases in the CBR library (Watson 1997).

$$\text{Similarity } (T, S) = \sum_{i=1}^n f(T_j, S_i) \times W_i \quad [\text{Equation 1}]$$

where

T is the target case

S is the source case

n is the number of attributes in each case

i is an individual attribute from 1 to n

f is a similarity function for attribute i in cases T and S

w is the importance weighting of attribute i

Weightings are assigned to the attributes. In our prototype, the domain expert determines the weighting of each attribute. For example, a person's academic qualification is considered as twice as important compare to other attributes and is assigned weighting of 2. The weight is

assigned an integer value to simplify the calculation. The formula calculates the distance of the target case from the original case. Table 1 shows a sample of attributes for our prototype system. All attributes of cases are represented as integer values.

Attribute Name		Integer Value	Weighting
Course			1
	Graduate Diploma	1	
Qualification			2
	Diploma	1	
	Bachelor	1	
	Master	2	
Year			2
		The same value as original input, desirable range 1-4 years	As it is
IT related	Yes	1	1
	No	0	
Certificate			1
	Nil	0	
	SCJP, MCP+I, MCSE, MCSE+I	1	
Occupation			1
	Nil	0	
	Non-IT related career	1	
	IT-related career	2	
Experience			1
		The same value as original input, desirable range 2 digits, 0-10 years	As it is

Table 1. A sample of tabulated attributes of prototype

When the matched or nearest matched case is found, the reuse agent will apply the solution from the retrieved case to solve the target case by recommending whether the applicant is eligible for the course s/he intends to apply. The reuse phase of the CBR cycle also allows the solution to be modified and the target case to be adapted so that it can be retained and stored in the knowledge repository as new cases. However before cases can be retained, the case must be verified and revised using knowledge adaptation technique. In our prototype, we use derivation adaptation technique for the revise phase. Derivational adaptation is a technique to reuse the rules of formulas that generated the original solution to produce a new solution to the current problem (Watson 1997). An agent is developed in our prototype to perform this task. The agent function is developed using Jena Semantic Web Toolkit (HPL 2002). Our adaptation rule is if all basic academic requirements for admission are met and

the applicant has other professional certificates that were not included in the professional certificate list provided by the Admission Officer, the system will check for qualification of the certificates from the ontology that maintain a list of valid professional IT qualifications; if the certificate is found to be valid, then the case will be adapted and added to the knowledge repository. The prototype also has a pending sub-system in which cases that the system cannot be adapted be stored for manual review by the Admission Officer. If after the review, the Admission Officer found that the case is of relevance and it will be added to the knowledge repository manually. An example of this when the system cannot determine the academic qualification of the applicant from a new country (that is country not found in the cases).

To differentiate between cases that were originally given by the Admission Officer (known as original case) and those that were adapted using the knowledge adaptation process, we called these cases as adapted case. To optimise the retrieval process, and agent that performs indexing has been developed. The agent will index the cases in the knowledge repository every time new adapted cases are stored.

5. Sample Outputs

The following section shows some sample output from the prototype system. Figure 6 shows a sample of user input screen. As discussed only attributes that are relevant (course, qualification, professional certificate and working experience) will be extracted from the user input screen and transformed into RDF format so that agents can be applied.

Project Prototype

Admission to eg) Graduate Diploma

Personal Details

Your Name First Name Last Name

Your Contact E-mail Address

Academic Qualification

Title of Yours eg) Bachelor

Year(s) from Non-IT related course(s) ☐ IT/IS/CS related course(s) ☒

Name of Your University eg) University of Wollongong

URL of Your University eg) <http://www.uow.edu.au>

Country (origin) eg) Australia

Professional Certificate (if there is any)

Title of Yours eg) MCSE

Work Experience (if there is any)

Your Occupation

Year(s)

Figure 6. A Sample of User Input Screen

In the instance of case retrieval, if there is a matched original case in the knowledge repository, the agent will retrieve the original case. As explained in the previous section, the agent uses pattern matching to perform this task. This is shown in Figure 7. For the purpose of presentation in this paper, we have produced the output as shown in Figure 7 to demonstrate the result. In the actual prototype, the user only see a message that advise the applicant whether s/he is eligible or ineligible for admission to the course that s/he wishes 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 for the course, please submit a formal application to the University”.

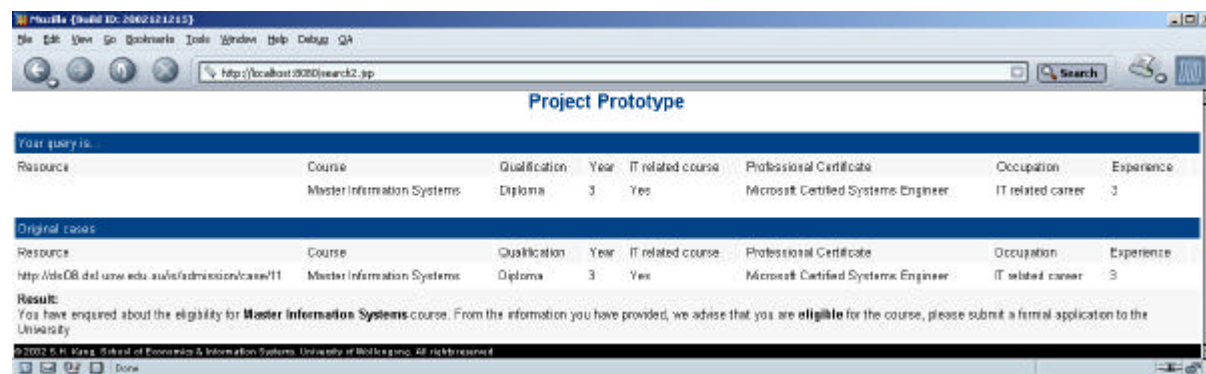


Figure 7. A Matched Original Case of User Inquiry

If no original case is matched with the user query (target case), the agent will attempt to find a matched adapted case in the knowledge repository. A sample output is shown in Figure 8.

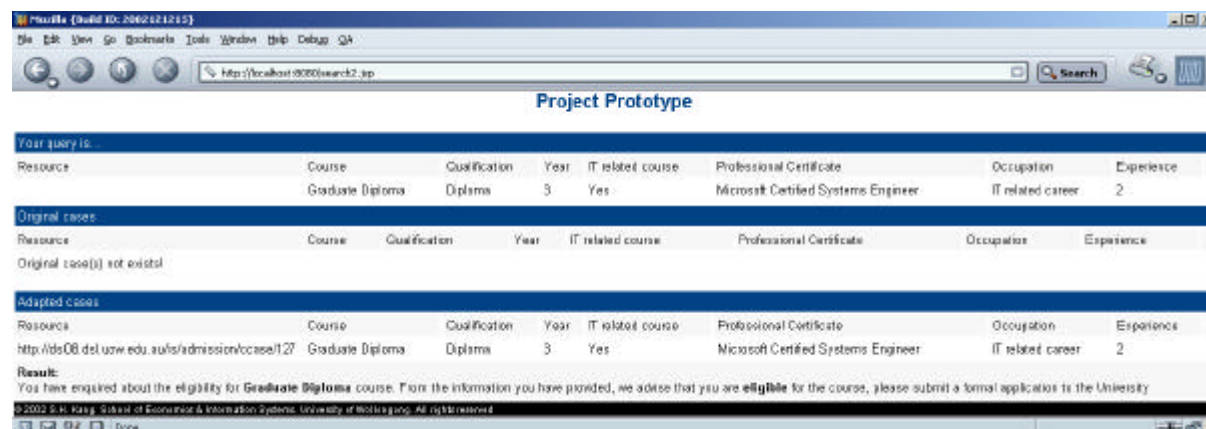


Figure 8. A Sample Output of Matched Adapted Cases

If none of the matched case is found, the system triggers the agent to process similarity assessment, and the case that is found to be nearest to the target case will be retrieved. This is determined by the lowest value calculated using Equation 1. In general, the lower the value the nearer the case is to its nearest neighbour. For demonstration purpose, we have presented the results for all cases found in the sample output in Figure 9. This output is not shown in the actual output screen.

Project Prototype

Your query is:

Resource	Course	Qualification	Year	IT related course	Professional Certificate	Occupation	Experience
	Master Information Systems	Bachelor	3	Yes	Microsoft Certified Systems Engineer	IT related career	5

Original cases

Resource	Course	Qualification	Year	IT related course	Professional Certificate	Occupation	Experience
Original case(s) not exist!							

Adapted cases

Resource	Course	Qualification	Year	IT related course	Professional Certificate	Occupation	Experience
Adapted case(s) not exist!							

Similar original cases

Resource	Course	Qualification	Year	IT related course	Professional Certificate	Occupation	Experience	*Similarity
http://ds0108.dsl.uow.edu.au/s/a/admission/case/1	Graduate Diploma	Diploma	3	Yes	Nil	IT related career	2	1
http://ds0108.dsl.uow.edu.au/s/a/admission/case/2	Graduate Diploma	Diploma	3	Yes	Nil	IT related career	3	1
http://ds0108.dsl.uow.edu.au/s/a/admission/case/3	Graduate Diploma	Diploma	3	Yes	Nil	IT related career	3	1
http://ds0108.dsl.uow.edu.au/s/a/admission/case/4	Graduate Diploma	Bachelor	3	Yes	Microsoft Certified Systems Engineer	IT related career	3	1
http://ds0108.dsl.uow.edu.au/s/a/admission/case/5	Graduate Diploma	Diploma	3	Yes	Sun Certified Programmer for Java 2 Platform	IT related career	0	0
http://ds0108.dsl.uow.edu.au/s/a/admission/case/6	Graduate Diploma	Bachelor	3	Yes	Nil	IT related career	1	0
http://ds0108.dsl.uow.edu.au/s/a/admission/case/7	Graduate Diploma	Diploma	3	Yes	Microsoft Certified Systems Engineer	Nil	0	-2
http://ds0108.dsl.uow.edu.au/s/a/admission/case/8	Graduate Diploma	Bachelor	3	Yes	Microsoft Certified Professional	Non-IT related career	2	0
http://ds0108.dsl.uow.edu.au/s/a/admission/case/9	Graduate Diploma	Diploma	3	Yes	Nil	IT related career	1	0
http://ds0108.dsl.uow.edu.au/s/a/admission/case/10	Graduate Diploma	Bachelor	3	Yes	Sun Certified Programmer for Java 2 Platform	Nil	3	-1
http://ds0108.dsl.uow.edu.au/s/a/admission/case/11	Master Information Systems	Diploma	3	Yes	Microsoft Certified Systems Engineer	IT related career	3	0
http://ds0108.dsl.uow.edu.au/s/a/admission/case/12	Master Information Systems	Diploma	3	Yes	Microsoft Certified Professional-Internat	IT related career	2	0
http://ds0108.dsl.uow.edu.au/s/a/admission/case/13	Master Information Systems	Diploma	3	Yes	Microsoft Certified Systems Engineer	IT related career	5	0
http://ds0108.dsl.uow.edu.au/s/a/admission/case/14	Master Information Systems	Bachelor	3	Yes	Sun Certified Programmer for Java 2 Platform	IT related career	3	0
http://ds0108.dsl.uow.edu.au/s/a/admission/case/15	Master Information Systems	Bachelor	3	Yes	Microsoft Certified Systems Engineer	IT related career	0	-1

Result: Whichever gives the smallest value (or closest to 0) in "Similarity" on the last column of the table is the nearest neighboring case to your inputted case. It means that if there is anything close to your case, you might be eligible to apply your desired course - **Master Information Systems**.

Figure 9. A Sample Output of Similarity Assessment

6. Conclusion and Discussion

We have demonstrated the feasibility of using agents in KMS by integrating CBR cycle to allow knowledge sharing and reuse to occur. We have also applied the ontology-driven mechanism to allow concept to be standardised in the KMS. The next phase of our research is the evaluation of the prototype. We will use the Goal-Question-Metrics (GQM) approach to evaluate the prototype (Basili 1992, Basili et al. 1994). The GQM is a measurement mechanism for feedback and evaluation. The measurement model has three levels: a defined object called Goal, a set of Question the way the assessment of a specific goal is going to be performed, and a set of data associated with every question called Metric in order to answer questions (Basili et al. 1994). The reason we use GQM is that because of its widespread acceptance of software measurement for feedback and evaluation and its mechanism allows us to evaluate the quality of the current processes and our prototype. The fundamental idea of our GQM is simply to set specific goals up to needs of our purpose. Then we refine the goals into questions that are tractable. Thirdly, we establish a set of metrics associated with every question in order to response it in a quantitative manner. Finally, gathering all feedback and evaluation is a key factor in order to keep improving ideas to action.

References

- Aamodt, A & Plaza, E (1994) 'Case-Based Reasoning: Foundational Issues, Methodological Variables, and System Approaches', *Artificial Intelligence Communication*, vol. 7, no. 1.
- Alavi, M. and Leidner, D (1999) 'Knowledge Management Systems: Issues, Challenges, and Benefits'. *Communications of the Association for Information Systems*, vol 1, no. 7
- Basili, V, (1992) 'The Goal Question Metric Paradigm', *Computer Science Technical Report Series*, CS-TR-2956 (UMIACS-92-96), University of Maryland & Fraunhofer Center, Maryland, <<http://www.cs.umd.edu/users/basili/>>
- Basili, V, Caldiera, G & Rombach, D (1994) 'The Goal Question Metric Paradigm', *Encyclopedia of Software Engineering*, vol 1, pp. 528-532, John Wiley & Sons, Inc.
- Berners-Lee, T, Hendler, J & Lassila, O (2001) 'The Semantic Web', May, *Scientific American*, <<http://www.sciam.com/2001/0501issue/0501berners-lee.html>>
- Brule, J & Blount, A (1989) *Knowledge Acquisition*, McGraw-Hill, New York
- Goldfarb, C (2001) *The XML Handbook*, 3rd Ed., Prentice Hall
- Gomez-Perez, A (1999) 'Applications of Ontologies and Problem-Solving Methods', *Workshop at Thirteenth Biennial European Conference on Artificial Intelligence*, the Brighton Centre, Brighton, UK, <<http://www.cogs.susx.ac.uk/ecai98/>>.
- Gruber, T (1993a) 'A Translation Approach to Portable Ontology Specifications', *Knowledge Systems Laboratory Technical Report KSL 92-71*, Knowledge Systems Laboratory, Computer Science Department, Stanford University, Stanford, California
- Gruber, T (1993b) 'Toward Principles for the Design for Ontologies used for Knowledge Sharing', In: Guarino, N & Polis, R (eds): *The International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic publishers, Padova, Italy
- Hayes-Roth, B (1995) 'An Architecture for Adaptive Intelligent Systems', *Artificial Intelligence: Special Issue on Agents and Interactivity*, vol. 72, pp.329-365
- Hendler, J (2001) 'Agents and the Semantic Web', *The Institute of Electrical and Electronics Engineers Intelligent Systems Journal*, Mar./Apr., <<http://www.cs.umd.edu/users/hendler/AgentWeb.html>>
- Hildreth, P. & Wright, P & Kimble, C (1999) 'Knowledge Management: Are we missing something' in proceedings of the 4th UKAIS Conference, York, UK, pp. 347-356, <http://www.cs.york.ac.uk/mis/>.
- Hornby, A (1995) *Oxford English Dictionary*, Oxford University Press, Oxford, New York
- HPL (2002) *The Jena Semantic Web Toolkit*, <http://www.hpl.hp.com/semweb/>
- Kolodner, J (1993) *Case-Based Reasoning*, Morgan Kaufman Publishers, Inc., San Mateo, Canada
- Konar, A (2000) *Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain*, CRC Press
- Lassila, O & Swick, R (eds) (1999) *Resource Description Framework (RDF) Model and Syntax Specification*, <<http://www.w3c.org/RDF/>>

- Maes, P (1995) 'Artificial Life Meets Entertainment: Life like Autonomous Agents', *Communications of the Association for Computing Machinery*, vol. 38, no. 11, pp.108-114
- Russell, S & Norving, P (1995) *Artificial Intelligence: A Modern Approach*, Prentice-Hall
- Smith, C, Cypher A & Spohrer, J (1994) 'KidSim: Programming Agents Without a Programming Language', *Communications of the Association for Computing Machinery*, vol. 37, no. 7, pp.55-67
- Sowa J (2000) 'Ontology, Metadata, and Semiotics', *Lecture Notes in AI #1867*, Springer-Verlag, Berlin, pp.55-81
- Turban, E & Aronson, E (2000) *Decision Support Systems and Intelligent Systems*, 6th edn, Prentice Hall, upper Saddle River, NJ
- W3C. (2000) *Extensible Markup Language (XML) 1.0 (Second Edition)*, <
<http://www.w3.org/TR/REC-xml>>
- W3C. (2002a) *Requirements for a Web Ontology Language*, URL:
<http://www.w3.org/TR/2002/WD-webont-req-20020307/>
- W3C. (2002b). *Semantic Web*, URL: <http://www.w3.org/2001/sw/>
- Wache, H, Vögele, T, Visser, U & Stuckenschmidt, H (2001) 'Ontology-Based Integration of Information – A Survey of Existing Approaches' in Proceedings of Seventeenth International Joint Conference on Artificial Intelligence: Workshop on Ontologies and Information Sharing, Seattle, Washington, USA
- Walsham, G (2001) 'Knowledge Management: The Benefits and Limitations of Computer Systems', *European Management Journal*, vol. 19, no. 6, pp. 559-608
- Watson, I (1997) *Applying Case-Based Reasoning: Techniques for Enterprise Systems*, Morgan Kaufmann Publishers, Inc., San Francisco
- White, E (2000) 'Tacit and explicit knowledge: Conceptual confusion around the commodification of knowledge' in Knowledge Management: Concepts and Controversies, BPRC, Warwick University, pp. 62-64
- Wooldridge, M & Jennings N (1995) *Agent Theories, Architectures, and Languages: a Survey*, In: Wooldridge, M & Jennings, RN (eds), *Intelligent Agents*, Springer-Verlag, Berlin, pp.1-22

Appendix

Original Case Number	Requirements for Admission to Information Systems						
	Course	Qualification	Year	IT related	Professional Certificate	Occupation	Work Experience
1	Graduate Diploma	Diploma	3	Yes	Nil	IT related career	2
2	Graduate Diploma	Diploma	3	Yes	Nil	IT related career	3
3	Graduate Diploma	Diploma	3	Yes	Nil	IT related career	3
4	Graduate Diploma	Bachelor	3	Yes	Microsoft Certified Systems Engineer	IT related Career	3
5	Graduate Diploma	Diploma	3	Yes	Sun Certified Programmer for Java 2 Platform	IT related career	0
6	Graduate Diploma	Bachelor	3	Yes	Nil	IT related career	1
7	Graduate Diploma	Diploma	3	Yes	Microsoft Certified Systems Engineer	Nil	0
8	Graduate Diploma	Bachelor	3	Yes	Microsoft Certified Professional	Non-IT related career	2
9	Graduate Diploma	Diploma	3	Yes	Nil	IT related career	1
10	Graduate Diploma	Bachelor	3	Yes	Sun Certified Developer for Java 2 Platform	IT related career	3
11	Master Information Systems	Diploma	3	Yes	Microsoft Certified Systems Engineer	IT related career	3
12	Master Information Systems	Diploma	3	Yes	Microsoft Certified Professional+Internet	IT related career	2
13	Master Information Systems	Diploma	3	Yes	Microsoft Certified Systems Engineer	IT related career	5
14	Master Information Systems	Bachelor	3	Yes	Sun Certified Programmer for Java 2 Platform	IT related career	3
15	Master Information Systems	Bachelor	3	Yes	Microsoft Certified Systems Engineer	IT related career	0

Appendix A. Original cases used in prototype