Intelligent knowledge acquisition using case-based reasoning: knowledge sharing and reuse

Seung H. Kang
University of Wollongong

Intelligent Knowledge Acquisition using Case-based Reasoning: Knowledge Sharing and Reuse

A thesis submitted in partial fulfilment of the requirements for the award of the degree

Master of Information Systems (Research)

From

UNIVERSITY OF WOLLONGONG

By

Seung Hwan Kang

Master of Information Systems, University of Wollongong
Bachelor of Commerce (Business Information Systems), University of Wollongong

Information Systems
School of Economics and Information Systems

2003

Thesis Certification

CERTIFICATION

I, Seung Hwan Kang, declare that this thesis, submitted in partial fulfilment of the requirements for the award of the Degree of Master of Information Systems (Research) at the University of Wollongong, is wholly my own work otherwise I have given fully documented references or acknowledgement to the work of others. The document has not been submitted for qualifications at any other academic institution.

Seung Hwan Kang

24 August 2003

Acknowledgement

Without the help of many people, I might never have finished this thesis. I am particularly enormously indebted to Dr. Sim Kim Lau who read humble thesis of mine, and gave me many valuable lessons. Our group, Decision Systems Laboratory (DSL), and all members of DSL have supported me over my work from the beginning of my study to the end with suggestions and feedbacks. In particular, I am greatly indebted to Professor Aditya Ghose who put me in the situation I have never thought about it. It always turns into valuable lessons. Our department, Information Systems, has supported me financially over years to allow me to attend the Conferences. Particularly, I am indebted to Associate Professor Helen Hasan who sorted out many problems when I face on during my course. I would like to make special thanks to a domain expert, Mrs Lily Soh who provides me expertises in the domain of the prototype. Finally, I am grateful to many colleagues of mine, and especially my family who encourage me and gave me an opportunity to continue my study.

August 2003

Seung Hwan Kang

List of Publications

This is a list of referred conference papers that is related to this research work.

Kang, S. & Lau, S. 2003 “A web-based student enquiry system: the application of ontology and cased-based reasoning”, in *the 4th International We-B Conference*, 24 – 25 Nov. 2003, Hotel Rendezvous, Scarborough, Perth, Australia

Kang, S. & Lau, S. 2003 "A Framework for Case-based Reasoning Integration on Knowledge Management Systems", in *the Proceedings of The 7th Pacific Asia Conference on Information Systems 2003 (PACIS2003)*, 10 -13 Jul., South Australia at Hilton International Hotel, Australia. pp.1327-1343

Kang, S. & Lau, S. 2003 "An Ontological Approach in Knowledge Management Systems: A Case Study", in *the Proceedings of The 14th International Conference of the Decision Sciences Institute (DSI2003)* (CD-ROM), 04-8 Jul., Shanghai, China.

Kang, S. & Lau, S. 2002 "Intelligent Knowledge Acquisition using Case-Based Reasoning", in *the Proceedings of The 13th Australasian Conference on Information Systems (ACIS2002)*, 4-6 Dec., Victory University, Melbourne, Australia. pp.403-410

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
ANSI	American National Standards Institute
AOP	Agent-Oriented Programming
CBR	Case-based Reasoning
DAML	DARPA Agent Markup Language
DARPA	US Defence Advanced Research Projects Agency
DBMS	Database Management Systems
DOM	Document Object Model
DSL	Decision Systems Laboratory
GML	Generalised Markup Language
HPL	Hewlett-Packard Labs
HTML	Hypertext Markup Language
IS	Information Systems
IT	Information Technology
KMS	Knowledge Management Systems
MCP+I	Microsoft Certified Professional + Internet
MCSE	Microsoft Certified System Engineer
MCSE+I	Microsoft Certified System Engineer + Internet
OIL	Ontology Inference Layer
OOP	Object-Oriented Programming
OWL	Ontology Web Language
RDF	Resource Description Framework
RDQL	RDF data Query Language

SAX	Simple API For XML Parsing
SCJP	Sun Certified Programmer for Java 2 Platform
SGML	Standard Generalised Markup Language
SHOE	Simple HTML Ontology Extensions
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

Table of Contentsi

Abstract

This research aims to investigate the feasibility of using techniques in case-based reasoning for the knowledge acquisition phase in the development of knowledge management systems to allow knowledge sharing and reuse. Case-based reasoning is one area of artificial intelligence that allows us to deal with situation that are similar to the current ones and use them to solve new problems. The application of case-based reasoning techniques allows the knowledge management systems to acquire new knowledge by adopting the knowledge gained in new cases and reusing the old ones in the case repository. This way, it allows old knowledge to be shared and new knowledge to be added in the knowledge repository. This research also proposes an ontology-driven mechanism that provides standardised vocabularies and conceptualisation of knowledge domain. A prototype for a student admission enquiry system is developed on the Semantic Web to demonstrate the application of case-based reasoning cycle so that knowledge sharing and reuse can be achieved autonomously. Our results show that the application of case-based reasoning techniques allows sharing and reuse of past experience and knowledge in the knowledge management systems.

Chapter 1 Background and Introduction

This research investigates the feasibility of using techniques in Case-based Reasoning (CBR) and ontology to allow knowledge sharing and reuse to be achieved in Knowledge Management Systems (KMS).

This chapter presents an overview and general introduction to the thesis. This chapter is organised as follow. Section 1 presents a general introduction to the thesis. Section 2 identifies the research problem. Section 3 presents an overview of the research and Section 4 states the research aim. Research objectives are presented in Section 5. Section 6 gives an overview of the proposed approach. Section 7 presents the organisation of this thesis.

1.1 Introduction

The purpose of building KMS can be seen as sharing corporate knowledge in the organisation to provide support for knowledge-sharing within communities-of-practice (Walsham 2001). However, due to downsizing and outsourcing functions of organisation in the recent years, organisations have began to lose knowledge as people leave and take the knowledge and skills with them (Hildreth et al. 1999). Moreover it is difficult to keep distributed knowledge up-to-date and to share the knowledge in the rapid changing environment. Globalization of organisations also means there is a need for knowledge to be shared in different locations. In general, successful KMS gives leverage for sharing the knowledge within an organisation.

The development of KMS involves the following steps: *create, capture, refine, store, manage* and *disseminate knowledge* (Turban and Aronson 2001 pp.362-363). One of the major problems identified in developing KMS is the knowledge acquisition phase, in particular *the create and capture knowledge* steps identified above. This research aims to investigate the feasibility of using techniques in CBR and ontology to improve knowledge acquisition process so that knowledge sharing and reuse can be achieved in KMS.

1.2 Research Problems

Knowledge sharing within an organisation is often focused on skills and facts that can be written down and taught to others. This type of knowledge is often classified as explicit knowledge. On the other hand, tacit knowledge is knowledge and skill that is hard to describe such as experience or native talent in people. In general, it is difficult to code tacit knowledge.

Knowledge acquisition is generally considered to be one of the most important steps in KMS development. Knowledge acquisition method is designed to construct explicit knowledge from implicit or tacit knowledge. This is often done using manual methods such as interviewing, tracking the reasoning process, and observing documented and undocumented knowledge. The aim of this process is to find what information or knowledge is being used and how it is being used.

One of the problems identified in the development of KMS process is the difficulty in knowledge acquisition phase. Current manual methods described above can be slow, costly and erroneous. In addition, it is difficult to expressively describe and represent knowledge in a particular domain of interest in the traditional development manner. Another problem that has been identified is the difficulty in sharing and reuse knowledge in KMS. Inefficient knowledge representation also results in the knowledge stored in KMS difficult to change and update, in particular if one wishes to extend the knowledge gained through new processes or new learning. Thus this research aims to investigate ways to allow knowledge to be shared and reuse in KMS.

1.3 Overview of Research

KMS is considered to be weak in terms of facilitating the concept of knowledge sharing and reuse (Dubitzky et al. 1999). To ensure knowledge sharing and knowledge reuse can be achieved in KMS, especially in a networked environment such as the World Wide Web (WWW), this research proposes to use CBR techniques to provide an opportunity to allow new knowledge to be updated, stored and retrieved in the KMS. The new knowledge can be generated based on past knowledge and experiences. CBR is used to resolve new problems through the process of retrieve, reuse and refine processes of past solutions. The *case* in CBR is a conceptualized piece of knowledge representing past experience (Kolodner 1993). Case representation includes a detailed problem description and a detailed solution. In general, when new problem arises, the retrieval process identifies the case with the most similar problem description from the past cases and applies the solution or adapts the solution to the new problem. To achieve this and to support rich

knowledge representation in the distributed networked environment such as the WWW, there is a need to present a well-defined set of domain of interest in community of practice.

This research proposes the use of ontology as a feasible approach to conceptualise a set of terms in the community of practice. Ontology is a study of existence and is described as "a science or study of being" (Hornby 1995). Ontology has been widely used in Artificial Intelligence, particularly in knowledge representation. It aims to capture domain knowledge in a generic way by providing a commonly agreed understanding of a domain (Gomez-Perez 1999). In this research, ontology is used to capture common interest of knowledge in the domain. It is applied to explicitly formalise the specification of a shared conceptualisation. It provides new opportunities to prevent ambiguities in knowledge representation by supporting well-agreed terms or vocabularies. We aim to apply ontology as a form of meta-knowledge to allow consistent conceptualisation to be referenced in the KMS.

To support a standardised platform for ontology, the Semantic Web features can be applied. Conventional knowledge representation in KMS does not support machine understandable format in general. The Semantic Web is an extension of the current web (Berners-Lee et al. 2001b). It is to give well-defined meaning of information, and provide an enabling environment for computers and people to work in cooperation. Its mechanism is to use logic and semantics to structure distributed knowledge on the web (Berners-Lee et al. 2001b). It allows software agents to look at the meaning of the contents and refers to relevant knowledge. The application of ontology and Semantic Web in a networked environment such as the WWW provides

an opportunity of creating a platform to support a machine understandable mechanism. This means human-centred view of knowledge can be changed to a machine-centred one, and it allows machine such as software agents to be developed and used with the aim to create machine-readable and machine-understandable forms of knowledge. Advance of this standard platform means that KMS now allows an agent to interoperate meaning of contents, and communicate with other agents so that machine understandable case representation can be achieved and supplied to agents. This way, agents can be developed to carry the assigned task autonomously.

In summary, this research focuses on using CBR techniques to improve the knowledge acquisition process so that knowledge sharing and reuse can be achieved. The concept of knowledge sharing and reuse is important to allow up-to-date new knowledge to be added or extended in KMS. The research will also investigate the application of ontology so that a well-defined set of domain of interest in community can be presented in a way that it provides commonly agreed understandable terms to be shared and reused easily. The research will investigate the application of software agents in the Semantic Web so that a machine-understandable knowledge representation can be implemented to capture and update new knowledge in the KMS.

1.4 Research Aim

This research aims to investigate the feasibility of using techniques in CBR, ontology, and the Semantic Web features to improve knowledge acquisition process so that knowledge sharing and reuse can be achieved in KMS.

1.5 Research Objective

The objectives of this research are as follows:

- To investigate the feasibility of using techniques in CBR to allow knowledge sharing and reuse to be achieved.
- To investigate the feasibility of using ontology to provide a mean to standardise conceptualisation of knowledge and to facilitate a well-defined domain knowledge in community of practice.
- To investigate the feasibility of using the Semantic Web as a platform to allow software agents to be implemented to facilitate the process of knowledge sharing and reuse through the process of representing knowledge in machine-readable and machine-understandable form

1.6 A Brief Overview of Research Approach

A framework that integrates CBR techniques to the KMS development cycle is proposed in this research. This aim is to improve knowledge sharing and reuse. The main feature of this approach is to apply the four phases of CBR cycle in KMS. Briefly, the approach allows knowledge that is stored to be retrieved from the knowledge repository. Then the knowledge is reused, refined and retained to allow the knowledge adaptation process to occur in KMS. By refining and reviewing the knowledge in the refine phase of CBR, new knowledge can be kept updated and retained and finally stored in the knowledge repository.

A prototype will be developed based on the domain of student admission criteria to an Information System course. The domain knowledge is provided by the domain expert who is the admission officer of the course. Knowledge is stored as cases in the form of successful application. Each case consists of applicant's details such as their academic qualification, working experience and relevant professional certificates. The proposed framework is developed in a distributed networked environment of the WWW. Ontology is used to provide standardised structure, set of vocabularies, and conceptualisation of knowledge domain. The aim is to minimise inconsistency and ambiguity in knowledge representation. The prototype is developed as a web based system. An evaluation of the prototype is carried out.

1.7 Organisation of Thesis

The rest of the thesis is organised as follows. Chapter 2 presents relevant literature review related to this research. The literature review will discuss literature in KMS, CBR, Ontology and the Semantic Web. Chapter 3 presents a framework that can integrate CBR techniques to the KMS development cycle. Chapter 4 discusses issues related to development of a prototype. It includes the discussion of knowledge domain, knowledge representation, ontology design, and the application of software and agents. Chapter 5 presents evaluation of the prototype. Chapter 6 concludes the thesis.

Chapter 2 Literature Review

This chapter presents theoretical background related to this research. Literature relating to KMS, CBR, ontology and the Semantic Web will be presented.

The chapter is organised as follows. Section 1 discusses issues related to KMS, and development of KMS. Section 2 discusses CBR, and the four phases of CBR cycles. Section 3 discusses issues related to ontology and the Semantic Web. Conclusion is followed in Section 4.

2.1 Knowledge Management Systems

We begin this section by first defining terms that include data, information and knowledge. Data are the raw inputs of individual facts, statistics, or items of information (Benson and Standing 2002). Information is processed or value-added data (Vance 1997) and knowledge is the understanding of what information means or implies (Benson and Standing 2002). Knowledge can be further classified as tacit and explicit knowledge (Whitley 2002). Tacit knowledge is knowledge that cannot be easily described such as skills, experience or native talent. Explicit knowledge is skills and facts that can be written down and taught to others such as technical documents (Polanyi 1967).

Since mid-1970s, knowledge started to play an important role in organisational strategy. By the 1980s, the importance of organisational knowledge is increasingly

recognised. Organisations have focused on processes and strategies to manage innovation and to build knowledge (Leonard-Barton 1995). As a result, a system is developed to provide a technological base for managing knowledge. The term knowledge management can be seen as management of knowledge related activities. These activities include broad, multi-dimensional and covers most aspect of the enterprise's activities (Wiig 1997). For example, MediaOne knowledge management project is initiated in 1998 to support shared corporate resources and individual experiences across the enterprise (Roberson 2002).

There are different approaches in which knowledge management is used in the organisations. The first approach is that of the repository model. It focuses on managing information and reusing knowledge in tangible formats. Malhotra (1998) states that "Knowledge Management caters to the critical issues of organizational adap[ta]tion, survival and competence in face of increasingly discontinuous environmental change. Essentially, it embodies organizational processes that seek synergistic combination of data and information processing capacity of information technologies, and the creative and innovative capacity of human beings". Knowledge management is also seen to be "making a direct connection between an organisation's intellectual assets" (Barclay and Murray 1997). This approach is viewed as a legal approach. It involves intellectual capital, copyright, patents and trademarks. Knowledge management can also be regarded as business intelligence. It is a process to produce valuable up-to-date information for operative and strategic decision-making (Hannula and Pirttimaki 2003). With the help of business intelligence, organisations achieve competitive advantages in a rapidly changing business environment by utilising intelligence to the relevant knowledge gained. Other

approaches include the cognitive and continuous learning approach which is the ability to learn. It involves an individual ability's to acquire continuous and ongoing renewal of organisational information and reuse it for problem solving and decision-making. In addition, cognitive approach of knowledge management focuses on learning within groups as well as individual's learning level.

Thus, it can be seen that a lot of organisations have begun to recognise knowledge as the most valuable assets in their organisations. These valuable assets include personal skills and experience as well as any stored information in the organisations. In general, KMS refers to a system designed specifically to provide the sharing and integration of knowledge (Alavi and Leidner 1999). It allows corporate knowledge to be shared in the organisations effectively and efficiently (Walsham 2001). According to Thierauf (1999), KMS is a system that can provide competitive advantage by giving decision makers the necessary insight into patterns and trends affect their domain. A KMS is able to make comparisons, trends analysis, and historical and current knowledge presentations. But the most important thing is KMS enables decision makers to analyze and understand the patterns quickly and identify the most significant trends. From this perspective, KMS provide essential knowledge and related information to decision makers in making better decisions.

To increase the effectiveness of the knowledge development process, it has been suggested that "organising strategies should be defined and initiated based on knowledge development phases" (Ganesh 2000). There are four phases of knowledge development cycle, which includes knowledge creation, knowledge adaptation, knowledge distribution and knowledge review (Ganesh 2000). The knowledge

creation phase involves proving, learning and evaluations of common means of managing the knowledge, and the knowledge adoption step involves the use and re-use of existing knowledge. Then the knowledge distribution phase provides ease of access, sharing and manipulation of knowledge via knowledge infrastructure, media selection and knowledge-fundamental. Finally the knowledge review and revision steps involve testing processes of validity and reliability.

A similar process is found in the KMS development life cycle. A typical KMS development cycle consists of: "create knowledge, capture knowledge, refine knowledge, store knowledge, manage knowledge and disseminate knowledge" (Turban and Aronson 2001 pp.360-364) (See Figure 2.1). *Create Knowledge* is the process of facilitating the solution of a problem in different contexts (Bhatt 2000) or viewing reality in new perspectives (Weick 1995). *Capture Knowledge* is the process of driving and adding value to an organisation from generated data, information, and so on (Turban and Aronson 2001). *Refine Knowledge* is the process of placing knowledge in context or specifying tacit to explicit knowledge (Turban and Aronson 2001). *Store Knowledge* is the process of accumulating knowledge in a repository. *Manage Knowledge* is the process of reviewing and revising knowledge so that it is maintained as current, relevant and accurate. *Disseminate Knowledge* is the process of making knowledge available, distribute and share throughout the organisation.

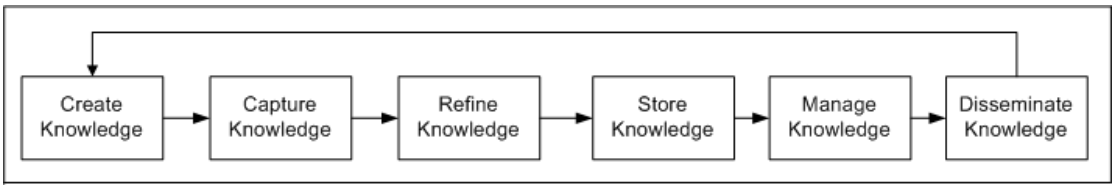

Figure 2.1 KMS development cycle (Source: Turban and Aronson 2001, p. 363)

In particular, the knowledge creation phase is the most important phase and is often referred as knowledge acquisition. Knowledge acquisition is a process of translating implicit knowledge into explicit form (Brule and Blount 1989). According to Partridge and Hussain (1995 p.7), "knowledge acquisition is a very labour-intensive activity. It is almost an art-form with questions arising for which there are no algorithms or computer programs". Extensive researches have been conducted on ways to improve the knowledge acquisition process. In particular, attempts have been made to either partially or fully automate the knowledge acquisition process using Artificial Intelligence (AI) techniques (Liebowitz 2001, Tshi et al. 2000). Partial or semi-automatic approach allows the interaction between knowledge experts and knowledge engineers to be reduced. On the other hand, the automatic approach refers to using AI technique that allows the experts to build their own knowledge bases with minimal or even without the assistance from the knowledge engineers (Turban and Aronson 2001). In another example, researches have used AI techniques such as neural networks to develop algorithms that acquire knowledge autonomously from data.

The widespread use of the WWW and Internet technologies have also helped to speed up the process of knowledge acquisition. The interface of WWW is incorporated to

the system to gather efficient and complete knowledge from the application perspective. Knowledge engineers often have to elicit knowledge from experts using interview method, which is often time-consuming. To speed up the traditional manual method of interviewing, a knowledge engineer can interview experts via electronic interviewing. Documented knowledge can be submitted via electronic forms and these forms can be retrieved and stored in the knowledge base or knowledge repository. In practice, video-conferencing and web meeting technology can be additionally used to support networks of people in the knowledge communities to share tacit knowledge. Other manual knowledge acquisition methods include tracking and observation. The tracking approach is used to find what information is being used and how it is being used. However, these manual knowledge acquisition methods are slow and prone to error.

According to Partridge and Hussain (1995), the knowledge acquisition process starts with planning knowledge base or knowledge repository in KMS. It organises knowledge for the knowledge base, followed by knowledge elicitation/extraction from the different relevant sources of knowledge. Then, it formulates and represents knowledge for inference making. For example, a decision table and production rules are used to express logical relationships, and to identify set of conditions and actions. After encoding knowledge in machine-readable form, implementation of knowledge-base is followed. When the knowledge base is ready for testing, knowledge engineer and the domain expert will verify and validate it to ensure the systems have met the requirements. Finally, it is ready for systems test. Figure 2.2 shows the relevant steps of knowledge acquisition in the development of knowledge processes.

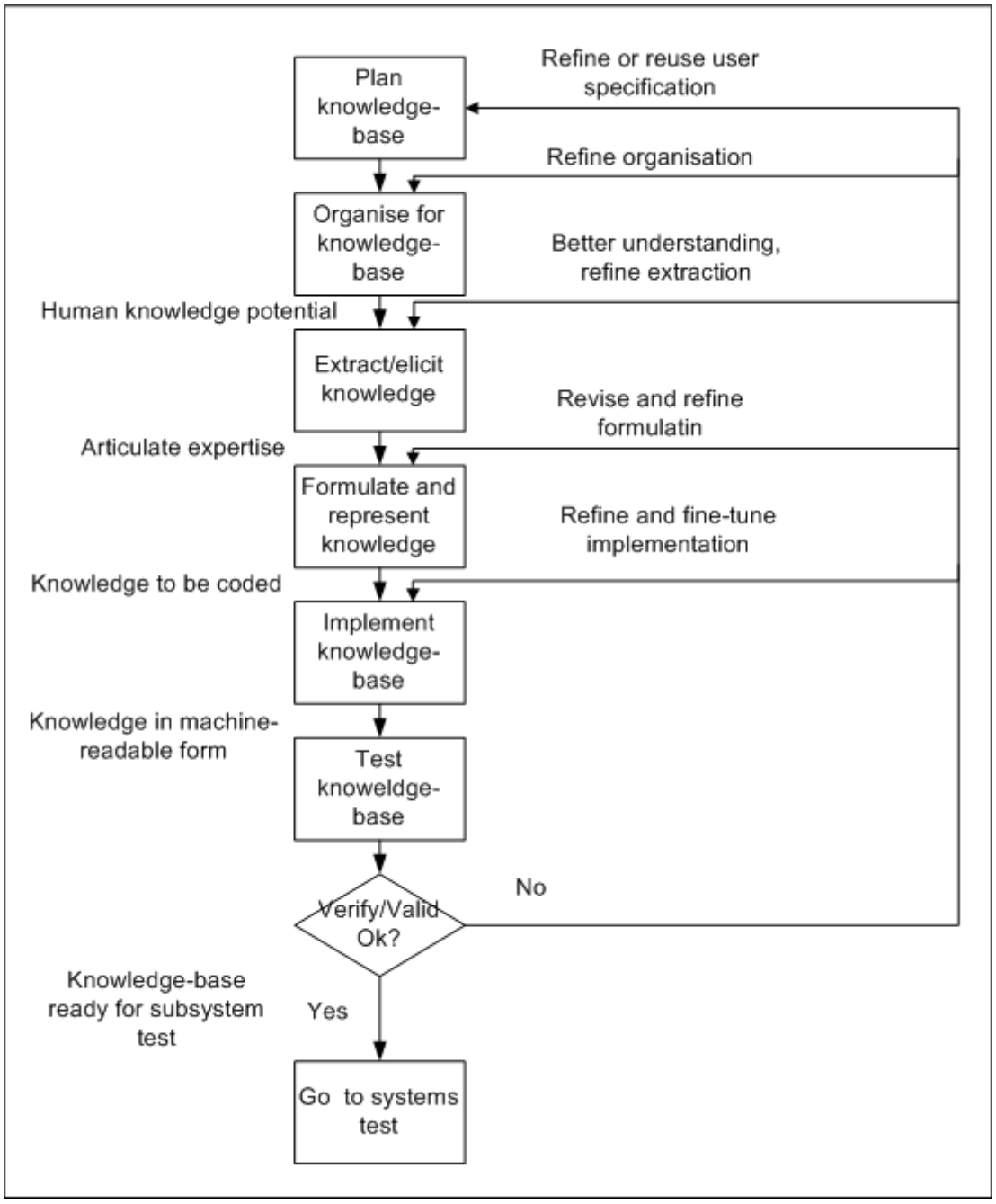

Figure 2.2 Overview of knowledge development processes (Source: Partridge and Hussain p.184)

As discussed, the purpose of building a KMS is to share corporate knowledge in the organisations. Walsham (2001 p.603) points out that "support for knowledge-sharing within communities-of-practice is a valuable focus for contemporary organisations". One of the ways of developing successful KMS is through facilitating the concepts of knowledge sharing and use in practice. Based on Churchman's conceptualisation of

knowledge, *sharing* refers to "communicating both tacit and explicit knowledge" and *use* refers to "making use of it" (Churchman 1972). However, the traditional development of KMS is considered to be weak in facilitating the concepts of knowledge sharing and reuse in practice. Making reuse of previous useful solutions to solve new problems by referencing to old solutions is often not easy to achieve. In addition, gathering well-agreed terms of reference in the communities of practice, it may not be easy to use knowledge in organisation.

2.2. Case-based Reasoning

This section discusses theoretical background on CBR. The research on CBR can be traced back to the work of Roger Schank's (1982) dynamic memory and Memory Organisation Packet (MOP). According to Schank (1982 p.2), "a dynamic memory is one that can change its own organization when new experiences demand it. A dynamic memory can learn." It is the way we deal with a new problem by observing new information to generalise new solutions from past experiences. By understanding a new problem, we can dynamically solve our new problem to reflect our experiences. A MOP is "information about how memory structures are ordinarily linked in frequently occurring combinations" (Schank 1982 p.83). It is an important approach in which past experience is structured. It values past experience which is not often integrated in the computing systems. These past experiences are used to interpret new inputs using the most closely related past cases.

Using cases in CBR, it is possible to provide better knowledge sharing and reuse solutions because CBR cycle involves revision and refinement phases. (*The CBR*

cycle will be discussed later) In general, there are two parts to a case (Kolodner 1993 p.18): the first part is the lesson(s) it teaches, and the second part is the context in which it can teach its lesson(s). According to Kolodner (1993 p.3), "a case is a conceptualised piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoned." Therefore, a case or problem situation can be defined as a conceptualised part of knowledge representing past experience.

In general, CBR refers to a problem-solving paradigm that relies on case representation, instead of only relying on general knowledge of a problem domain. 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. For example, stored data in a relational database, photographs, sound, and video can be represented in a case. However it may be difficult to represent large amount of inter-related data in a case. Therefore the functionality and acquisition of information need to be clarified first before deciding what should be represented in cases. Watson (1997 p.19) points out what information should be in a case using two pragmatic measures: the functionality of the information and the ease of acquisition of the information. In fact, CBR is dependent on the structure and collected case in case repository, so it is important to have a mechanism that organises information that can be retrieved when it is required. Case representation also should have a standardised mechanism that is supportable, suitable and appropriate to support case retrieval.

There are four phases in the CBR cycle: retrieve, reuse, review and retain as shown in Figure 2.3 (Aamodt and Plaza 1994, Kolodner 1993, Watson 1997 p.17).

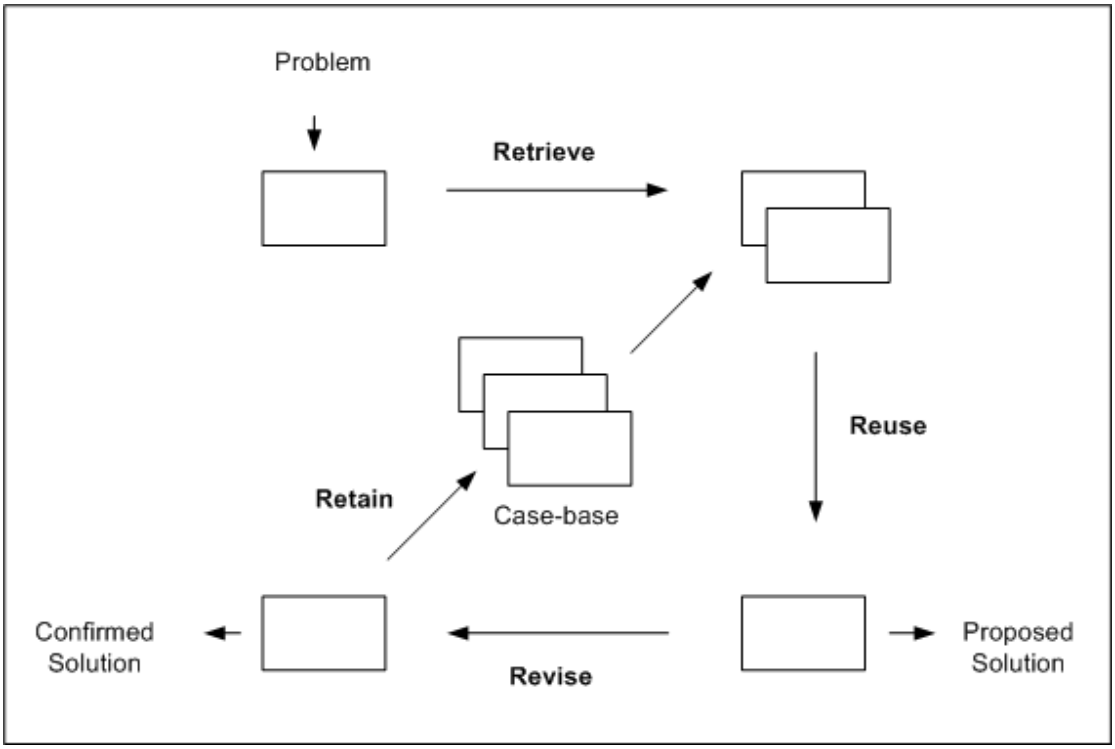

Figure 2.3 The CBR cycle (Source Watson 19997 p.17)

The retrieval phase is to decide which case in the case repository is similar to a target case (target case 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-neighbor retrieval and inductive retrieval.

Nearest-neighbor retrieval is a technique used to measure how similar the target case is to a source case (Watson 1997). It processes retrieval of cases by using the comparison approach of a collection of weighted attributes in the target case to source

cases in the CBR library. If there is no matched case in the CBR library, CBR system will return the nearest-matched source case. The return of the nearest-matched case can be found using the following equation (Watson 1997 p.28):

$$\text{Similarity } (T, S) = \sum_{i=1}^{n} f(Tj, Si) \times Wi \qquad \text{(Equation 1)}$$

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*

The equation of the nearest neighbor retrieval technique represents the sum of similarity of the target case to the source case for all attributes multiplied by the importance weighting of individual attributes. The CBR system retrieves a meaningful case that may provide a detailed solved problem description to a new problem. However, the nearest neighbor retrieval technique is not efficient. This is because whenever new cases are introduced, indexing needs to be performed and this could affect efficiency.

Inductive retrieval is a technique used to extract rules or construct decision trees from past cases (Watson 1997). This technique processes a target case based on indexed source cases. The source cases are normally indexed by keywords and stored into a set of cases. 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. Aamodt and Plaza (1994) and Watson (1997) suggest 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.

In the reuse phase, the solution from the retrieved case is used to solve the target case. In general, 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 produce a new solution to a new problem. The solution of 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.

Structural adaptation is a technique to apply adaptation rules or formulas directly to the stored solution in the CBR library. Once a case has been applied by the adaptation rules or formulas, the CBR system adapts the case 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 (Watson 1997). The retrieved solution then must be stored as an additional case in the CBR library so that it reproduces a new solution to the new case.

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. However, it is important that indexing be provided at an appropriate level of generality in terms of global and local context, so that it reflects the hierarchical structure of cases (Watson 1997).

In summary, when the new problem issue arises, the retrieval process identifies the problem as a case to find out the most similar one in the past cases. Then, if there is any matched one in the past cases, it will be presented as a solution of new case. If it is necessary, adaptation occurs and a new case is created.

There are a few considerations when building CBR systems. One of the consideration is the quality of CBR reasoning (Kolodner 1993 pp.6-8). Generally, the more experienced the CBR system, then it is more likely to perform better than the one that is less experienced. In other words, the ability to understand a new problem in terms of old experience has important role in the CBR systems. This is because more experience CBR system has more successful ability to propose solutions that are stored in the case repositories to solve new problems. On the other hand, less experience CBR system may not be able to propose solutions to new problems during the reasoning processes. In addition, the adaptation ability in the CBR systems is important, especially when it attempts to use an old solution to fit in to the new situation. The ability to integrate new experiences into its case memory appropriately is important so that solved cases are achieved for later reasoning purpose. Evaluation

or repair needs to be achieved to provide feedback to other similar cases. The CBR system must learn from its experiences in such a way that it integrates mistakes and it is able to tell us what was wrong or right when new problems repeated the next time.

Various CBR systems have been built over the last twenty-five years. As discussed, CBR starts with Roger Schank's dynamic memory and MOPs. The Computerized Yale Retrieval and Update System (CYRUS) is the first CBR system that is built based on Schank's dynamic memory model and MOP theory (Kolodner 1983). The system allows users to ask questions about the travels and meetings of former US Secretary of State Cyrus Vance (Watson 1997).

PROTOS is a case-based classification system and uses knowledge acquisition approach to handle a problem of classifying auditory diseases (Kolodner 1993). Domain knowledge of hearing disorders is represented as cases. PROTOS uses a trial-and-error method of problem solving. It is known as a heuristic approach that learns, discovers and solves problems on its own by trial and error (Partridge and Hussain 1995). When PROTOS faces a problem, it tries to find a solution. If a problem is successfully solved in the first trial, PROTOS does not learn domain knowledge of hearing disorders that is represented in the new case. However, if PROTOS solves the problem in the second or third trials, PROTOS learns the cause of the particular case during classification. If a user requires further modification and explanation, it can be stored in the new case for future usage.

Another example is CASEY which uses case-based and model-driven approach to complement the CBR process. It is a case-based and model-based reasoning approach

because "when a problem turns out to be unsolvable by retrieving a past case, a general domain knowledge model is used in a second attempt to solve the problem" (Tecuci, 1995 p.219). CASEY is used to analyse patient' symptoms. For example, where two patients show differences in symptoms, then it has to be reconciled by CASEY's evidence rules. The rules are applied and used as part of the model. The evidence rules examine previous diagnosis until it matches. Especially, rules go on to create an explanation of one patient's symptoms by adapting the other patient's diagnosis.

HYPO is another example of CBR application which is used to produce and assess arguments for both the defendant's and the plaintiff's side in the domain of law. It is case-based argumentation where it compares and contrasts procedures in reasoning. If several different cases are available to make legal argument, it provides an idea of which case is preferable in terms of taking the strongest arguments. According to Kolodner (1993 p.50) the contribution made by HYPO is that it shows some of the steps involved in the cognitive processes and the knowledge in engaging such reasoning.

In literature, CBR has been applied successfully in various KMS to reuse previous solutions to resolve new problems. The application of CBR techniques allows KMS to acquire new knowledge, by adopting knowledge gained in the new cases and reusing the old ones. This allows new knowledge to be shared and added in the knowledge repository. In this section, we provide some examples in which CBR has been successfully applied to KMS. The first example is British Airways. British Airways needs a diagnostic tool to support maintenance technicians in solving

problems in Concorde Olympus powerplant. Due to the complex assembly procedures and the need of cost effective operation, British Airways requires a repository of diagnostic experiences that can be made available to all technical engineers (Magaldi 1999). A software package called SportLight™ from CaseBank Technologies Inc™ is used to develop the KMS. It is the CBR software to support the troubleshooting of complex equipment, systems or processes (Casebank Technologies 2002). One of the intangible values identified is the knowledgeable assets gained by the organisation.

The second example is the The World Bank. The World Bank needs a knowledge-centred mechanism to utilise economic and social development projects. It aims to provide cooperated knowledge repository to lead a better search and browsing mechanism in order to make decisions and judgement that exists in the collection of related experiences (Moussavi 1999). Problems, relevant useful information and solutions are stored in case library. Then it is indexed between similar cases. A situation assessment user interface is developed to help the end user to search the cases. In this example, CBR improves the quality of operational values in the World Bank.

The last example is The Great Lakes Geriatric Interdisciplinary Team Training (GITT) project which is a collaborative research to support the long term care of Alzheimer's disease patients (GITT 2002). In this example, CBR is not only used to solve problems for the patients' symptoms, but it is applied to support GITT participants in decision making when conflicting perspectives arise.

2.3 Ontology and the Semantic Web

The widespread use of the WWW and the Internet can help to improve the knowledge acquisition and knowledge distribution processes of the knowledge life cycle. This section discusses issues for successful and effective deployment of knowledge in the WWW environment. Issues related to ontology and Semantic Web will be discussed, particularly in terms of enabling information and knowledge reuse and sharing over the web.

Since 1990's, the WWW has evolved in many ways. Even though the WWW provides an ideal approach to the development of rich knowledge base by enlarging domain of human knowledge via the WWW channel, the need for effective retrieval methods are essential. The following discusses some of the commonly used search engines available in the WWW. A search engine use software robots known as "Web Spiders" to find information for its database by roaming the Internet. The web search engine can be used to integrate distributed information in centralised storages. For example, huge amounts of Web links can be collected by Web Spiders and placed in the storage devices. Links are then indexed to develop relationships based on keywords, title, Meta tags and filters. In practice, keywords, title, and meta tags are often used to filter information from collected web links to create relevant relationship over the indexed links. The relevant relationship is then presented in the statistical manner known as rankings or scores. The methods used to search text in the WWW are briefly described as follows.

Keyword searching uses “index words” to search. An example of keyword searching engine is AltaVista (http://www.altavista.com). The disadvantage of this method is it may involve irrelevant results when the query is returned. For example the search using the words “cold”, “flu” and “influenza” may not have the same meaning in the returned documents, but it refers to the same keywords in the query. On the other hand, concept-based searching uses “meaning” rather than words. An example of concept-based search engine is Excite (http://www.excite.com). Excite returns hits on documents that are relevant to the subject. An example of words “cold” in documents with other words such as climate, and medical will return different subject in relation to the contexts.

Another method is refining search, which uses user-defined options. Generally, this method of searching is capable to include or/and exclude more than one word to the search terms. An example of inclusive and exclusive refine search is achieved by specifying the logical terms such as AND, OR, NOT, +, -, and quotation marks in query. At the present time, most of the search engine includes refining search method.

Relevancy ranking is another method of searching. It returns a list with search term frequency. Most search engines use relevancy ranking to determine the relevance of a document. For example, Lycos (http://www.lycos.com) ranks hits according to the number of times of keywords appearance in indices of the document such as headers, titles or text (Barlow 2002). Finally, some search engines such as AltaVista use Meta tags to index web documents. Meta tags are information about a document rather than the document content. The meta element can be used to identify properties of a

document (for example author, expiration date, a list of key words, etc.) and assign values to those properties (Lassila and Swick 1999). For example, to specify the author of a document, one of Meta element called "author" is used as follows:

```
<META name="author" content="Albert Einstein">
```

Table 2.1 shows the features of major search engines currently available in the WWW.

	Type of Search	Search Option	Search Refining	Domain Searched
AltaVista	Keyword	Simple, Advanced, Search Refining	AND, OR, NOT, NEAR, *	Web, Usenet
Excite	Concept-based, Keyword	Simple, Advanced	AND, OR, AND NOT, +, -	Web, News, Pictures, MP3s and classified ads
Google	Keyword	Basic, Advanced	Full Boolean	Web, news, stock quotes, PDF documents
Lycos	Keyword	Basic, Advanced	Full Boolean	Web, news, stock quotes, jobs, weather, multimedia
HotBot	Keyword	Simple, Modified, Expert	Boolean-like choices in pulldown box, Phrase	Web
Yahoo	Keyword	Simple, Advanced	Boolean AND, OR	Yahoo's index, Usenet, E-mail addresses

Table 2.1 Summary of major search engines

2.3.1 Ontology

Ontology is an emerging research area that attracts research interest among researchers in Information Systems (IS), particularly its applications on the Semantic Web (Zuniga 2001). To achieve explicit conceptualisation of knowledge and to resolve the ambiguity problem in knowledge representation, researchers began to apply an interoperable interpretation approach based on the concept of ontology. There is also a need to share meanings of terms or a set of names as those used in a given domain by an individual or community. In order to effectively deploy a set of shared terms, and a clear understanding of the particular domain has to be accomplished by the community of practice. In other words, a way to conceptualise the given domain by community has to be published and broadly agreed among interest groups. It is known as the use of ontology.

2.3.1.1 Definition

The Oxford dictionary describes ontology as "The science or study of being" (Hornby 1995). A quite similar point of view is introduced to Artificial Intelligence (AI) for representation of the world in late 1960s (McCarthy and Hayes 1969). Then the term ontology is derived from cognitive semantics that related to expressions to conceptual structures. It can be seen as one stream of cognitive science that "specifics a relation between linguistic expression and referents of the expressions" (Gardenfors 1995). Sowa (2000 p.51) notes that "Logic itself has no vocabulary for describing the things that exist. Ontology fills that gap: it is the study of existence, of all the kinds of entities – abstract and concrete – that make up the world". Gruber (1993) defines ontology as a formal explicit specification of a shared conceptualisation. It can be used to define a computer-usable definitions of basic concepts in the domain

knowledge and the relationships that exist in the domain (Gómez-Pérez 1999a, W3C 2002). In the literature ontology is applied as a form of meta-knowledge (knowledge about knowledge). It allows conceptualisation to be structured in systems such as knowledge-based or knowledge management systems.

2.3.1.2 Why Ontology?

Recently, various application of ontology has been reported in the literature (Aart et al. 2002, Abasola and Gomez 2000, Aitken and Reid 2000, Altman et al. 1999, Benjamins et al. 1998, Davies et al. 2002, Farquhar et al. 1995, Fensel et al. 2002, Motta et al. 2000, Nour et al. 2000, Zúniga 2001). A long-term objective of such work is to enable reusable knowledge, meta-knowledge, and sharing knowledge to be achieved. Examples of applications include Ontoligua (Farquhar et al. 1995), RiboWeb (Altman et al. 1999), MELISA (Abasola and Gomez 2000), OntoWebber (Jin et al. 2001), On-To-Knowledge (Fensel et al. 2002), OntoShare (Davies et al. 2002), OntoEdit (Sure et al. 2002) and Protégé (Stanford Medical Informatics 2003). Table 2.2 shows five common reasons why ontology is used.

1. To share common understanding of the structure of information among people or software agents.
2. To enable reuse of domain knowledge.
3. To make domain assumptions explicit.
4. To separate domain knowledge from operational knowledge.
5. To analyse domain knowledge.

Table 2.2 Reasons why ontology is used (Noy and McGuinness, 2001)

The first reason of allowing sharing of common understanding of the structure of information among people or software agents has been cited as one of the common goals of using ontology (Gruber 1993b). The reason being interoperability over

distributed environment requires 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 among the use of various terms. The second reason of enabling reuse of domain knowledge is one of the main reason of driving current trends of developing reusable ontology (Ding and Fensel 2001, Klein and Fensel 2001). In this case, organization or business can extend an existing ontology with or without adding or modifying new definitions or terms they need. An example is Universal Standard Products and Services Classification (UNSPSC) is already developed by Stanford University, and it can be easily adapted to any business in order to meet the needs of desired definitions or terms. The third reason of making domain assumptions explicit is not only significant in terms of underlying implementation of computing systems but it is also useful to help new users who must understand the meaning of the terms used in the domain (Everett et al. 2002, Noy and McGuinness 2001). Fourthly, separating domain knowledge from the operational knowledge is another common use of ontology (Noy and McGuinness 2001). As an example of the web ontology language such as OWL (Ontology Web Language) starts to support reasoning system that provides comprehensive access to knowledge expressivity. OWL is capable of formalising a domain by defining classes, the relationships among classes and the properties associated with those classes. It allows us to reason about domain knowledge. According to McCarthy and Hayes (1969), a reasoning system is capable of defining problem, find its causes and solutions in the domain or the world it represented. Finally, some researchers (Heflin and Hendler 2000, Klein and Fensel 2001, Norman 1995) argue that ontology can be changed, evolved, and revised over time. In this case, there is a need to extend existing

ontology. Thus, formal analysis of existing ontology and analysing domain knowledge is cited as one of the reasons to use ontology (Boicu et al. 2001).

2.3.1.3 Structure and Design of Ontology

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 scheme and logic theories (W3C 2002). However according to Gruber (1993b), when formal ontologies are designed, they must satisfy the following criteria: clarity, coherence, extendibility, minimal encoding bias and minimal ontological commitment. 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 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 specifyed 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.

Ontology design in ontological engineering is concerned about the principle design approaches (Gómez-Pérez 1999b). Currently, there are a number of approaches in ontology design: inspirational, inductive, deductive, synthetic and collaborative

approaches (Holsapple and Joshi 2002). The inspirational approach focuses on the need of an ontology using individual imagination, creativity and personal views on the domain of interest, whereas the inductive approach concentrates on observing, examining and analysing a specific case in the domain interest so that a specified case can be applied to other cases in the same domain. The deductive approach is concerned with the general adoptive principles using example of filtering and distilling of the general notions. The synthetic approach is a method used to characterise a relatively widespread use of ontologies, rather than separate ontologies, to express synthetic relationships of multiple ontologies. It means that the synthetic concerns more than true or falsehood to its meaning alone. In other words, identifying meaning with reference is useful to distinguish whether the meaning of a term or phase is the same as what it refers to. Finally, the collaborative approach focuses on cooperative method of ontology design and development using multiple viewpoints such as iterative improvement of ontology development, and its consensus-building mechanism. It can be linked to one of ontology initiatives: "to share common understanding of the structure of information among people or software agents". At this point, we believe that an analytic approach to ontology design gives additional usefulness. For example, how are terms, phrases or statements that have the same meaning be differentiated? Is a term "cold" refers to meaning of "A condition of low air temperature" or "any of several diseases caused by bacteria or viruses and marked especially by respiratory symptoms". The former meaning might be referred to cold, but there is no guarantee that the latter meaning, known as "flu or influenza", refers to the term "cold" without having explicit relationship with the context of medical or medicine. Therefore, analytical approach is necessary and valuable to ontology

design. This approach can be linked to one of the ontology initiates: “to analyse domain knowledge”.

Jasper and Uschold (1999) classify four main categories of ontology applications. These four categories are neutral authoring, specification, common access, and ontology-based search. Neutral authoring is an idea of using a single ontology, and applied to various operational applications. This way, having a single ontology in multiple application means that it improves use and reuse of knowledge as well as maintainability of its applications. Ontology Builder and Ontology Server are examples of this application to provide industrial strength ontology management as well as other commercial applications (Das et al. 2001). Ontology Builder is a multi-use collaborative ontology generation and maintenance tool, and Ontology Server is a server that drives e-commerce applications with ontologies (Das et al. 2001). Ontology as a specification is specifying a term or a name according to the requirement so that it gives reliable specification of the requirement. The following gives examples of ontology that have been applied or used as a specification. Xerox developed a Knowledge-sharing system called Eureka. It contains 40,000 technical documents and it allows intellectual capital to be shared. Ontological design is used to represent natural language to get rid of redundancies in Eureka’s documents (Everett et al. 2002). Other well specified ontologies developed by Stanford Medical Informatics are Health Level Seven (HL7) data types and Top-Level Reference Information Model (RIM) classes, and UNSPSC. It is clearly beneficial that rich knowledge expression in terms of specifying a set of terms or vocabularies can prevent potential ambiguities in software development. Common access to information is an idea of using shared or mapped ontologies to enable various

developers or multiple target applications to have access to heterogeneous source of information (Jasper and Uschold 1999). An example of ontology that has been developed for the WWW is Ontolingua. It is developed by the University of Stanford to provide users with the ability to publish, browse, create and edit ontologies stored on an "Ontology Server" online (Farquhar et al. 1995) (See Figure 2.4). The architecture of Ontolingua development is designed to build and reuse commonly acceptable reusable and adaptable ontologies in a variety of community over the Internet. Finally, ontology-based search is applied to minimise searching time. In order to meet the comprehensive requirement, it may be necessary to merge one or more ontology application scenarios discussed above.

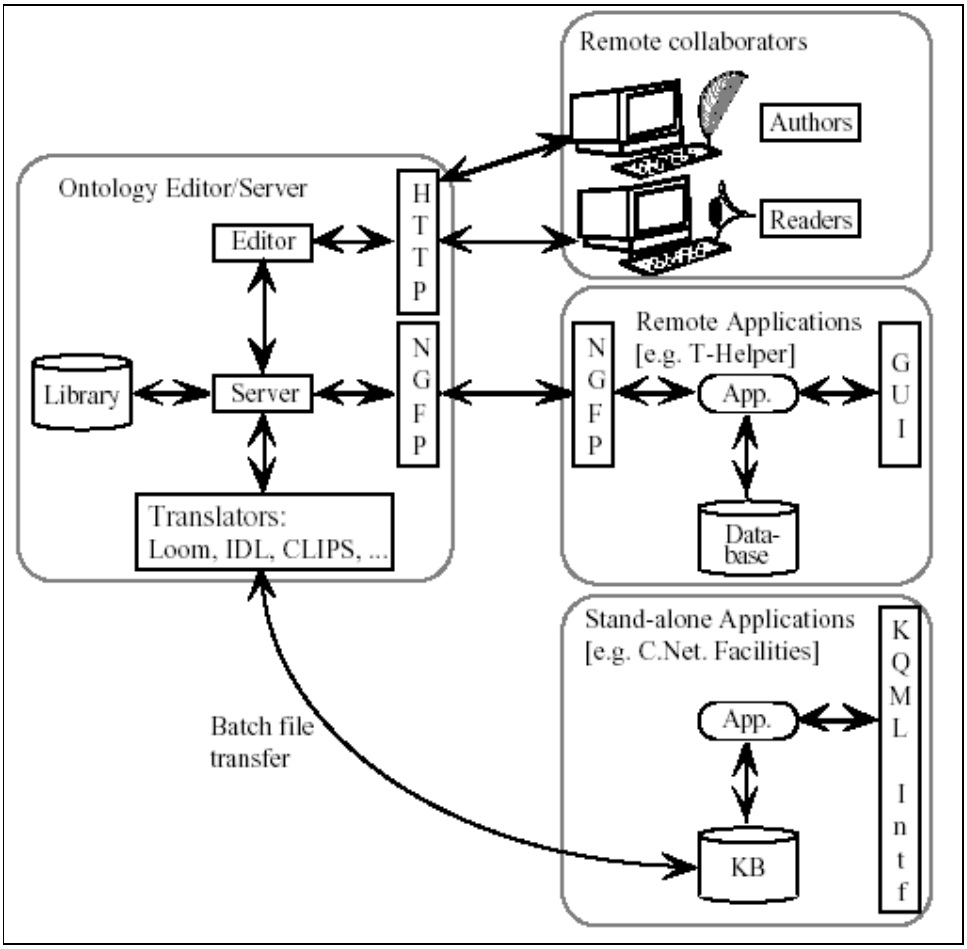

Figure 2.4 Architecture of Ontology Server (Source: Farquhar et al. 1995 p.5)

2.3.2 The Semantic Web

This section discusses the Semantic Web that provides standardised platform for ontology languages.

Berners-Lee, Hendler and Lassila (2001b) describe the Semantic Web as an extension of the current web in which information is given well-defined meaning, better enabling computers (in specific software agents) and people to work in cooperation. It allows software agents to be able to communicate with other software agents. This is achieved by software agents sharing terms of mapped or merged ontologies defined usually in a language of XML or/and the Resource Description Framework (RDF).

We will briefly discuss concept related to agents first. There are a number of definitions for agents in the literature. For example agents have been described as: "Special purpose" (Smith et al. 1994), "perceives and acts in its environment though preceptors" (Russell and Norvig 1995), "autonomously" (Maes 1995), "behaving its dynamic property of functions such as social ability, reactivity, pro-activeness" (Hayes-Roth 1995, Wooldridge and Jennings 1995). In particular, agents can be considered as a software, intelligent, or learning agent. Software agent is a piece of software application that acts like human? It can be used to assist knowledge acquisition processing such as locating and filtering data. It is possible to use software agent to manipulate distributed data in the WWW. Software agents can be designed to replace repetitive human tasks such as searching databases, retrieving and filtering information, and delivering it back to the end user. Intelligent agent has

human like characteristics such as autonomy, temporal continuity, reactivity, and is goal driven (Maes 1995, Wooldridge and Jennings 1995). A learning agent has an inherent characteristic of the human beings to adapt its behaviour in order to improve its performance (Konar 2001). With learning capability, the learning agent adapts the abstract patterns of relationship in the domain autonomously.

Tim Burners-Lee drew the Semantic Web as a "layer cake" in terms of knowledge terms, ontology vocabularies, logic and rules as shown in Figure 2.5 (Berners-Lee et al. 2001b). The feature of the Semantic web can be seen as bringing a structure to the meaningful content of web pages so that machine-centred initiatives can be achieved. The machine-centred initiatives mean that access of software agents to this meaningful content of web is more systematic and knowledge-rich on the standard platform. The purpose of such an approach can be seen as an effort to improvement of knowledge acquisition (Benjamins and Fensel 1998, Fensel et al. 2002) or simply information retrieval of a system (Aitken and Reid 2000, Farquhar et al. 1995, Wache et al. 2001).

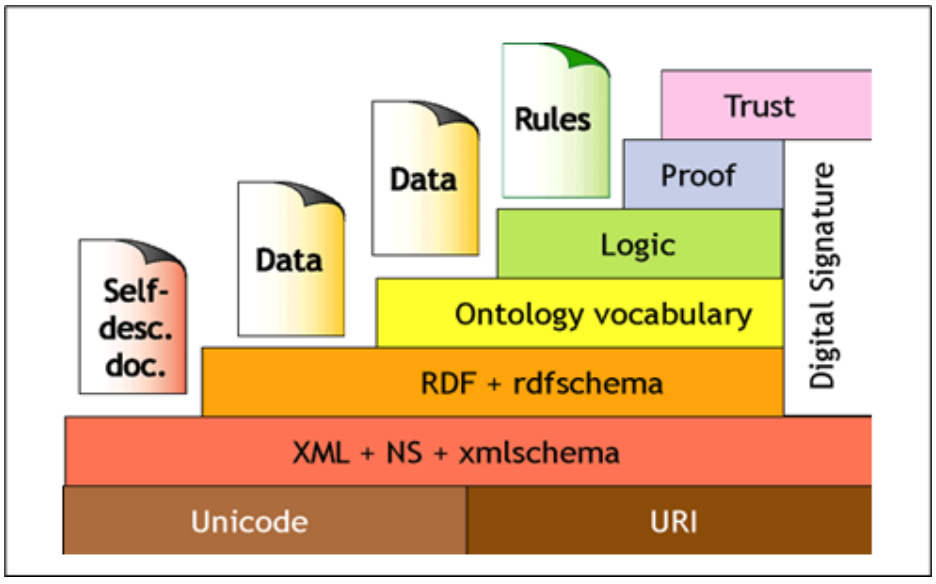

Figure 2.5 The Semantic Web "layer cake" (Source: Berners-Lee et al. 2001b)

In relation to the Semantic Web and ontologies, James Hendler forecasts a vision of the Semantic Web Ontologies. The first step of its use is to create web pages with ontological information (Hendler 2001). It means that logic experts and/or individuals will develop decentralised small size of ontologies. In addition, one or more ontologies will be linked to other ontologies in order to share repositories. The second vision of the Semantic Web Ontology is the definition of services in a machine-readable form (Hendler 2001). It means that the ontologies can be used to agree on terms and/or constraints for web services. For example, software agents or e-commerce programs are able to share and reuse B2B e-commerce transactions based on machine-readable ontologies. The final vision of the Semantic Web Ontology is the use of logic and agents (Hendler 2001). Logic and rules are being used to improve the description of software agents' services. Software agents are communicating with other agents using the terms in ontologies, exchanging portion of other agents' ontologies, and merging other agents' ontologies.

Traditionally, computer systems are developed as a human-centered view of knowledge. The early file-based computer systems represent and store data in a file using a string data, known as records. Sequential data processing is normally used to process the logical related data elements (Partridge and Hussain 1995 p.195). Then the need of flexible random data processing grew as the complexity of relationships among files increases. The increase complexity of records and the interrelationships between files and records have resulted in the development of database management systems (DBMS), in particular the relational DBMS, to support complex data

manipulation and representation. Table 2.3 summarises the relationship of computer systems in terms of how data are viewed from the human and machine perspectives. The human perspective refers to the way human read and understand the data, whereas machine's perspective refers to the way the computer and software agent read and understand data.

	Human User		Machine User	
Type	**Readable**	**Understandable**	**Readable**	**Understandable**
File-based systems	✓	✓	✓	✗
DBMS	✓	✓	✓	✗
WWW	✓	✓	✓	✗
WWW + Rules and Logic + Ontology	✓	✓	✓	✓

Table 2.3 Computer systems from human and machine user perspectives

In order to share knowledge, in particular over the distributed networked environment such as the WWW, it needs a mechanism that can retrieve information locally and globally. Communication lines are established among human users to enable information to be retrieved. Software robot, which is regarded as machine user, is also developed to retrieve information over the distributed network. Various terms have been used in the literature to refer to software robot or machine user. These include terms such as "Web Spiders", "Web Crawlers" and "Web Wonderers". Basically, software robot is a piece of software that collects the WWW hypermedia (Heinone et al. 1996, Koster 1995). Software robot often uses AI techniques such as rules and logic to extract knowledge from the WWW. It travels through the WWW to retrieve information within domain knowledge of interests. However the machine user's retrieval task is often restricted by limited knowledge domain. The challenge is to ensure software robot does not wonder around irrelevant WWW websites when retrieving information. There is a need to ensure relevant and meaningful information

is retrieved. Very often the human user poorly describes the keyword or the combination of keywords used to retrieve the information in the WWW. Software robot such as Web Spider then retrieves information based on the given keywords. However a lot of the times, same keywords can have different meanings depending on the context. An example is the use of keywords such as "cold" or "flu" or "influenza" which has the same meaning in commonly used human-context but it may not have the same meaning in different knowledge domain of interest such as in the medical or weather contexts. If the Web Spider is not able to differentiate the meaning or the context in which the keywords are used or referenced to, then it may respond to irrelevant references and retrieve irrelevant information. In such cases, KMS is said to be not able to respond to what the user wants. This scenario is often described as the knowledge representation problem in the literature.

One of the challenges in the development of KMS is the difficulty in accurately representing knowledge explicitly, particularly in the domain of interest. Knowledge or domain expert often finds that it is not possible to expressively describe his or her knowledge to the knowledge engineer about the knowledge in a particular domain of interest. In a distributed environment, such as the WWW, it is also important to use a knowledge representation that can provide machine understandable format so that capturing new knowledge and updating knowledge can be implemented by machine such as software bots or agents once a well-defined set of domain of interest in the community of practice is presented. The term "machine understandable format" refers to a format that supports and allows machine such as software agents to understand the content of the texts. The purpose of supporting machine understandable format meets the requirement of ontological design concept as it can

improve the knowledge acquisition process using software agents. Sometimes, agents need to compare or merge information of two or more terms to check whether it is refer to the same thing. This happens when logical expressions, such as AND or OR, are used by the software agents. This means knowledge representation will become more complicated when relationships are taken into consideration. In such cases, knowledge representation has to be more systematic. A well-defined and agreed terminology should be adopted within the domain of interest or within the community of practice. This can be achieved using ontology. Ontology not only provide better support for standardisation of terminologies, it also provides an opportunity knowledge sharing and reuse. For that reason, ontology is said to explicit formal specification of a shared conceptualisation (Boicu et al. 2001, Gruber 1993a).

Most recently, a number of research groups have been focusing on ontology languages in terms of standardising knowledge representation language for the Semantic Web. DAML and OWL both are developed to bring a communication bridge among agents, and building its equivalent machine-readable and understandable mechanism over the web. The US Defence Advanced Research Projects Agency (DARPA) released a draft language known as the DARPA Agent Markup Language (DAML) in October 1999. It is an effort to bring out the core of the language that is the best features of the Semantic Web includes RDF, SHOE and OIL. The PLUS group at the University of Maryland originally developed Simple HTML Ontology Extensions (SHOE) in 1995 (Luke and Heflin 2000). It provides a hierarchical classification mechanism for HTML documents. In other words, it allows us to define valid classifications, relationships, and inferred rules so that it is possible to incorporate machine-readable semantic knowledge in web documents. OIL stands

for Ontology Inference Layer, is a language for describing ontologies on the web. In December 2001, Joint US/EU committee on Agent Markup languages yields a more stable version of DAML (Berners-Lee et al. 2001a). It is called DAML+OIL. Both DAML and DAML+OIL are built up on RDF and have a description logic base. The Ontology Web Language (OWL) 1.0 was released by W3C, and it can be used to support more accurate web search, intelligent software agents and knowledge management (W3C 2002). It compromises the best features of previous DAML+OIL and the Semantic Web needs so far.

2.3.3 Evolution of Markup Language

During the last several decades, evolution of markup language has been changed. In 1969, the Generalised Markup Language (GML) is introduced by the IBM team (Goldfarb 2001 p.19). The intention was to construct system that interchanges information with each other. In 1978, the American National Standards Institute (ANSI) started research in the field of generic document markup. Afterwards, Goldfarb joined the ANSI working group, Standard Generalised Markup Language (SGML) is started as a project for a text description language standard based on GML in the Computer Languages for the Processing of Text committee (Goldfarb 1993). In 1980, the first draft of the SGML standard was published, then in 1983 its working draft became an industry standard named GCA 101-1983 (Goldfarb 1993). In 1984, the International Organisation for Standardization (ISO) joined, and then in 1986 ISO standard of SGML (named ISO 8879:1986) published (Goldfarb 1993). In 1989, Tim Berners-Lee and Robert Caillau developed a language, HTML, to share hyperlinked text document over his system (Goldfarb, 2001). HTML became a simple solution to

link “information system that would be accessible across the wide range of different computer systems in use at CERN” (W3C 1999b). It was proposed to the World Wide Web Consortium (W3C), and became today’s HTML. The revision of HTML 4.0, HTML 4.01 was published in December 1999 to serve “a wide range of features reflecting the needs of a very diverse and international community wishing to make information available on the Web” (W3C 1999b). In 1998, a subset of SGML is designed to be the eXtensible Markup Language (XML) to response to large scale of digital publishing, and information exchange on the web (W3C 2000). In 1999, RDF model and syntax specification is released as a W3C recommendation to provide “machine-understandable” metadata (W3C 1999a). RDF is a framework for Meta data. In RDF, the basic model consists of resources, properties and statements (W3C 1999a).

- Resources: all things being described by RDF expressions
- Properties: a specific aspect, characteristics, attributes or relations used to describe a resource.
- Statement: defines a resource with a named property and the value of that property

With the RDF model, it is possible to offer a structure to assist with interoperability between agents and web applications for the exchange of web-based machine-readable information. Therefore, RDF and XML provide simple and effective solution to facilitate interoperability in KMS based on metadata models.

2.4 Agent

2.4.1 Definition

In 1958, John McCarthy developed a program called “Advice Taker” which was designed to use knowledge to search for solutions to problems. It is worth noting that

this approach takes a program that learns from its experience, and takes certain actions for the human user. As a result, it refers to the first application that has agent concepts in mind. In general, the definition of the agent is derived from the concept of agency. The concept of agency refers to "employing someone to act on your behalf". In computing, "someone" can be a computer program that is capable of acting intelligently. In fact, the agent's levels of intelligence is various in the literature. A more satisfying explanation will be discussed next section. Russell and Norving notes (1995 p.33) that "An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors." In Object-Oriented (OO) concepts, it can be explained as event-condition-action paradigm. An event is anything that occurs to change the environment. When an event happens, the agent may response to that event after determining what the condition of the world is. According to McCarthy and Hayes (1995 p.1), "the world must be have a general representation of the world in terms of its inputs and interpreted." At the end, the agent must be able to take action according to the event. This action is not just pressing a "ENTER" key on the keyboard it is a result from the event that the agent relies on and behaves rationally.

2.4.2 Characteristics

As an introduction in the previous section, a brief definition of the term agent is presented in relation to a number characteristic that is often used to describe an agent in the literature. The term "characteristics" often refers to properties or intelligence or taxonomies of agents. The term is various amongst researchers but it is all useful efforts of adding values to agents. The first is "autonomous", according to Maes (1995 p.108). "Autonomous agents are computational systems that inhabit some complex

dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed." It means that for an agent to be autonomous, it has to be a separate process or thread in OO concepts so that it can be ran in long-term and autonomously. A thread is defined as a single sequential flow of control within a program (Sun Microsystems Inc. 2003). The second term is "intelligence", it can be seen as an introduction of logic to the agent that is dependent on sophisticated reasoning and learning capabilities. It is worthy noting Hayes-Roth's idea of reasoning in agent. She notes that "intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to effect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions. Conceptually, perception informs reasoning and reasoning guides action, although in some cases perception may drive action directly." (Hayes-Roth 1995 p.3) The third term is "mobility". It is ability to transport itself across network so that a mobile agent can move from one site to another and send data to and retrieve data from the user or another agent. Importantly, a weak notion of agency described by Wooldridge and Jennings (1995 p.4) is well-known in terms of describing properties might be in an agent.

- autonomy: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
- social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language (Genesereth and Ketchpel, 1994);

- reactivity: agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;
- pro-activeness: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.

In 1996, Franklin and Graesser (1996) further classified agent properties. The following Table 2.4 shows list of several properties, and other names to those properties if there is any, and meaning of those properties.

Property	**Other Names**	**Meaning**
reactive	sensing and acting	responds in a timely fashion to changes in the environment
goal-oriented	pro-active purposeful	does not simply act in response to the environment
temporally continuous		is a continuously running process
communicative	socially able	communicates with other agents, perhaps including people
learning	Adaptive	changes its behaviour based on its previous experience
mobile		able to transport itself from one machine to another
flexible		actions are not scripted
character		believable "personality" and emotional state

2.4 Classified agent properties

2.4.3 Environments and Tools

By agent building environments and tools, this means a system that allows one to program in terms of some of the concepts discussed above. The first, an OO programming language such as C++ and Java is often used because it supports many attributes that make its idea for implement agents. This includes support for threads, distributed objects and network. The second choice of building agents is to use Agent-Oriented Programming (AOP). Shoham (1993 p.6) introduced the key idea of AOP, and proposed three components that might be used in AOP system.

- A restricted formal language (including several modalities such as belief and commitment) width clear syntax and semantics for describing mental states;
- An interpreted programming language in which to define and program agents, with primitive commands
- An "agentifier", converting neutral devices into programmable agents.

New style of AOP language called 3APL (Triple-a-p-l) is proposed for implementing cognitive agents in 1998. According to Hindriks et al.(1999 p.2), the idea of this language is to "incorporates features from both imperative and logic programming… which allows for an elegant description of many agent oriented features." It extends values of agents to the fields of AI. Some example of the agent environment and tools are following. The IBM Agent Building and Learning Environment (ABLE) is a tookit for building multiagent autonomic systems that is Java-based framework for developing and deploying hybrid intelligent agents and agent applications (Bigus et al. 2002). JATLite (Java Agent Template, Lite) is developed at Stanford University,

and is focus on communication related issues such as Agent Message Router (AMR) to send and receive messages, transfer files, and exchange information to other agents. (Joen et al. 2000).

2.5 Conclusion

To ensure knowledge sharing and knowledge reuse can be achieved in KMS, especially in a networked environment such as the WWW, CBR techniques can be applied to provide an opportunity to allow new knowledge to be updated, stored and retrieved in the KMS. CBR techniques can be applied to knowledge management to provide effective problem solving solutions, creating rich knowledge repositories, and decision supporting mechanisms. It can be used to support knowledge management tasks (Aha 1999).

Knowledge management can be viewed as a process of multiple activities. A design of CBR system can be embedded to retrieve and adapt knowledge management activities. CBR can be used to improve knowledge acquisition processes of KMS by allowing new knowledge to be updated and learned. To support CBR, a need of a well-defined set of domain interest in community has to be presented clearly to prevent ambiguity. Ontology provides an opportunity and is a feasible approach used to conceptualise a set of terms in the community of practice. Ontology provides new opportunities to prevent ambiguities in knowledge representation by supporting well-agreed terms or vocabularies. This gives better support for knowledge acquisition processes. To support a standardised platform to ontology to create and machine-understandable format, the Semantic Web features can be used. The purpose of

machine-centred approach is to provide a knowledge rich mechanism to enable software agents to function in a distributed networked environment.

Chapter 3 Application of CBR and Ontology

This chapter discusses the application of CBR and ontology to allow knowledge sharing and reuse to achieve in KMS.

Section 1 discusses the framework which proposes the use of CBR technique to KMS so that knowledge sharing and reuse can be achieved. The section also discusses the application of agents, ontology and the Semantic Web features to the framework.

3.1 Proposed Research Framework

Literature review in Chapter 2 shows that CBR techniques can be applied to KMS to allow knowledge sharing and reuse to achieve. CBR techniques can be applied to resolve new problems by applying retrieved cases stored in the case repository and apply or adapt past solutions to new problems. In addition, we propose the use of software agents to assist in the development of KMS. We use the KMS cycle of Turban and Aronson (2001) as a basis in our framework. In our approach, knowledge is stored as cases in the knowledge repository. Figure 3.1 shows how the agents are applied at different stages of the KMS cycle using CBR techniques. The following describes in details how the agents work.

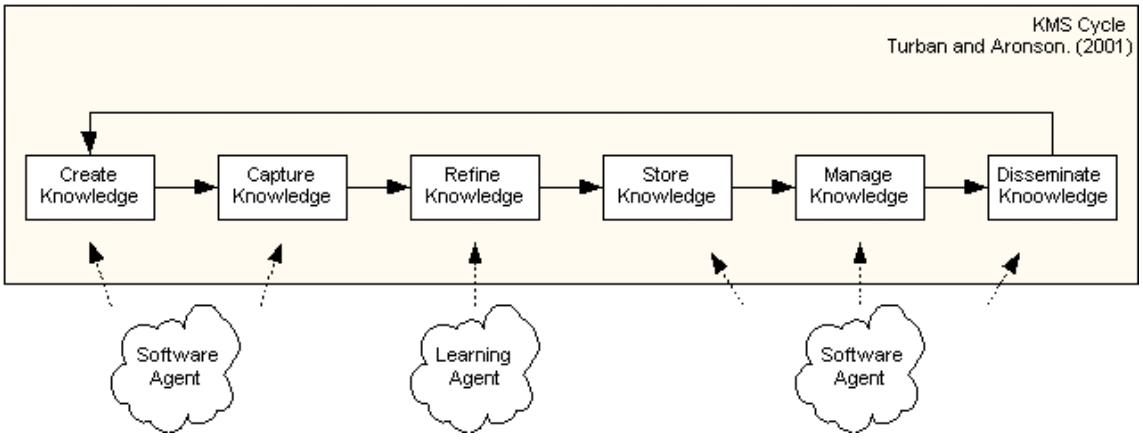

Figure 3.1 Application of CBR techniques in KMS cycle

There are four phases in the CBR cycle: *retrieve, reuse, revise and retain*, and six phases in the KMS cycle. There are similarities between CBR and KMS cycle. Table 3.1 compares the phases of CBR and KMS cycle. The *retrieve* phase in the CBR cycle matches with the *create knowledge* and *capture knowledge* phases in the KMS cycle. The *reuse and revise phases* in CBR corresponds to the *refine knowledge* phase in KMS. Finally, the *retain* phase in the CBR cycle matches with the *store, manage and disseminate knowledge* phases in KMS.

CBR Cycle	KMS Cycle
Retrieve	Create knowledge, Capture knowledge
Reuse	Refine knowledge
Revise	
Retain	Store knowledge, Manage knowledge, Disseminate knowledge

Table 3.1 CBR Cycle vs. KMS Cycle

In Figure 3.1, a software agent is developed in the *Create Knowledge* and *Capture Knowledge* processes of the Turban and Aronson's KMS cycle. This agent will perform the *retrieval* process. We propose to use the approach discussed by Aamodt and Plaza (1994) and Watson (1997), in which a combination of inductive retrieval

and nearest neighbour retrieval techniques are applied to allow pattern to be matched and similar cases to be compared. According to Kolodner (1993), pattern matching is the process of comparing two cases to one another and then determine their degree of match. In this proposed framework, pattern matching refers to the process of comparing attributes of target case and original case to each other and determining their degree of match. Attributes of the target and original cases are associated with problem description and solution. The inductive retrieval technique can be used for pattern matching. The nearest neighbour retrieval technique is used for similarity assessment. In similarity assessment, if similar cases are found, then the software agent will retrieve existing cases from the knowledge repository. This way existing knowledge is retrieved and its solution can be shared and reused. Thus, the *reuse* phase of the CBR cycle is also implemented in this instance.

An agent is applied in the *Refine Knowledge* process in the above KMS cycle. If similar case is not found, then a learning agent will be deployed. The learning agent allows new knowledge to be learned based on new case behaviour. For example, an element of the previous solution can be substituted or inserted with new element. The learning agent uses the derivational adaptation technique of the *revise* phase of CBR cycle to adapt new knowledge. This way new knowledge represented in the form of new cases with new problem and solution descriptions can be added to the knowledge repository. This process allows knowledge in the KMS to be refined and updated.

However, before new knowledge is stored in the *Store Knowledge* step of the KMS cycle, a software agent that is capable of performing indexing will be deployed first. The task of this software agent is to optimise the retrieval and accessibility of new

knowledge in the knowledge repository to allow fast and efficient retrieval of knowledge during the *Manage Knowledge* and *Disseminate Knowledge* phases. During these phases, new cases are stored in the knowledge repository.

The application of agents allows the above processes to be performed independently and automatically over the standardised Semantic Web platform. The retrieval agent is used to find the best or the closest matches case in the repository. The reuse agent allows the solution from past cases to be applied. The revise agent is used to adapt a new case so that it can be retained and stored in the knowledge repository.

The above framework will be implemented as a student enquiry system which is a web-based application. The prototype is developed using rapid prototype development approach. Rapid prototyping give us "an iterative process of systems development in which initial requirements are converted to a working system that is continually revised through close work between an analyst [or a developer] and users" (Hoffer et al. 1996, pp.29-30). The purpose of using the prototyping approach is to catch up evolving issues in the prototype development process. The prototype will apply ontology over the Semantic Web platform. An advantage of using ontology is to define the concepts relevant to domain of knowledge. One of the most common goals of ontology development is to share common understanding of the structure information, and to enable reuse of domain knowledge among people and software agents (Gruber 1993b).

Chapter 4 Prototype Development

This chapter discusses the prototype development of a student admission enquiry system.

The chapter is organised as follows. Section 1 discusses the domain of the prototype. The discussion includes the admission criteria and the decision making process of the student admission. Section 2 presents the knowledge representation of the domain. Section 3 discusses ontology design. Section 4 discusses the use of agents in the prototype. Section 5 concludes the chapter.

4.1 Domain

The domain of the prototype 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 Master of Information Systems courses at the University of Wollongong. The Graduate Diploma course is specifically designed for those students 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 countries from China, Indonesia, Thailand, India, Pakistan and others. One of the problems encountered is how does one determine that a qualification from country X is equivalent to the local Australian University qualification. Very often, the academic qualification from each country is different. For example, is a 4-year undergraduate degree from country X and a 3-year undergraduate qualification from country Y equivalent to a 3-year undergraduate degree in Australia? One of the methods used by the Department and the University to determine the degree from a country is equivalent to the local undergraduate qualification is that the student must have completed twelve years of schooling before they commence 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. Thus the admission officer needs to have some knowledge on the educational qualifications from different countries in order to assess whether the applicants qualification is considered to be equivalent to that of the local undergraduate qualification.

To complicate the matter, the general admission criteria listed above 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, applicants who do not possess the equivalent University degree, but have qualifications such as a Diploma in Information Technology or Diploma in Computer Science with relevant working experience in Information Technology (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. Figure 4.1 shows a sample list of special cases in which the students can gain admission to the course. Thus the admission officer needs to consider the case individually based on individual's academic qualifications, professional certificate and employment history.

1. 3-years Diploma (after 12 years schooling) + relevant work experiences of at least 2 years
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 study in other courses such MIB, or MBA with good results.
5. 2-years bachelor/diploma (after 12 years schooling) will not be accepted to postgraduate courses.

Figure 4.1 Admission to Graduate Diploma in Information Systems – special cases

The Master of Information Systems course is a postgraduate coursework program offered by the Department. It 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 also be admitted to the course. Figure 4.2 shows a sample 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 experience 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 4.2 Admission to Master of Information Systems - special cases

The admission officer often receives enquiries (via telephone, email, or mail) from prospective students from different countries to enquire about the course and to find out if they are eligible to apply for the course. A student admission enquiry system is to be developed to assist the admission officer in handling these 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 a formal application to the University for assessment. As a large majority of the applicants are from overseas, 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.

4.2 Knowledge Representation

In our prototype, we present knowledge using cases. The applicant's details are stored as problem description in the case and the outcome of whether she/he is admitted to the course (Graduate Diploma and Master 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). These 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 transformed 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, and also used to reference the ontology. Figure 4.3 shows a sample of user input screen. Only attributes that are relevant to the problem description (course, qualification, professional certificate and working experience) will be extracted from the user input screen and transformed into RDF format so that it can be made available for agents to perform the CBR tasks. Other attributes such as the University that the applicant attended previously will be stored in the database system so that it can be retrieved when the admission officer requires this information.

A software agent is developed to transform the user query in XML format to RDF format using the eXtensible Stylesheet Language Transformations (XSLT). We do this so that only relevant information that is referenced to the ontology is extracted. To implement the XML conversion, Xerces Java 2 parser is used. It is one of the Apache XML project to provide standardised XML solution (The Apache XML Project 2002-03). 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.

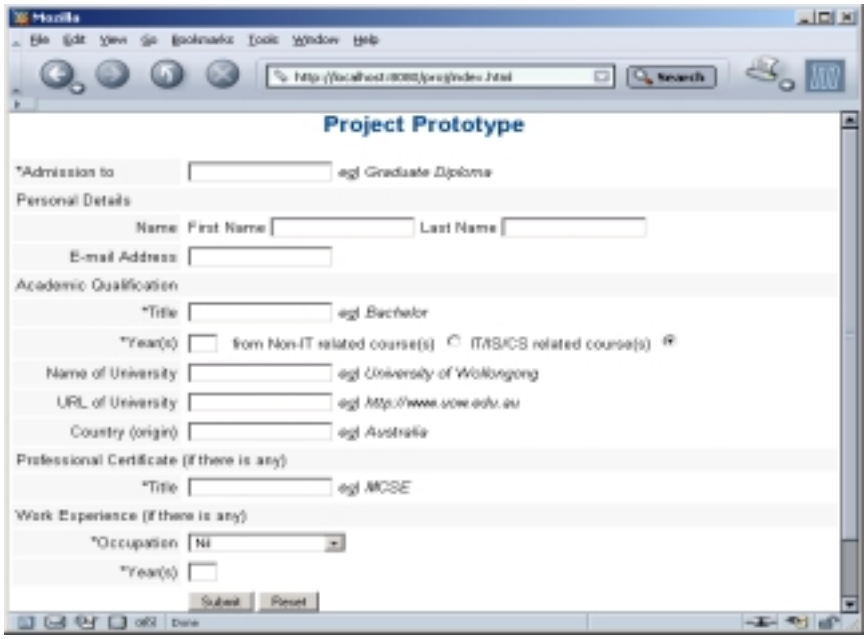

Figure 4.3 A user input screen

The ontology in the system 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.

4.3 Ontology Design

The ontology that we use in this project is simple in concept. Ontology is used to define the concepts relevant to the admission criteria to the course. The ontology is developed to define the concepts relevant to the admission criteria which include the definitions of equivalent University qualifications, relevant or recognised professional IT certificates and relevant IT working experience. We use the *inspirational approach* which is based on designing ontology using individual imagination, creativity and personal views of the domain of interest (Holsapple and Joshi 2002). In

this case, the ontology is developed based on the conceptualisation of the knowledge expert: the admission officer.

The use of ontology allows standardised concepts of academic qualifications, professional certificates and work experience to be referenced in the system. In addition, the use of ontology also allows easy maintenance of the system. If there is any new or additional requirement (such as adding new professional IT certificates to the list), it can be added to the system by adding new entry to the ontology. We have also included rules in the ontology component of the system. For example, the twelve years of schooling is added as a rule to determine the concept of "equivalent qualification".

We have developed the prototype in the Semantic Web. In addition to the use of agents as shown in Figure 3.1 (see p.61), we have also developed software agents to perform additional tasks which include: verifying and validating inputs, and delivering answers or outputs to the user. All the software agents developed in the system share terms defined in the ontology using XML and RDF. The use of the Semantic Web enables software agents to communicate more coherently with other agents.

Figure 4.4 shows a sample RDF data model for the course attribute. The basic RDF data model consists of three object types: resource, properties and statement (W3C 1999). Resource is anything that is being described (for example: a web page, a part of web page or a web site). Properties are the characteristics, attributes or relations that are used to describe a resource. The property defines the meaning for the

resource, which may include permitted values, type or relationship with other properties. An RDF statement refers to a specific resource together with a named property plus the value of that property for that resource (W3C 1999).

The RDF model can be represented pictorially using node and arc. The node is shown as ellipse and represents the resource. In Figure 4.4, the resource is Master of Information Systems, and is identified by the Uniform Resource Identifier (URI): "http://…/MasterInformationSystems". The resource has properties that include name (N) and Universal Resource Locator for the web page (URL). The notation we use here are: AO for Admission Ontology, N for Name, and URL for the Universal Resource Locator for the web page (in this case, it is http:// ../moc.htm). Figure 4.4 also shows that the property Name is made up of compound name of Full Name (FN) and Short Name (SN). The full name is "Master (of) Information System" and the short name is "MIS". Note that the blank node in Figure 4.4 is used to represent a compound name that has no URI.

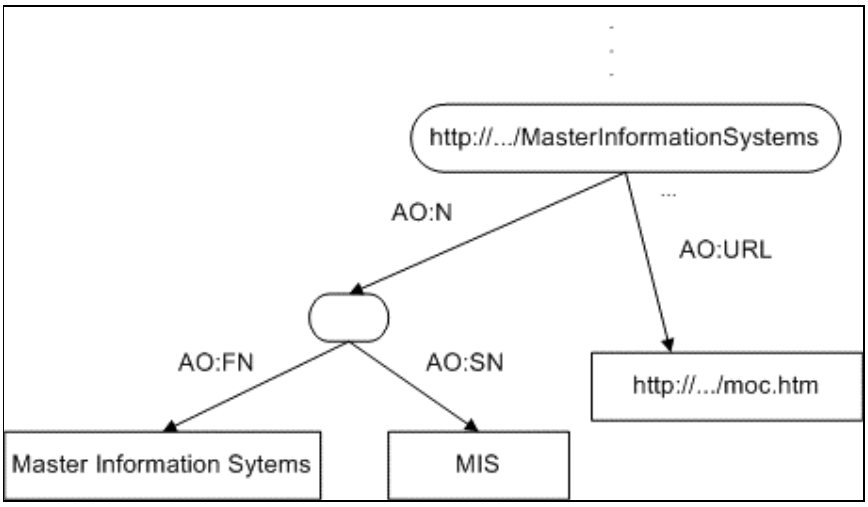

Figure 4.4 A sample RDF representation of course

Figure 4.5 shows a sample screen shot of RDF/XML representation for the course Master of Information Systems. This representation can be read as "http://…/MasterInformationSystems", and it has a full name "Master of Information System" and a short name "MIS", and an URL " http//:../moc.htm".

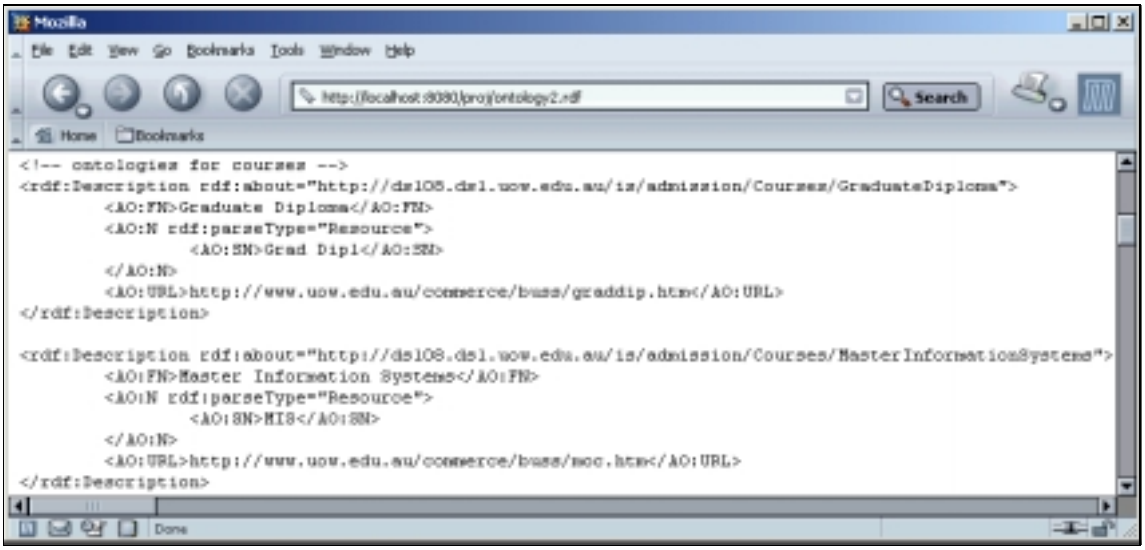

Figure 4.5 A screen shot of RDF representation of the course

We have included a list of abbreviations for each of the concept or attribute related our domain in the ontology. This way, the system is able to detect any abbreviation or partial names for each of the concept. Example of abbreviation used includes "mis" refers to "Master of Information Systems" and "dipl" refers to "Diploma" (Figure 4.5). It has been suggested that a strategy using aliases can improve the vocabulary problem in human-system communication (Furnas et al. 1987). In our prototype, we achieve this through the use of full and short names. In addition, we have included the concepts and terms associated with Information Systems department such as "School" and "About us" in the ontology to provide associated background knowledge that are relevant to the course (see Figure 4.6).

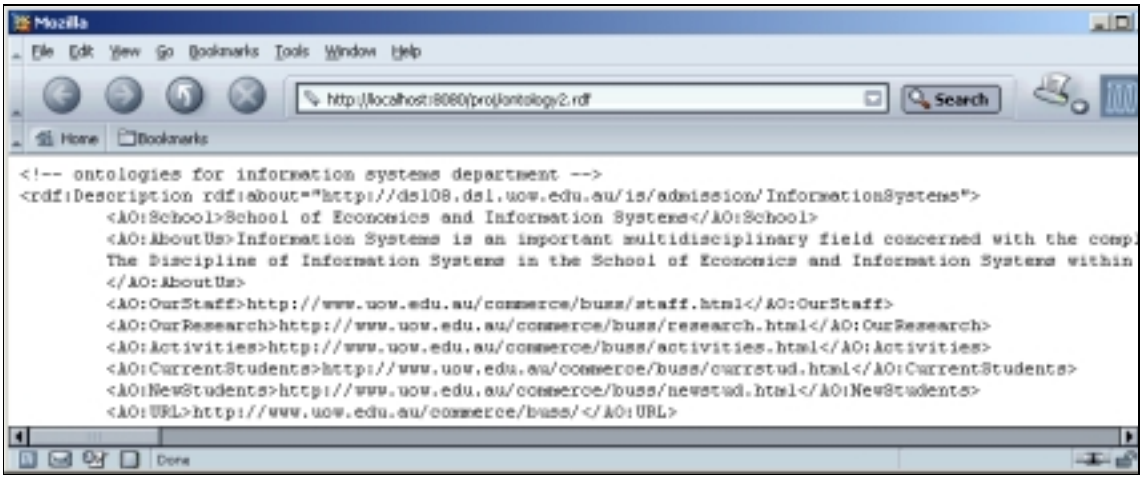

Figure 4.6 A sample screen shot for the representation of Information Systems department

As described in Section 1 of this Chapter, relevant IT professional certificates can be taken into consideration when an applicant does not have the required equivalent University qualification. The criterion is that the IT professional certificate must be one of the recognised qualifications that have been officially included in the list

maintained by the Department. At the moment, there are four IT professional certificates that are recognised and accepted by the Department: *Microsoft Certified Professional + Internet (MCP+I), Microsoft Certified System Engineer (MCSE), Microsoft Certified System Engineer + Internet (MCSE+I) and Sun Certified Programmer for Java 2 Platform (SCJP)*.

Figure 4.7 shows a sample RDF representation for professional certificates. The resource is the node with URI "…/Certified/2002". Figure 4.7 shows an example of using containers to refer to a collection of resources. There are three types of container objects: Bag, Sequence and Alternative (W3C 1999a). 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 (W3C 1999a). Figure 4.7 shows an example 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 (W3C 1999a). The membership relation between this Bag container resource and other resources are named as "_1", "_2", "_3" and "_4" (W3C 1999a). The representation modeled in Figure 4.7 can be read as: "The certificates that were certified in the year 2002 are MCP+I, MCSE, MCSE+I and SCJP". Figure 4.8 shows a sample screen shot of RDF/XML representation of IT related professional certificates.

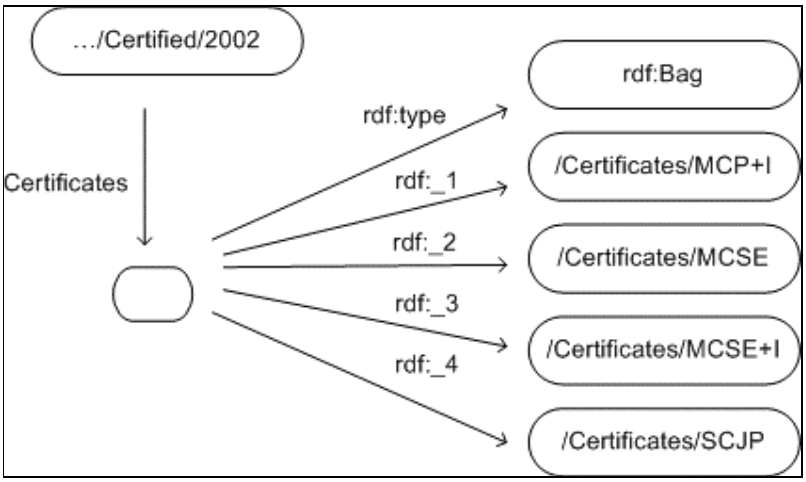

Figure 4.7 A sample RDF representation of IT related professional certificates.

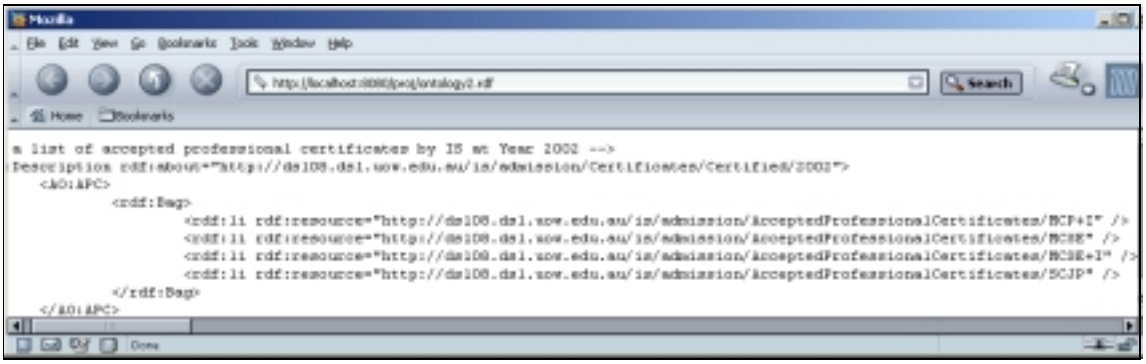

Figure 4.8 A sample screen shot of RDF/XML representation of IT related professional certificates

4.4 CBR Design

As explained in Chapter 3, we use agents to carry out the tasks allocated to in our prototype. Software agents are used in the following tasks: retrieval, reuse and revise of the CBR cycle. Figure 4.9 shows a graphical view of prototype processes in which agents are used.

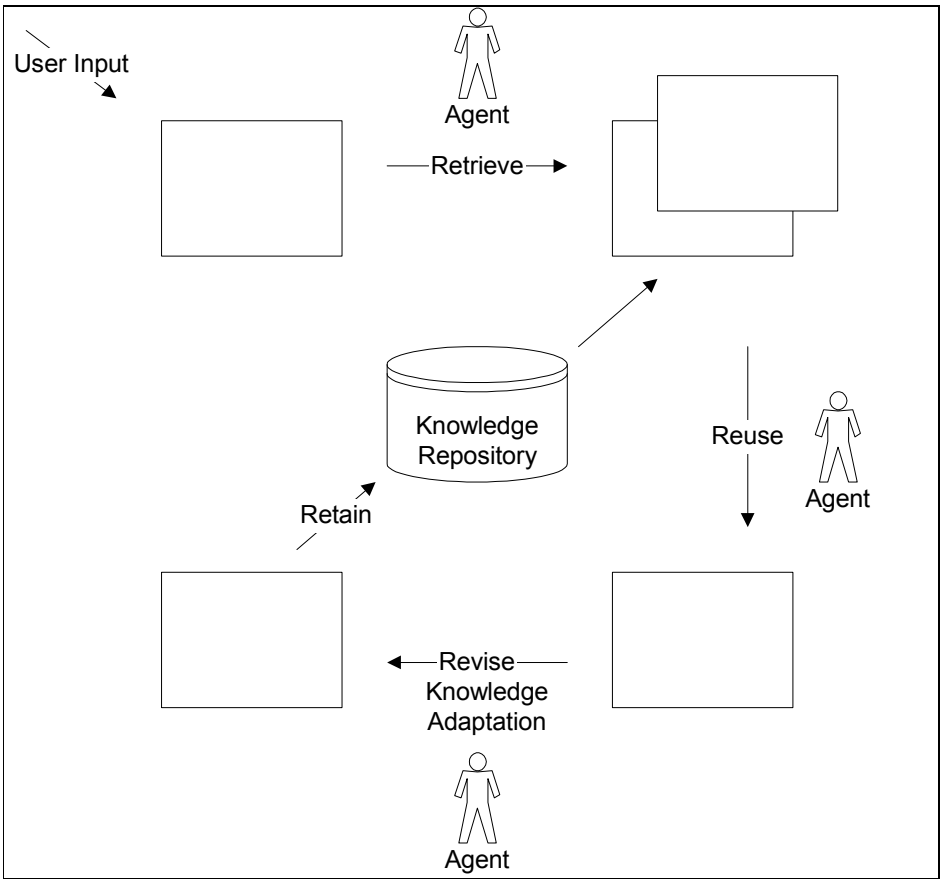

Figure 4.9 A Graphical View of Prototype Processes

The retrieval agent that performs similarity assessment is developed using RDQL (RDF data Query Language) of Jena Semantic Web Toolkit (HPL 2002) to find the best or the closest matched case in the knowledge repository. Jena Semantic Web

Toolkit is a JAVA API (Application Program Interface) developed by Hewlett-Packard Labs (HPL). Jena Semantic Web Toolkit includes built-in support for RDF containers, integrated RDQL, and support for storing DAML ontology in a model (HPL 2002). This is the first software agent introduced in Figure 3.1 (see p.61) in relation to the *Create Knowledge* and *Capture Knowledge* phases in KMS cycle. 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).

Figure 4.10 shows the flowchart for the retrieval phase of the CBR. The user query is performed using RDF query (RDQL). The system considers each user query as a new case (target case). In general, there are two types of cases stored in the case repository: the original case and the adapted case. The original cases are cases that were given by the domain expert before the system is developed. In our case, the domain expert has provided fifteen original cases. On the other hand, the adapted cases are cases that have been adapted by the system as a result of the revise and retain phases of the CBR cycle. Note that the original cases have precedence over the adapted cases in the retrieval process. Figure 4.10 shows that 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, the system 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 cases are found, the system will process similarity assessment, and the case that is found to be the nearest to the target case will be retrieved from the knowledge repository.

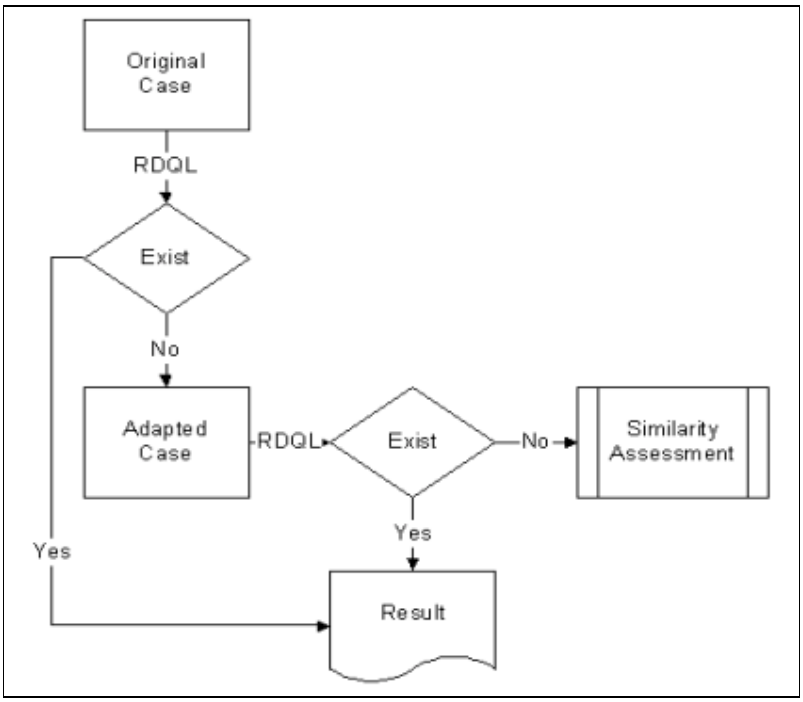

Figure 4.10 A flowchart for case search

The similarity assessment is performed using the nearest-neighbour technique (See Equation 1 in Chapter 2, p.30). It processes retrieval of cases by comparing a list of weighted attributes in the target case to source cases in the knowledge repository. Weighting is assigned to the attributes in the form of relative importance. In our prototype, the domain expert determines the weighting of each attribute. In the prototype, the person's academic qualification and the number of the year in the degree are considered twice as important compare to other attributes and are given a weighting of 2. Note that the weight is assigned an integer value to simplify the calculation. The formula in Equation 1 (see p.30) calculates the distance of the target case from the original case. Table 4.1 shows a sample list of attributes used in the prototype. For example, the course attribute in Table 4.1 refers to the course which the applicant is applying, it can be the Bachelor or Diploma or Master course.

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

Table 4.1 A sample of tabulated attributes of prototype

After the case is retrieved, 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 she/he intends to apply when the matched or nearest matched case is found. The reuse phase of the CBR cycle 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.

For the revise phase, we use derivation adaptation technique in our prototype. An agent is developed in our prototype to perform this task. This is the function of the learning agent process described in Figure 3.1 (see p.61) which is related to the phase

of *Refine Knowledge* of KMS cycle. 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).

Our adaptation rule is as follows. 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, then the system will check for qualification of the professional certificates from the ontology which maintains 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 will be stored for manual review by the admission officer. An example of this is when the system cannot determine the academic qualification of the applicant based on past cases in the knowledge repository. If after the review, the admission officer finds that the case is of relevance then it will be added to the knowledge repository manually. This can be done periodically by the admission officer. An example of this when the system cannot determine the academic qualification of the applicant from a new country (that is country is not found in the cases).

In the retain phase, indexing is performed when cases are stored in the knowledge repository. Indexing allows cases to be retrieved more efficiently. This phase is related to a software agent that performs *Store Knowledge, Manage Knowledge* and *Disseminate Knowledge* phases in terms of KMS cycle which are described in Figure 3.1 (see p.61).

4.5 Conclusion

This chapter describes the prototype development for the student admission enquiry system. The CBR technique, ontology and agents are used to perform the proposed framework to allow solutions of similar cases to be reused, revised and new cases to be adapted in the knowledge repository.

Chapter 5 Prototype Evaluation

This chapter discusses the evaluation of the prototype based on the fifteen original cases (see Appendix A) given by the domain expert. We will evaluate the prototype based on the following four scenarios to demonstrate the application of the CBR techniques: (i) a matched original case is found; (ii) a matched adapted case is found (iii) a best matched case is found; (iv) a new case is adopted.

This chapter is organised as follows. Section 1 shows evaluation results of the prototype system. Section 2 concludes the chapter.

5.1 Evaluation of the Prototype

We will consider four sample screen outputs for the following scenarios: (i) a matched original case is found; (ii) a matched adapted case is found (iii) a best matched case is found; (iv) a new case is adopted.

5.1.1 A Matched Original Case

This scenario is used to evaluate the case retrieval process of the CBR cycle. When an original case is found in the knowledge repository. Using Figure 4.9 (see p.76) as a guide, when an original case from the knowledge repository is found to match with the target case (that is the user query), then the software agent will retrieve the original case and apply the solution found in the original case to the target case. As explained in Chapter 4, the agent matches patterns on the original case to the target

case to perform this task. A sample screen output for this scenario is shown in Figure 5.1. For the purpose of presentation in this thesis, we have produced the output as shown in Figure 5.1 to demonstrate the result. In the actual prototype, the user will only see a message that advise the applicant whether she/he is eligible or ineligible for admission to the course that she/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*".

A sample user query for this scenario is: "*I have a qualification of a three-year Bachelor of IT and I have professional certificate that includes Microsoft Certificate Professional Systems Engineer. Will I be eligible for admission the Master course in Information Systems?*" Based on the original cases that were stored in the knowledge repository, we expect case 15 to be matched with this user query and the solution that was stored in this case can be retrieved and applied to this query. The expected output is the applicant satisfies the admission criteria and thus is eligible to apply for admission to the Master of Information Systems course.

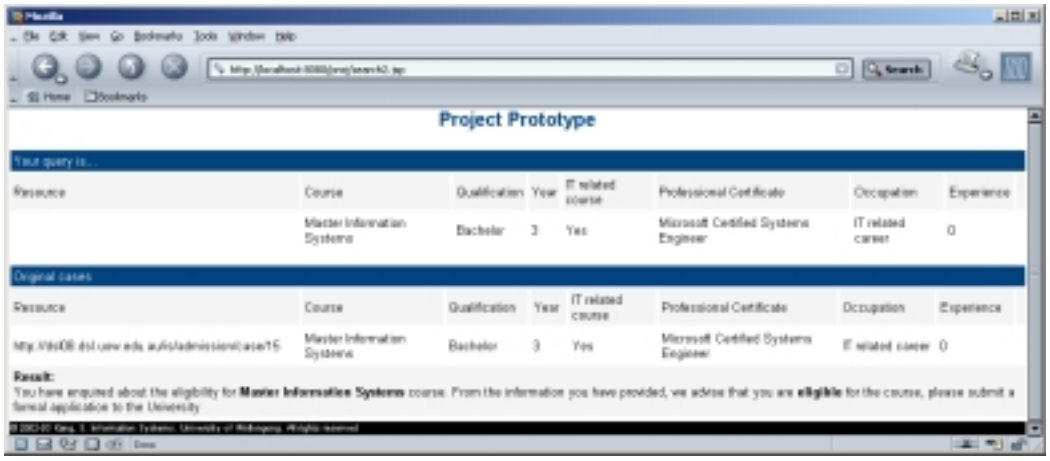

Figure 5.1 A matched original case of user inquiry

5.1.2 A Matched Adapted Case

This scenario is used to evaluate the case retrieval process when original case is not found to match with the user query (target case). In this instance, the agent will attempt to find a matched adapted case in the case repository. 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 for admission 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 thesis, we have included the result "No original case is found!" in Figure 5.2. 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 5.2.

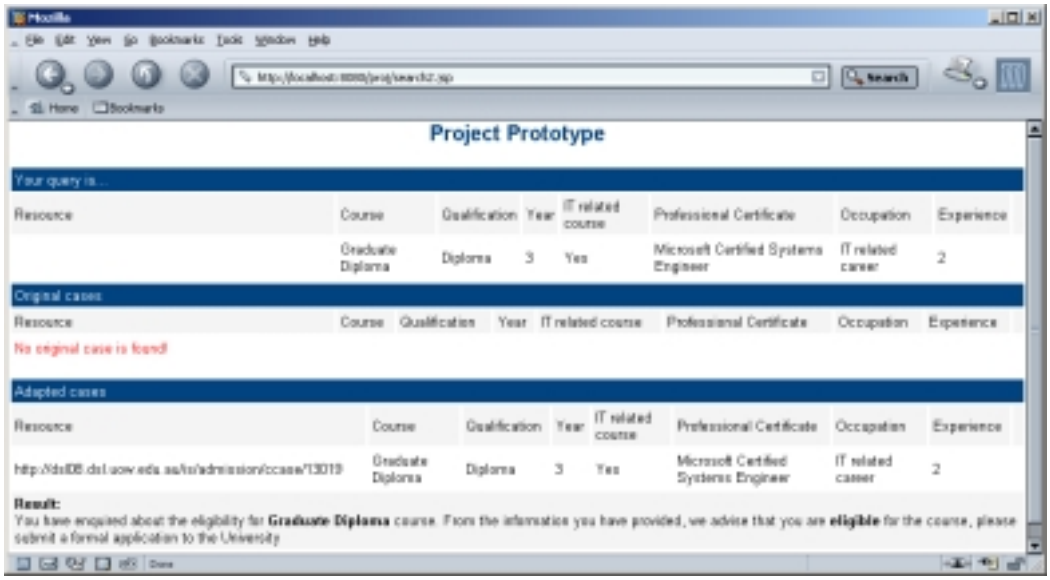

Figure 5.2 A sample output of matched adapted case

5.1.3 A Best Matched Case

The third scenario is when none of the case in the case repository (original and adapted cases) is found to match with the user query. In this situation, the system triggers an agent to process 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 Equation 1 in Chapter 2 (see p.30).

As explained in Chapter 4, the weightings of each attribute show the importance of the attribute in decision-making. In our system, the criteria of qualifications of the applicants and the number of years of the qualifications (undergraduate or diploma) are considered to be twice as important as the other attributes.

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 Certified Systems*

Engineer. Am I eligible for admission to the Graduate Diploma in Information Systems course?" Figure 5.3 shows a sample output of the value computed using the equation of similarity assessment in Chapter 2.

For the purpose of demonstration in this thesis, we have included in the result to show "No original case is found!" and "No adapted case is found!". In addition, the calculated similarity assessment values for all original cases have been included in Figure 5.3. In this example, only case numbered 7 is found to be equally close to the target case. The solution will be verified and revised to return the outcome for the target case. Once the solution for the target case is verified and evaluated using pattern matching 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 knowledge repository as a new adapted case.

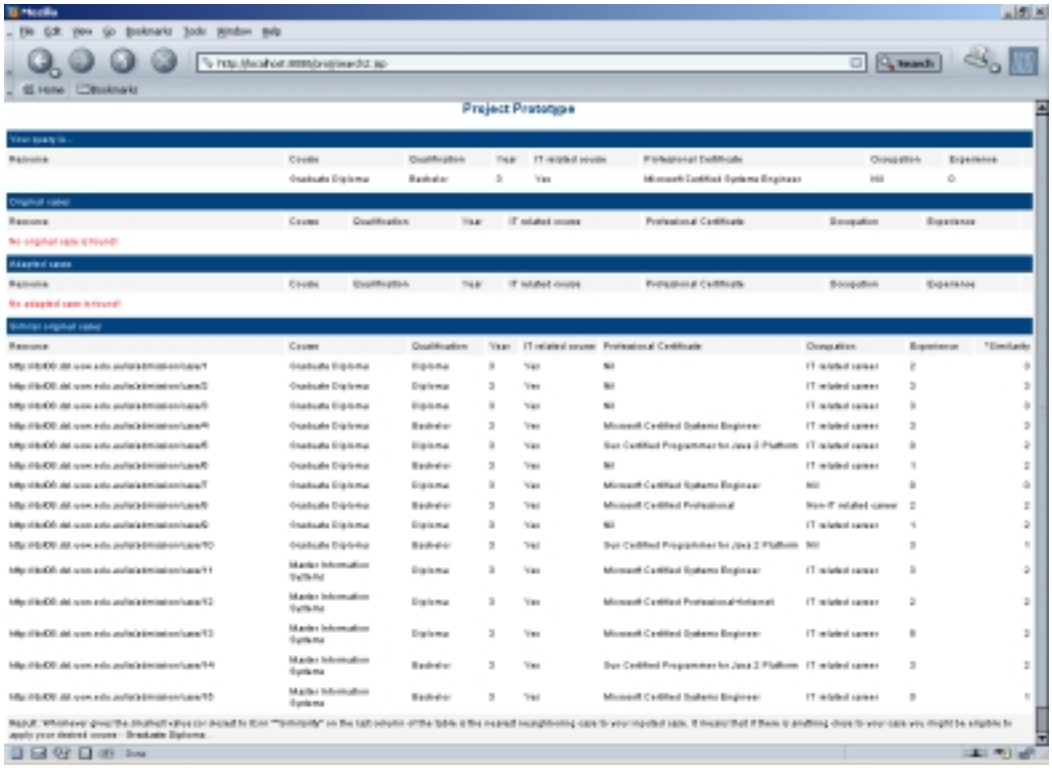

Figure 5.3 A sample output of similarity assessment

5.1.3.1 Similarity Assessment With Weighting

The following section discusses how the similarity assessment is calculated. We will use the sample example shows above in which the distance of target case is calculated from the original case number 1 (as shown in Appendix A). Case number 1 can be stated as follows "*The person in this case who has successfully admitted to the Graduate Diploma course in Information Systems, has 3 years diploma in IT, and 3 years working experience in IT related career.*" As explained previously, the academic qualification and the length of the course are considered to be twice as important as other attributes and are thus assigned weighting of 2, other attributes are given weights of 1. Table 5.1 shows the values used in the calculation.

Attribute Name	**Weighting**	**Original Case Number 1**	**Integer Values (assigned to each attributes)**	**Target Case**	**Integer Values (assigned to each attributes)**
Course	1				
		Graduate Diploma	1	Graduate Diploma	1
Qualification	2				
		Diploma	1	Diploma	1
Year	2				
		3	3	3	3
IT related	1				
		Yes	1	Yes	1
Certificate	1				
		Nil	0	Java	1
Occupation	1				
		IT-related career	2	Non-IT related career	1
Experience	1				
		2	2	2	2

Table 5.1 Values assigned to target case and original case number 1

Using Equation 1 in Chapter 2, the calculation for the weighted distance (D_w) of a target case ($_T$) from an original case ($_O$) is as follows:

$$
\begin{aligned}
D_w \quad = \quad & ((1^* \text{Course}_T) - (1^*\text{Course}_O)) + \\
& ((2^*\text{Qualification}_T) - (2^*\text{Qualification}_O)) + \\
& ((2^*\text{Year}_T) - (2^*\text{Year}_O)) + \\
& ((1^*\text{IT}_T) - (1^*\text{IT}o)) + \\
& ((1^*\text{Certificate}_T) - (1^*\text{Certificte}_O)) + \\
& ((1^*\text{Occupation}_T) - (1^*\text{Occupation}_O)) + \\
& (1^*(\text{Experience}_T) - (1^*\text{Experience}_O)) + \\
= \quad & (1\text{-}1) + (2\text{-}2) + (6\text{-}6) + (1\text{-}1) + (0\text{-}1) + (2\text{-}2) + (2\text{-}2) \\
= \quad & \text{-}1 \text{ Unit}
\end{aligned}
$$

In this case, the distance between the target case and original case is –1. This means the person who is presented in the target case does not have similar requirements to original case. The reason for rejecting this case is that the applicant needs to have at least two years working experience in the area of IT, but this applicant has only 1 year

in non-IT related one. Therefore, she/he does not possess sufficient qualification to be admitted to the Graduate Diploma course. In other words, the application is rejected.

5.1.3.2 Similarity Assessment Without Weighting

Figure 5.4 shows a graph of weighted distance of the target case from the fifteen original cases for the above query. The line "No weighting" shows that no weighting is applied to both the target and original cases. The line "With weighting" means that weighting is applied to the attributes as explained previously. In this case, the academic qualifications and the year of the course are given a weight of 2 in both the target and original case. For the presentation of this thesis, we have included the calculations of the 15 cases. From the graph, it can be seen that case number 2, 3 and 14 are the nearest neighbouring cases to the target case in no weighting is applied to the attributes. These are determined by the value of 0. It means that the person presented in the target case is eligible to apply a Graduate Diploma course according to the solution presented in the original two cases of number 2 and 3. Note that in this case, original case number 14 should be discarded because it is for admission to the Master of Information Systems course, whereas the target case is to seek admission to the Graduate Diploma course. Thus, it can be seen that by adding weighting to the attributes the results of calculation using similarity assessment equation can make a difference in terms of the cases that were returned.

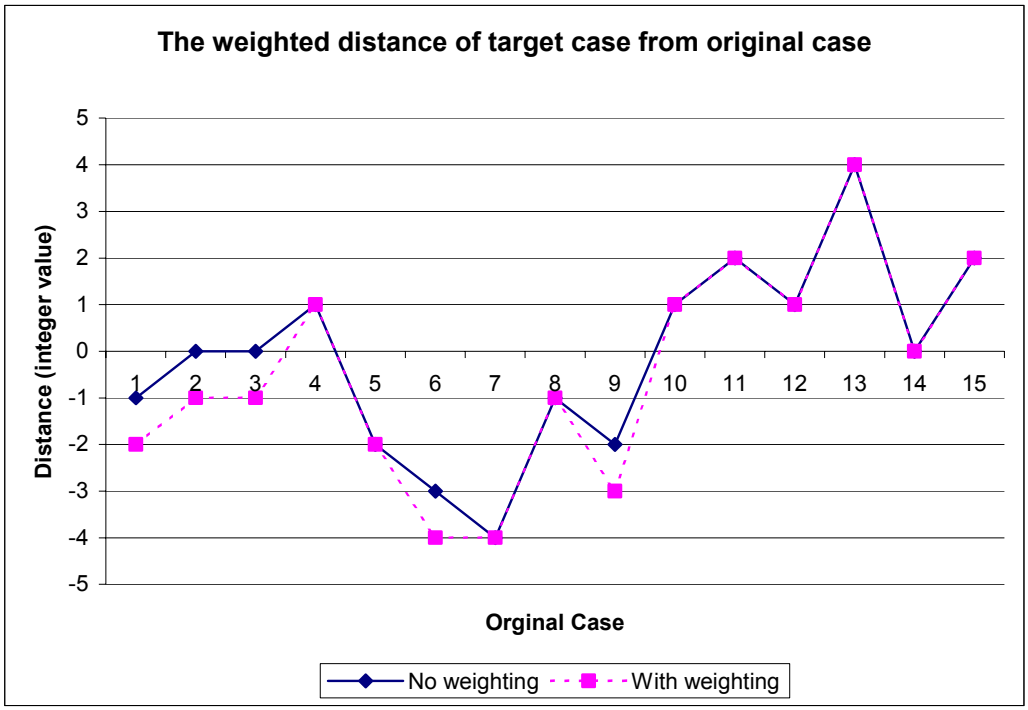

Figure 5.4 The weighted distance of target case from original

5.1.4 An Adapted New Case

The fourth scenario presented here is to adopt a new case in the knowledge repository. In our prototype, an agent is developed for this task. The target case is verified and evaluated to match the correctness of the admission to the course. As explained in Chapter 3, pattern matching is used to verify and evaluate admission criteria. Once the verification is completed, the target case is retained in the knowledge repository using the derivational adaptation rules. These rules are derived from the requirement of admission to the Information Systems courses given by the admission officer which is given in Chapter 4.

A sample user query for this scenario is as follows: "*I have a qualification of a three-year Diploma degree majoring in IT and I have a Microsoft Certificate Professional*

+ Internet certificate. I also have three years working experience as a programmer. Am I eligible for admission to the Graduate Diploma in Information Systems course?"

In this example, there is no original and adapted case that is found to match with the target case in Figure 5.5. Then, the system will try to find a best-matched case stored in the case repository. Figure 5.5 shows a sample output of the calculated similarity assessment values for all original cases have been included. The nearest matched cases are case 1, 2, 3 and 4. The solution from one of these cases can be adapted to the target case. Thus, when the target case is verified and found to match the correctness of the admission criteria to the course, then the agent will attempt to adopt the target case as a new adapted case in the knowledge repository.

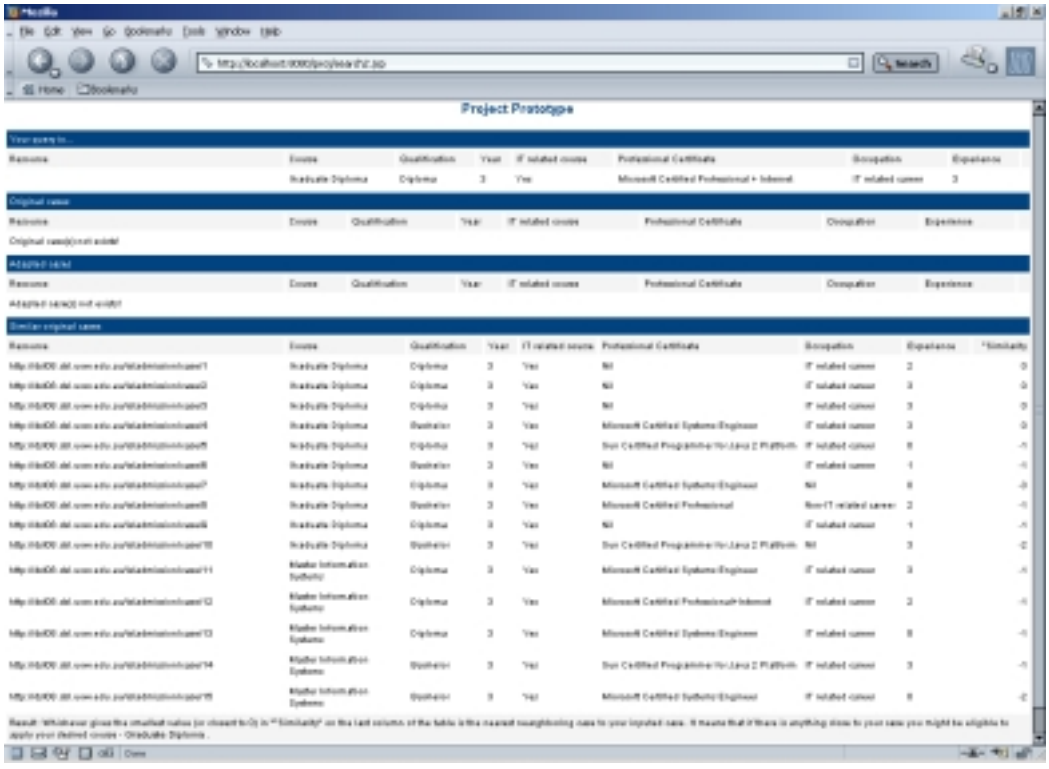

Figure 5.5 Result of similarity assessment

Thus, after the agent has adopted the target case as a new adapted case and stored the case in the knowledge repository, the system is now able to find the adapted case if another similar query is entered. In this case, the solution from the newly adapted case will be returned.

If a similar sample user query is entered (similar to the one entered in Figure 5.5), "*I have a qualification of a three-year Diploma degree majoring in IT and I have a Microsoft Certificate Professional + Internet certificate. I also have three years working experience in IT. Am I eligible for admssion to the Graduate Diploma in Information Systems course?*", then we expect the solution of the newly adapted case to be returned. Figure 5.6 shows the output for this query. As there is no original case found to match with the target case, so the system will try to look for a matched adapted case stored in the knowledge repository. In this case, the newly adapted case is returned.

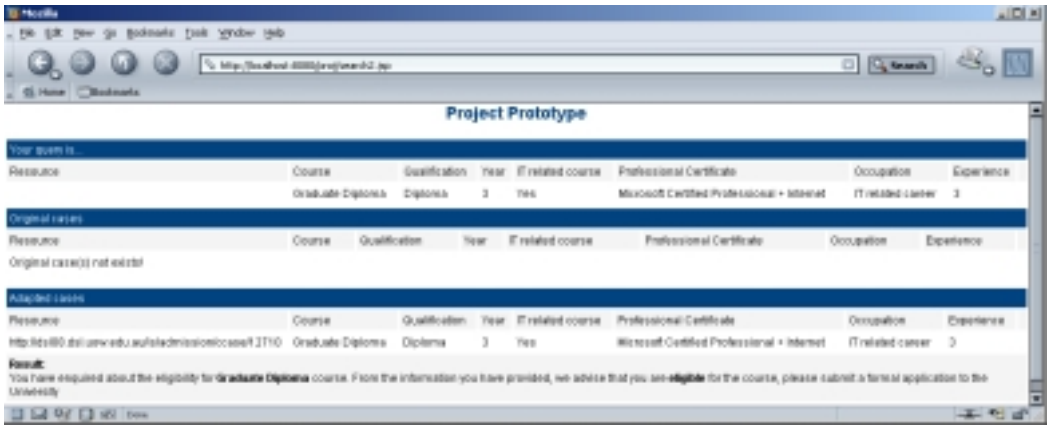

Figure 5.6 After newly adapted cases

5.2 Conclusion

The chapter presents the evaluation results of the prototype. It can be seen that the prototype was able to adapt new knowledge using the knowledge adaptation process of CBR cycle. This way, old cases can be used to solve new problems and new cases can be adapted as new knowledge in the knowledge repository to solve future relevant cases. Our results also show that the assignment of weighting to various attributes can influence the calculation of the similarity assessment from the target case to the original case. For example, if the weights of the professional certificates are assigned as twice as important compare to other attributes, then the case that is found to be nearest-matched will be different. Thus, weightings on each attribute can influence the outcome.

Chapter 6 Conclusion

This chapter concludes the presentation of the thesis. This chapter is organised as follows. Section 1 discusses research results. Section 2 outlines research contributions and the chapter concludes with future research direction in Section 3.

6.1 Research Result

This research aims to investigate the feasibility of using CBR techniques to achieve knowledge sharing and reuse in the KMS. The research is conducted based on two broad areas: CBR and Ontology. We have developed a student admission enquiry system based on a framework that integrates CBR and agents. Agents are developed to perform the four phases of CBR cycles to allow the knowledge acquisition process to improve by allowing knowledge sharing and reuse.

In this research, we have demonstrated that CBR techniques can be applied to help knowledge acquisition process to resolve new problems. This is achieved by retrieving past solution and reapply the solution to new problems. New knowledge has been gained through the knowledge adaptation process, and new knowledge can be applied to solve new problems.

We have applied ontology to allow a well-defined definition for domain knowledge. This was achieved by providing conceptualised knowledge representation in terms of

standardised structured vocabularies. This way, ontology is function as meta-knowledge.

We have also applied the Semantic Web features to facilitate the implementation of agents in KMS. The application of agents allows the tasks of CBR to be performed independently and autonomously over the standardised Semantic Web platform.

6.2 Research Contribution

This research has been able to demonstrate that the application of CBR allows sharing and reuse of past experience in the KMS. In particular, the knowledge adaptation capability of CBR offers benefits to the development of KMS. The integration of ontology, the Semantic Web and agents in the development of KMS provides an opportunity and emerging perspective to facilitate the process of knowledge sharing and reuse in the KMS.

6.3 Future Research

In terms of knowledge sharing and reuse, this research has found that ontology is increasingly important and useful. According to Ding & Fensel (2001), "Ontology library systems are an import tool in grouping and re-organising ontologies for further reuse, integration, maintenance, mapping and versioning" (2001). However, few ontology maintenance aspects have been considered in literature. For example, limitation of possible changes in ontologies needs to be addressed. A term "Ontology

Versioning" (Klein and Fensel 2001) or "Ontology library"(Ding and Fensel 2001) has been introduced to maintain system that allows changed and unchanged ontology to be inter-operatable. This is based on the following rationale. Changes in knowledge domain, particularly shared conceptualisation and specification, might occur through the course of the development cycle. When this happens, ontology needs to be evolved too (Klein and Fensel 2001), otherwise it causes incompatible inter-operation in the system. Ontology versioning can help to reduce operability problems caused by the evolution of ontology. The versioning system will allow comparability issues to be taken into consideration when new knowledge is added to the system over time.

From the development perspective, future version of the prototype development can be developed using different storage model such as OWL. It has been proposed in the literature recently that ontology development should be kept as a back-end development, and it should be separated from a front-end web-based KMS. This approach is particularly useful in terms of ontology development, particularly if ontology versioning approach is used in future development.

Another interesting research area is the use of global or enterprise ontology (Uschold et al. 1995). Enterprise ontology is an approach of using several globally existing ontologies. It can be incorporated to current KMS so that well-defined domain of interests can be easily extended.

References

Aamodt, A. and Plaza, E. (1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *Artificial Intelligence Communications.* 7(1). pp.39-59

Aart, v. C., Pels, R., Caire, G. and Bergenti, F. (2002) Creating and Using Ontologies in Agent Communication, In *Proceedings of the 1st International Joing Conference on Autonomous Agents and Multi-Agent Systems.* 15-19 Jul. Bologna, Italy. <http://cis.otago.ac.nz/OAS2002/>

Abasola, J. M. and Gomez, M. (2000) MELISA. An ontology-based agent for information retrieval in medicine, In *Proceedings of Fourth Europoean Conference on Research and Advanced Technology for Digital Libraries.* 18-20 Sep. Lisbon, Portugal. <http://teste.bn.pt/ecdl2000/>

Aha, D. W. (1999) The AAAI-99 KM/CBR Workshop: Summary of Contributions, In *Proceedings of the Sixteenth National conference on Aritificial Intelligence Workshop on Exploring Synergies of Knowledge Management and Case-Baed Reasoning.* 18 - 22 Jul. Orland, Florida, USA. <http://www.aic.nrl.navy.mil/~aha/aaai99-kmcbrw/papers/>

Aitken, S. and Reid, S. (2000) Evaluation of an Ontology-Based Information Retrieval Tool, In *Proceedings of 14th European Conference on Artificial Intelligence (ECAI'00).* 20 - 25 Aug. Berlin, Germany. <http://delicias.dia.fi.upm.es/WORKSHOP/ECAI00/accepted-papers.html>

Alavi, M. and Leidner, E. D. (1999) Knowledge Management Systems: Issues, Challenges, and Benefits. *Communications of the Association for Information Systems.* 1 (7).

Altman, B. R., Chai, J. X., Garillo, W. M., Chen, O. R. and Abernethy, F. N. (1999) RiboWeb: An Ontology-Based Systems for Collaborative Molecular Biology. *IEEE Intelligent Systems.* pp.68-76.

Barclay, R. O. and Murray, P. C. (1997) *What is knowledge management.* Knowledge Management Associates. <http://www.media-access.com/whatis.html>. Accessed: 05 Mar., 2003

Barlow, L. (2002) *How Search Engines Work.* Monash Information Services. <http://www.monash.com/spidap.html>. Accessed: 27 Apr., 2002

Benjamins, V. R. and Fensel, D. (1998) The Ontological Engineering Initiative (KA). In Guarino, N. (Ed). *Formal Ontology in Information Systems* IOS Press.

Benjamins, V. R., Fensel, D. and Gomez-Perez, A. (1998) Knowledge Management through Ontologies, In *Proceedings of The 2nd International Conference on Practical Aspects of Knoweldge Management (PAKM98).* 29 - 30 Oct. Basel, Switzerland. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-13/>

Benson, S. and Standing, C. (2002) "Information Systems: A Business Approach". John Wiley & Sons Australia, Ltd.

Berners-Lee, T., Brickley, D., Connolly, D., Dean, M., Decker, S., Fensel, D., Fikes, R., Hayes, P., Heflin, J., Hendler, J., Lassila, O., McGuinness, D., Patel-Schneider, P. and Stein, L. A. (2001a) *DAML+OIL (December 2000).* <http://www.daml.org/2000/12/daml+oil-index.html>. Accessed: 23 Apr., 2002

Berners-Lee, T., Hendler, J. and Lassila, O. (2001b) *The Semantic Web.* Scientific American. <http://www.sciam.com/2001/0501issue/0501berners-lee.html>. Accessed: 1 Apr., 2002

Bhatt, D. G. (2000) Organizing knowledge in the knowledge development cycle. *Journal of Knowledge Management.* 4 (1). pp.15-26.

Bigus, P. J., Schlosnagle, A. D., Pilgrim, R. J., Mills III, N. W. and Diao, Y. (2002) ABLE: A toolkit for building multiagent autonomic systems. *IBM Systems Journal.* 41 (3). pp.350-371.

Boicu, M., Tecuci, G., Stanescu, B., Balan, G. C. and Popovici, E. (2001) Ontologies and the Knowledge Acquisition Bottleneck, In *Proceedings of International Joint Conference on Artificial Intelligence.* 4 - 10 Aug. Seatle, Washington. <http://www.ijcai.org/past/ijcai-01/>

Brule, F. J. and Blount, A. (1989) "Knowledge Acquisition". McGraw-Hill. New York.

Casebank Technologies. (2002) *SpotLight Reasoning Engine Software.* <http://www.casebank.com>. Accessed: 24 May., 2002

Churchman, C. W. (1972) "The Design of Inquiring Systems: Basic Concepts of Systems and Organisations". Bencis Books. New York.

Das, A., Wu, W. and McGuinness, D. L. (2001) Industrial Strength Ontology Management, In *Proceedings of International Semantic Web Working Symposium (SWWS) Workshop on Ontology and Ontology Maintenance.* 30 Jul. - 1 Aug. Stanford University, California, USA. <http://www.semanticweb.org/SWWS/program/full/>

Davies, J., Duke, A. and Stonkus, A. (2002) OntoShare: Using Ontology for Knowledge Sharing, In *Proceedings of The WWW2002 International Workshop on the Semantic Web.* 7 May. Hawaii. <http://sunsite.informatiik.rwth-aachen.de/Publications/CEUR-WS/Vol-55/>

Ding, Y. and Fensel, D. (2001) Ontology Library Systems: The key to successful Ontology Re-use, In *Proceedings of International Semantic Web Working Symposium (SWWS) on Ontology and Ontology Maintenance.* 30 Jul. - 1 Aug. Stanford University, California, USA. <http://www.semanticweb.org/SWWS/program/full/>

Dubitzky, W., Buchner, G. and Azuaje, J. (1999) Viewing Knowledge Management as a Case-Based Reasoning Application, In *Proceedings of the AAAI-99 Workshop on Exploring Synergies of Knowlege Management & Case-Based Reasoning.* Munich, Germany. <http://www.aic.nrl.navy.mil/~aha/aaai99-kmcbrw/papers/>

Everett, J. O., Bobrow, D. G., Stolle, R., Crouch, R., Paiva, V. D., Condoravdi, C., Berg, M. v. d. and Polanyi, L. (2002) Making ontologies work for resolving redundancies across documents. *Communication of the ACM.* 45(2). pp.55-60

Farquhar, A., Fikes, R., Pratt, W. and Rice, J. (1995) *Collaborative Ontology Construction for Information Integration.* <frp://ksl.stanford.edu/pub/KSL_Reports/>

Fensel, D., Harmelen, v. F., Klein, M. and Akkermans, H. (2002) On-To-Knowledge: Ontology-based Tools for Knowledge Management. *German Journal Kunstliche Intelligenz.* (Special Issus on Knowledge Management).

Franklin, S. and Graesser, A. (1996) Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, In *Proceedings of Third International Workshop on Agent Theories, Architectures, and Language.* 12-13 Aug. Budapest, Hungary. <http://www.dfki.uni-sb.de/~jpm/atal96.html>

Furnas, G. W., Laudauer, T. K., Gomez, L. M. and Dumais, S. T. (1987) The Vocabulary Problem in Human-System Communication. *Communications of the ACM.* 30 (11). pp.964-971.

Ganesh, D. B. (2000) Organizing knowledge in the knowedge development cycle. *Journal of Knowledge Management.* 4 (1). pp.15-26.

Gardenfors, P. (1995) Meanings as conceptual structures. In Machamer, K. P. (Ed). *Mindscapes: Philosophy, Sciences, and the Mind* Pittsburgh University Press. pp.61-86.

GITT. (2002) *About The GITT.* The Great Lakes Geriatric Interdisciplinary Team Training. <http://gitt.cwru.edu/>. Accessed: 30 Apr., 2002

Goldfarb, C. (1993) *SGML User's Group History.* Newswire. <http://xml.coverpages.org/sgmlhist0.html>. Accessed: 3 Jun., 2002

Goldfarb, C. F. (2001) "The XML Handbook". Prentice Hall.

Gomez-Perez, A. (1999) Applications of Ontologies and Problem-Solving Methods, In *Proceedings of Workshop at Thirteenth Biennial European Conference on Aritificial Intelligence.* 1 - 4 Oct. Berlin, Germany.

Gómez-Pérez, A. (1999a) Applications of Ontologies and Problem-Solving Methods, In *Proceedings of Workshop at Thirteenth Biennial European Conference on Aritificial Intelligence.* 1 - 4 Oct. Berlin, Germany.

Gómez-Pérez, A. (1999b) Ontological Engineering: A State of The Art. *Expert Update: Knowledge Based Systems and Applied Artificial Intelligence.* 2 (3). pp.33-44.

Gruber, R. T. (1993a) *A Translation Approach to Portable Ontology Specifications*. Knowledge Systems Laboratory, Computer Science Department, Stanford University. KSL 92-71.

Gruber, T. R. (Ed.) (1993b) *Toward Principles for the Design for Ontologies used for Knowledge Sharing.* Kluwer Academic Pulishers. Padova, Italy.

Hannula, M. and Pirttimaki, V. (2003) Business intelligence empirical study on the top 50 Finnish companies. *Journal of American Academy of Business.* 2 (2). pp.593-600.

Hayes-Roth, B. (1995) An Architecture for Adaptive Intelligent Systems. *Artificial Intelligence: Special Issue on Agents and Interactivity.* 72 (1-2). pp.329-365.

Heflin, J. and Hendler, J. (2000) Dynamic Ontologies on the Web, In *Proceedings of the seventeenth National Conference on Aritificial Intelligence.* 30 Jul. - 3 Aug. Austin, Texas. <http://www.aaai.org/Conferences/National/2000/aaai00.html>

Heinone, O., Hatonen, K. and Klemettinen, M. (1996) *WWW robots and serarch engines*. Helsinki University of Technology, Department of Computer Science. TKO-C79.

Hendler, J. (2001) Agents and the Semantic Web. *The IEEE Intelligent Systems Journal.* (Mar./Apr.).

Hildreth, P., Wright, P. and Kimble, C. (1999) Knowledge Management: Are we missing somthing, In *Proceedings of Proceedings of the 4th UK Academy for Information Systems Confercence.* 7 - 9 Apr. York, UK. <http://www.cs.york.ac.uk/mis/>

Hindriks, V. K., Boer, S. d. F., Hoek, v. d. W. and Meyer, C. J.-H. (1999) Agent Programming in 3APL. *Autonomos Agents and Multi-Agent Systems.* 2 (4). pp.357-401.

Hoffer, J. A., George, J. F. and Valacich, J. S. (1996) "Modern Systems Analysis and Design". The Benjamin/Cummings Publishing Company, Inc.

Holsapple, W. C. and Joshi, D. K. (2002) A Collaborative Approach to Ontology Design. *Communications of the ACM.* 45(2). pp.42-47

Hornby, S. A. (1995) "Oxford English Dictionary". Oxford University Press. Oxford, New York.

HPL. (2002) *Jena Semantic Web Toolkit.* Hewlett-Packard Company. <http://www.hpl.hp.com/semweb/jena.htm>. Accessed: 19 Jul., 2002

Jasper, R. and Uschold, M. (1999) A Framework for Understanding and Classifying Ontology Applications, In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence Workshop on Ontology.* 31 Jul. - 6 Aug. City Conterence Center, Stockholem, Sweden. <http://www.sdv.su.se/ijcai-99/>

Jin, Y., Decker, S. and Wiederhold, G. (2001) OntoWebber: Model-Driven Ontology-Based Web Site Management, In *Proceedings of International Semantic Web Working Symposium (SWWS) Workshop on Ontology and Ontology Maintenance.* 30 Jul. - 1 Aug. Stanford University, California, USA. <htttp://www.semanticweb.org/SWWS/Program/full/>

Joen, H., Petrie, C. and Cutkosky, R. M. (2000) JATLite: A Java Agent Infrastructure with Message Routing. *IEEE Internet Computing.* 4 (2). pp.87-96.

Klein, M. and Fensel, D. (2001) Ontology versioning on the Semantic Web, In *Proceedings of International Semantic Web Working Symposium.* 30 Jul. - 1 Aug. Stanford University, California, USA. <http://www.semanticweb.org/SWWS/>

Kolodner, J. (1983) Reconstructive momory, a computer model. *Congnitive Science.* 7. pp.281-328.

Kolodner, J. (1993) "Case-Based Reasoning". Morgan Kaufman Publisher, Inc. San Mateo, CA.

Konar, A. (2001) "Artificial Intelligence and Soft Computing: Behavioral and Congnitive Modeling of the Human Brain". CRC Press.

Koster, M. (1995) *Robots in the Web: threat or treat?* ConneXions. <http://www.robotstxt.org/wc/threat-or-treat.html>. Accessed:

Lassila, O. and Swick, R. R. (1999) *Resource Description Framework (RDF) Model and Syntax Specification*. World Wide Web Consortium. REC-rdf-syntax-19990222. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

Leonard-Barton, D. (1995) "Wellsprings of Knowledge: Building and Sustaining the Sources of Innovation". Harvard Business Schoold Press. Bostton, Mass.

Liebowitz, J. (2001) Knowledge management and its link to artificial intelligence. *Expert Systems with Applications.* 20. pp.1-6.

Luke, S. and Heflin, J. (2000) *SHOE 1.01.* SHOE Project. <http://www.cs.umd.edu/projects/plus/SHOE/spec.html>. Accessed: 09 Jun., 2003

Maes, P. (1995) Artificial Life Meets Entertainment: Life Like Autonomous Agents. *Communications of the ACM.* 38(11). pp.108-114

Magaldi, R. (1999) A Case-Based Approach to the Management and Development of Knowledge Assests, In *Proceedings of the International Conference on Case-Based Reasoning.* 27 - 30 Jul. Monastery Seeon, Munich, Germany. <http://www.iccbr.org/iccbr99/>

Malhotra, Y. (1998) *Knowledge Management, Knowledge Organizations & Knowledge Workers: A View from the Front Lines.* Maeil Business Newspaper. 19 Feb. <http://www.brint.com/interview/maeil.htm>

McCarthy, J. and Hayes, J. P. (1969) Some Philosophical Problems from the Standpoint of Artificial Intelligence. Eds, Meltzer, B. and Michie, D. (Ed). *Machine Intelligence 4*. pp.463-502.

Motta, E., Shum, S. B. and Domingue, J. (2000) Ontology-driven document enrichment: principles, tools and applications. *International Journal of Human-Computer Studies*. 52 (6). pp.1071-1109.

Moussavi, M. (1999) A Case-Based Approach to Knowledge Management, In *Proceedings of the International Conference on Case-Based Reasoning*. 27 - 30 Jul. Monastery Seeon, Munich, Germany. <http://www.iccbr.org/iccbr99/>

Norman, F. (1995) Ontology Revision, In *Proceedings of the 3rd International Conference on Conceptual Structures*. 14-18 Aug. University of California, Santa Cruz. <http://www.cse.unsw.edu.au/~ksg/Abstracts/Conf/ontology_rev.html>

Nour, P., Holz, H. and Maurer, F. (2000) Ontology-based Retrieval of Software Process Experiences, In *Proceedings of The 22nd International Conference on Software Engineering*. 6 Jun. Limerick, Ireland. <http://sern.ucalgary.ca/~maurer/icse2000ws/ICSE2000WS.html>

Noy, F. N. and McGuinness, L. D. (2001) *Ontology Development 101: A Guide to Creating Your First Ontology*. <http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-2001-0880.html>

Partridge, D. and Hussain, K., M. (1995) "Knowledge Based Information Systems". McGraw-Hill Book Company. London.

Polanyi, M. (1967) "The Tacit Dimension". Routledge and Kegan Paul. London.

Roberson, S. (2002) A tale of two knowledge-sharing systems. *Journal of Knowledge Management*. 6 (3). pp.295-308.

Russell, S. and Norvig, P. (1995) "Artificial Intelligence: A Modern Approach". Prentice-Hall.

Schank, R. C. (1982) "Dynamic memory: a theory of remining and learning in computers and people". Combridge Univeristy Press. New York.

Shoham, Y. (1993) Agent Oriented Programming. *Artificial Intelligence*. 60 (1). pp.51-92.

Smith, C. D., Cypher, A. and Spohrer, J. (1994) KidSim: Programming Agents Without a Programming Language. *Communications of the ACM*. 37(7). pp.55-67

Sowa, F. J. (2000) "Knowledge Representation: Logical, Philosophical, and Computational Foundations". Brooks/Cole Thomson Learning.

Stanford Medical Informatics. (2003) *Protégé-2000: User's Guide*. <http://protege.stanford.edu/>. Accessed: 12 Jun., 2003

Sun Microsystems Inc. (2003) *The Java Tutorial*. Sun Microsystems Inc. <http://java.sun.com/docs/books/tutorial/index.html>. Accessed: 09 Dec., 2003

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R. and Wenke, D. (2002) OntoEdit: Collaborative Ontology Development for the Semantic Web, In *Proceedings of The fisrt International Semantic Web Conference (ISWC 2002)*. 9 - 12 Jun. Sardinia, Italia. <http://www.aifb.uni-karlsruhe.de/WBS/ysu/>

The Apache XML Project. (2002-03) *Xerces Java 2*. The Apache Software Foundation. <http://xml.apache.org/>. Accessed: 19 Jul., 2002

Thierauf, R. J. (1999) "Knowledge Management Systems for Business". Quorum Books.

Tshi, E., Garner, J. B. and Staab, S. (2000) The role of Artificial Intelligence in Knowledge Management. *Knowledge-Based Systems.* 13 (5). pp.235-239.

Turban, E. and Aronson, J. E. (2001) "Decision Support Systems and Intelligence in knowledge management". Prentice Hall.

Uschold, M., King, M., Moralee, S. and Zorgios, Y. (1995) *The Enterprise Ontology*. AIAI, The University of Edinburgh.

Vance, D. M. (1997) Information, Knowledge and Wisdom: The Epistemic Hierarchy and Computer-Based Information System, In *Proceedings of the 1997 America's Conference on Information Systems.*

W3C. (1999a) *Resource Description Framework (RDF) Model and Syntax Specification*. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

W3C. (1999b) *HTML 4.01 Specification*. <URL: http://www.w3.org/TR/html4/>

W3C. (2000) *Extensible Markup Language (XML) 1.0 (Second Edition)*. <http://www.w3.org/TR/REC-xml>

W3C. (2002) *Requirements for a Web Ontology Language.* <http://www.w3.org/TR/2002/WD-webont-req-20020307/>. Accessed: 18 Apr., 2002

Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hubner, S. (2001) Ontology-based Integration of Information - A survey of Existing Approaches, In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence.* 4 - 10 Aug. Seatle, Washington, USA. <http://ijcai.org/past/ijcai-01/>

Walsham, G. (2001) Knowledge Management: The Benefits and Limitations of Computer Systems. *European Management Journal.* 19 (6). pp.559-608.

Watson, I. D. (1997) "Applying Case-Based Reasoning: Techniques for Enterprise Systems". Morgan Kaufmann. San Francisco, California.

Weick, E. K. (1995) "Sensemaking in organizations". Sage Publications. Thousand Oaks.

Whitley, E. A. (2002) Tacit and explicit knowledge: Conceptual confusion around the commodification of knowledge, In *Proceedings of BPRC Conference on Knowledge Management: Concepts and Controversies.* 10-11, Feb. University of Warwick, Coventry, UK.

Wiig, K. M. (1997) Knowledge management: an introduction and perspective. *Journal of Knowledge Management.* 1 (1). pp.6-14.

Wooldridge, J. M. and Jennings, R. N. (1995) Intelligent Agents: Theory and Practice. 10 (2). pp.115--152.

Zuniga, G. L. (2001) Ontology: Its Transformation From Philosophy to Information Systems, In *Proceedings of the International Conference on Formal Ontology in Information Systems.* 17 - 19 Oct. Ogunquit, Maine, USA. <http://www.fois.org/fois-2001/>

Zúniga, L. G. (2001) Ontology: Its Transformation From Philosophy to Information Systems, In *Proceedings of the International Conference on Formal Ontology in Information Systems.* 17 - 19 Oct. Ogunquit, Maine, USA. <http://www.fois.org/fois-2001/>

Appendix A – List of original cases

This appendix shows the original cases used in the prototype.

Original Case Number	Requirements for Admission to Information Systems						
	Course Applied	Qualification	Year	IT related	Professional Certificate	Occupation	Work Experience
1	Graduate Diploma	Diploma	3	Yes	Nil	IT related career	4
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. A list of original cases

Appendix B – Program codes

The order of files are in alphabetic ascending.

AgentA.java
AgentB.java
AgentC.java
Cases.java
cases.rdf
DbToCcases.java
DbToRcases.java
footer.html
header.html
index.html
ontology.rdf
Rdql.java
Rcases.rdf
Search2.jsp
SimpleTransform1.java
SimpleTransform2.java

```
/* C:\Tomcat\webapps\ROOT\proj\AgentA.java */

/*
 * AgentA
 *
 * 1. Converts user typed cases (DB data) to XML format (rcase.xml)
 * 2. The XML file (rcases.xml) to newly retrived RDF file
(rcases.rdf)
 *
 */

import java.io.IOException;
import javax.xml.transform.TransformerException;
import java.io.FileNotFoundException;
import java.io.IOException;

public class AgentA implements Runnable
{
   private long aWhile;
   DbToRcases dbr = new DbToRcases();
   SimpleTransform1 st1 = new SimpleTransform1();

   public AgentA(long delay)
   {
      aWhile = delay;
   }

   public void run()
   {
      try
      {
         while (true)
         {
            try
            {  // DB data to XML
               dbr.runDbToRcases();
               // XML to RDF
               st1.xsl_transform();
               Thread.sleep(aWhile);
            }
            catch (TransformerException te)
            {
               System.err.println(te);
            }
            catch (FileNotFoundException fnfe)
            {
               System.err.println(fnfe);
            }
            catch (IOException ioe)
            {
               System.err.println(ioe);
            }
         }
      }
      catch (InterruptedException e)
      {
         System.out.println(e);
      }
   }
```

```
   public static void main(String args[])
   {
      // Create a thread
      Thread agentA = new Thread(new AgentA(300000L));

      // Set thread as daemon
      agentA.setDaemon(true);

      System.out.println("Agent A");
      System.out.println("Press Enter to stop Agent A... \n");

      // Start the xmlToRdf thread
      agentA.start();

      try
      {
         System.in.read();
         System.out.println("Enter pressed... \n");
      }
      catch (IOException e)
      {
         System.err.println(e);
      }
      return;
   }
}
```

```
/* C:\Tomcat\webapps\ROOT\proj\AgentB.java */

/*
 * Coverts cases.xml to cases.rdf
 */

import java.io.IOException;
import javax.xml.transform.TransformerException;
import java.io.FileNotFoundException;
import java.io.IOException;

public class AgentB implements Runnable
{
   private long aWhile;
   SimpleTransform2 st2 = new SimpleTransform2();

   public AgentB(long delay)
   {
      aWhile = delay;
   }

   public void run()
   {
      try
      {
         while (true)
         {
            try
            {
               st2.xsl_transform();
               Thread.sleep(aWhile);
            }
            catch (TransformerException te)
            {
               System.err.println(te);
            }
            catch (FileNotFoundException fnfe)
            {
               System.err.println(fnfe);
            }
            catch (IOException ioe)
            {
               System.err.println(ioe);
            }
         }
      }
      catch (InterruptedException e)
      {
         System.out.println(e);
      }
   }

   public static void main(String args[])
   {
      // Create a thread
      Thread xmlToRdf = new Thread(new AgentB(300000L));

      // Set thread as daemon
      xmlToRdf.setDaemon(true);

      System.out.println("Agent B");
```

```
        System.out.println("Press Enter to stop Agent B... \n");

        // Start the xmlToRdf thread
        xmlToRdf.start();

        try
        {
           System.in.read();
           System.out.println("Enter pressed... \n");
        }
        catch (IOException e)
        {
           System.err.println(e);
        }
        System.out.println("Ending main()");
        return;
    }
}
```

```
/* C:\Tomcat\webapps\ROOT\proj\AgentC.java */

/*
 * AgentC
 *
 * 1. Does case adaptation
 * 2. Creates confirmed cases (ccases.rdf)
 *
 */

import java.io.IOException;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.sql.*;

public class AgentC implements Runnable
{
   private long aWhile;
   Rdql r = new Rdql();
   DbToCcases dtc = new DbToCcases();
   SimpleTransform2 st2 = new SimpleTransform2();

   public AgentC(long delay)
   {
      aWhile = delay;
   }

   public void run()
   {
      try
      {
         while (true)
         {
            try
            {
               // Case adaptation
               r.cleanupSQL();
               r.runRdql();

               // DB to ccases.rdf
               dtc.runDbToCcases();
               st2.xsl_transform();
               Thread.sleep(aWhile);
            }
            catch (Exception e)
            {
               System.err.println(e);
            }
         }
      }
      catch (Exception e)
      {
         System.out.println(e);
      }
   }

   public static void main(String args[])
   {
      // Create a thread
      Thread rdqlAgent = new Thread(new AgentC(300000L));
```

```
        // Set thread as daemon
        rdqlAgent.setDaemon(true);

        System.out.println("Agent C");
        System.out.println("Press Enter to stop Agent C... \n");

        // Start the xmlToRdf thread
        rdqlAgent.start();

        try
        {
           System.in.read();
           System.out.println("ENTER pressed... \n");
        }
        catch (IOException e)
        {
           System.err.println(e);
        }
        return;
     }
}
```

```
/* C:\Tomcat\webapps\ROOT\WEB-INF\classes\proj\Cases.java */

/*
 *  Cases class is consists of coustructions and setter
 *  & getter methods such as course, qualificaton, year,
 *  it, certificate, occupation and experience.
 *
 */
package proj;

public class Cases
{
    private String course = "";
    private String qualification = "";
    private int    year = 0;
    private String it = "";
    private String certificate = "";
    private String occupation = "";
    private int    experience = 0;

   /* Cases constructor */
    public Cases()
    {}

   /* Cases constructor */
    public Cases( String c,
         String q,
         int y,
         String i,
         String ce,
         String o,
         int e)
    {
        course = c;
        qualification = q;
        year = y;
        it = i;
        certificate = ce;
        occupation = o;
        experience = e;
    }

    /** Getter for property course.
     * @return Value of property course.
     *
     */
    public String getCourse()
    {
        return course;
    }

    /** Setter for property course.
     * @param course New value of property course.
     *
     */
    public void setCourse(String c)
    {
        course = c;
    }

    /** Getter for property qualification.
```

```
     * @return Value of property qualification.
     *
     */
    public String getQualification()
    {
        return qualification;
    }

    /** Setter for property qualification.
     * @param qualification New value of property qualification.
     *
     */
    public void setQualification(String q)
    {
        qualification = q;
    }

    /** Getter for property year.
     * @return Value of property year.
     *
     */
    public int getYear()
    {
        return year;
    }

    /** Setter for property year.
     * @param year New value of property year.
     *
     */
    public void setYear(int y)
    {
        year = y;
    }

    /** Getter for property IT.
     * @return Value of property IT.
     *
     */
    public String getIt()
    {
        return it;
    }

    /** Setter for property IT.
     * @param IT New value of property IT.
     *
     */
    public void setIt(String i)
    {
        it = i;
    }

    /** Getter for property certificate.
     * @return Value of property IT.
     *
     */
    public String getCertificate()
    {
        return certificate;
    }
```

```
    /** Setter for property certificate.
     * @param cretificate New value of property certificate.
     *
     */
    public void setCertificate(String ce)
    {
        certificate = ce;
    }

    /** Getter for property occupation.
     * @return Value of property occupation.
     *
     */
    public String getOccupation()
    {
        return occupation;
    }

    /** Setter for property occupation.
     * @param occupation New value of property occupation.
     *
     */
    public void setOccupation(String o)
    {
        occupation = o;
    }

    /** Getter for property experience.
     * @return Value of property experience.
     *
     */
    public int getExperience()
    {
        return experience;
    }

    /** Setter for property experience.
     * @param experience New value of property experience.
     *
     */
    public void setExperience(int e)
    {
        experience = e;
    }
}
```

```
<!-- ccases.rdf ->

<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:admission="http://dsl08.dsl.uow.edu.au/is/admission-rdf/1.0#">
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13017">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>No</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>Non-IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13018">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Professional +
Internet</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13019">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems
Engineer</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13020">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems
Engineer</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>4</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13021">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems
Engineer</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>10</admission:experience>
</rdf:Description>
```

```
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13022">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer +
Internet</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13023">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer +
Internet</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>6</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13024">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13025">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>10</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13026">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13027">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
```

```
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13028">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>5</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13029">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>6</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/ccase/13030">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>10</admission:experience>
</rdf:Description>
</rdf:RDF>
```

```
/* C:\Tomcat\webapps\ROOT\proj\DbToCcases.java */

/*
 * This class apadats cases based on requirements or conditions
 * and then savedconverts to a xml file called rcases.xml.
 *
 */

import java.sql.*;
import java.io.*;

public class DbToCcases
{
   public void runDbToCcases()
   {
      String url = "jdbc:mysql://localhost:3306/test";
      String user = "sk33";
      String pass = "o151fs";

      try
      {  // Load the MySQL driver
         // JDBC 3.0 driver
         Class.forName("com.mysql.jdbc.Driver").newInstance();
         // JDBC 2.x driver
         // Class.forName("org.gjt.mm.mysql.Driver");

      }  // If the driver not loaded,
         // an exception would have been thrown
      catch (Exception e)
      {
         System.err.println("Unable to load driver.");
         e.printStackTrace();
         return;
      }
      try
      {  // Setup a connection to the database
         Connection conn =
         DriverManager.getConnection(url, user, pass);
         Statement stmt = conn.createStatement();

         String SQLText =
         " SELECT SQL_CACHE * FROM ccases " +
         " GROUP BY course, qualification, year, it, " +
         " certificate, occupation, experience";

         // Execute the query statement and
         // capture the results
         ResultSet rs = stmt.executeQuery(SQLText);

         // Display all columns and rows from the result set
         DisplayResults(rs);

         // Freeup any resources associated
         // with the statement and result set
         rs.close();
         stmt.close();
         conn.close();
      }  // if the connection to the DB failed,
          // an exception would have been thrown
      catch (SQLException e)
      {
```

```
      System.err.println("SQLException: " +
      e.getMessage());
      System.err.println("SQLState    : " +
      e.getSQLState());
      System.err.println("VendorError : " +
      e.getErrorCode());
   }
   catch (Exception ex)
   {
      ex.printStackTrace();
   }

}

synchronized private static void DisplayResults(
   ResultSet results) throws SQLException, IOException
{
   int i;
     String xmlString = "";
     String str1 = "";
     String str2 = "";
     String str3 = "";
     String str4 = "";
     String str5 = "";
     String str6 = "";
     String str7 = "";
     String str8 = "";
     String str9 = "";

     // Directory name
     String dirname = "c:\\Tomcat\\webapps\\ROOT\\proj";
     String filename = "ccases.xml";

     File aFile = new File(dirname, filename);
     // Create outputstream
     DataOutputStream out =
         new DataOutputStream(
             new BufferedOutputStream(
                 new FileOutputStream(aFile, false)));

     // Get the ResultSetMetaData
     ResultSetMetaData rsmd = results.getMetaData();

     // Get the number of columns in the result set
     int numCols = rsmd.getColumnCount();

     // Set default xml titles
     out.writeBytes("<?xml version=\"1.0\" ?>\n\n");
     out.writeBytes("<admission>\n");

   // Display data, fetching until end of the result set
     boolean more = results.next();
     while (more)
     {
     // loop through each column,
     // getting the column data and displaying
     for (i = 1; i <= numCols; i++)
      {
          if (i == 1)
          {
            out.writeBytes("   <case id=\"" +
```

```
                results.getString(i) + "\">\n");
              }
              if (i == 2)
              {
                out.writeBytes("   <course>" +
                results.getString(i) + "</course>\n");
              }
              if (i == 3)
              {
                out.writeBytes("   <qualification>" +
                results.getString(i) + </qualification>\n");
              }
              if (i == 4)
              {
                out.writeBytes("   <year>" +
                results.getString(i) + "</year>\n");
              }
              if (i == 5)
              {
                out.writeBytes("   <it>" +
                results.getString(i) + "</it>\n");
              }
              if (i == 6)
              {
                out.writeBytes("   <certificate>" +
                results.getString(i) + "</certificate>\n");
              }
              if (i == 7)
              {
                out.writeBytes("   <occupation>" +
                results.getString(i) + "</occupation>\n");
              }
              if (i == 8)
              {
                out.writeBytes("   <experience>" +
                results.getString(i) + "</experience>\n");
              }
              if (i == 9)
              {
                out.writeBytes("   <published>" +
                results.getString(i) + "</published>\n");
            }
        }
        out.writeBytes("   </case>\n");
        // Fetch the next result set row
        more = results.next();
        }
        out.writeBytes("</admission>");
        out.close();
    System.out.println("The result is in ccases.xml!");
    }
}
```

```
/* C:\Tomcat\webapps\ROOT\proj\DbToRcases.java */

/*
 * This class retrieves all inputed cases from DB
 * and then converts to a xml file called rcases.xml.
 */

import java.sql.*;
import java.io.*;

public class DbToRcases
{
    public void runDbToRcases()
   {
      String url = "jdbc:mysql://localhost:3306/test";
      String user = "sk33";
      String pass = "o151fs";

      try
      {  // Load the MySQL driver
         // JDBC 3.0 driver
         Class.forName("com.mysql.jdbc.Driver").newInstance();
         // JDBC 2.x driver
         // Class.forName("org.gjt.mm.mysql.Driver");

      }  // If the driver not loaded,
         // an exception would have been thrown
      catch (Exception e)
      {
         System.err.println("Unable to load driver.");
         e.printStackTrace();
         return;
      }
      try
      {  // Setup a connection to the database
         Connection conn =
         DriverManager.getConnection(url, user, pass);
         Statement stmt = conn.createStatement();

         String SQLText =
            " SELECT SQL_CACHE * FROM icases " +
            " group by course, qualification, year, it, " +
            " certificate, occupation, experience";

         // Execute the query statement and
         // capture the results
         ResultSet rs = stmt.executeQuery(SQLText);

         // Display all columns and rows from the result set
         DisplayResults(rs);

         // Freeup any resources associated with
         // the statement and result set
         rs.close();
         stmt.close();
         conn.close();
      }  // If the connection to the DB failed,
         // an exception would have been thrown
      catch (SQLException e)
      {
         System.err.println("SQLException: " +
```

```
      e.getMessage());
      System.err.println("SQLState    : " +
      e.getSQLState());
      System.err.println("VendorError : " +
      e.getErrorCode());
   }
   catch (Exception ex)
   {
      ex.printStackTrace();
   }

}

synchronized private static void DisplayResults(
   ResultSet results) throws SQLException, IOException
{
  int i;
  String xmlString = "";
  String str1 = "";
  String str2 = "";
  String str3 = "";
  String str4 = "";
  String str5 = "";
  String str6 = "";
  String str7 = "";
  String str8 = "";
  String str9 = "";

  // Directory name
  String dirname = "c:\\Tomcat\\webapps\\ROOT\\proj";
  String filename = "rcases.xml";

  File aFile = new File(dirname, filename);
  // Create outputstream
  DataOutputStream out =
      new DataOutputStream(
          new BufferedOutputStream(
              new FileOutputStream(aFile, false)));

  // Get the ResultSetMetaData
  ResultSetMetaData rsmd = results.getMetaData();

  // Get the number of columns in the result set
  int numCols = rsmd.getColumnCount();

  // Set default xml titles
  out.writeBytes("<?xml version=\"1.0\" ?>\n\n");
  out.writeBytes("<admission>\n");

// Display data, fetching until end of the result set
  boolean more = results.next();
  while (more)
  {
// Loop through each column,
// Getting the column data and displaying
for (i = 1; i <= numCols; i++)
{
       if (i == 1)
       {
         out.writeBytes("   <case id=\"" +
         results.getString(i) + "\">\n");
```

```
            }
            if (i == 2)
            {
              out.writeBytes("   <course>" +
              results.getString(i) + "</course>\n");
            }
            if (i == 3)
            {
              out.writeBytes("   <qualification>" +
              results.getString(i) + "</qualification>\n");
            }
            if (i == 4)
            {
              out.writeBytes("   <year>" +
              results.getString(i) + "</year>\n");
            }
            if (i == 5)
            {
              out.writeBytes("   <it>" +
              results.getString(i) + "</it>\n");
            }
            if (i == 6)
            {
              out.writeBytes("   <certificate>" +
              results.getString(i) + "</certificate>\n");
            }
            if (i == 7)
            {
              out.writeBytes("   <occupation>" +
              results.getString(i) + "</occupation>\n");
            }
            if (i == 8)
            {
              out.writeBytes("   <experience>" +
              results.getString(i) + "</experience>\n");
            }
            if (i == 9)
            {
              out.writeBytes("   <published>" +
              results.getString(i) + "</published>\n");
            }
         }
            out.writeBytes("   </case>\n");
         // Fetch the next result set row
            more = results.next();
      }
      out.writeBytes("</admission>");
      out.close();
      System.out.println("The result is in rcases.xml!");
      }
}
```

```
<!—- footer.html -- >

<p>
<center>
<table border="0" width="100%">
<tr>
<td bgcolor="#000000"><font color="#ffffff">&copy; 2002 - 2003 Kang,
S. Information Systems. University of Wollongong. All rights
reserved</font></td>
</tr>
</table>
</center>
</body>
</html>
```

```
<!-- C:\Tomcat\webapps\ROOT\proj\header.html -->

<p>
<center>
<table border="0" width="100%">
<tr>
   <td bgcolor="#000000">
   <font color="#ffffff" size="1">
   &copy; 2002-03 Kang, S. Information Systems.
   University of Wollongong.
   All rights reserved
   </font>
   </td>
</tr>
</table>
</center>
</body>
</html>
```

```
<!-- C:\Tomcat\webapps\ROOT\proj\index.html -->

<!--

Last Updated : 18/08/2003
Developed by : Kang, S.

-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title></title>
</head>

<body marginwidth="0" marginheight="0" topmargin="0" leftmargin="0">

<h2 align="center">Project Prototype</h2>

<form action="search2.jsp" method="post">
<center>
<table width="100%" border="0" cellpadding="3" cellspacing="3">
<tr>
   <td class="bg_color" width="150">*Admission to</td>
   <td width="650">
<input type="text" name="course">
<i>eg) Graduate Diploma</i>
</td>
</tr>
<tr>
   <td colspan="2" class="bg_color">Personal Details</td>
</tr>
<tr>
   <td align="right" class="bg_color" width="150">Your Name</td>
   <td width="650">
First Name <input type="text" name="fname">
Last Name <input type="text" name="lname">
</td>
</tr>
<tr>
   <td align="right" class="bg_color" width="150">
Your Contact E-mail Address
</td>
   <td><input type="text" name="email"></td>
</tr>

<tr>
<td colspan="2" class="bg_color">Academic Qualification</td>
</tr>
<tr>
   <td align="right" class="bg_color" width="150">
*Title of Yours
</td>
   <td width="650"><input type="text" name="qual">
<i>eg) Bachelor</i>
</td>
</tr>
<tr>
   <td class="bg_color" align="right">*Year(s)</td>
   <td>
<input type="text" name="qyear" size="1" maxlength="1">
```

```
     from Non-IT related course(s)
<input type="radio" name="itco" value="No">
   IT/IS/CS related course(s)
<input type="radio" name="itco" value="Yes" checked>
   </td>
</tr>
<tr>
   <td align="right" class="bg_color" width="150">
Name of Your University
</td>
   <td width="650">
<input type="text" name="uniname">
<i>eg) University of Wollongong</i>
</td>
</tr>
<tr>
   <td align="right" class="bg_color" width="150">
URL of Your University
</td>
   <td width="650">
<input type="text" name="uniname">
<i>eg) http://www.uow.edu.au</i>
</td>
</tr>
<tr>
   <td align="right" class="bg_color" width="150">
Country (origin)
</td>
   <td width="650">
<input type="text" name="country">
<i>eg) Australia</i>
</td>
</tr>
<tr>
<td colspan="2" class="bg_color">
Professional Certificate  (if there is any)
</td>
</tr>
<tr>
<td align="right" class="bg_color" width="150">
*Title of Yours
</td>
<td width="650">
<input type="text" name="cert">
<i>eg) MCSE</i>
</td>
</tr>
<tr>
<td colspan="2" class="bg_color">
Work Experience (if there is any)
</td>
</tr>
<tr>
<td align="right" class="bg_color" width="150">
*Your Occupation
</td>
<td width="650">
<select name="occu">
<option value="Nil">Nil</option>
<option value="IT related career">IT related career</option>
<option value="Non-IT related career">
```

```
Non-IT related career</option>
    </select>
</td>
</tr>
<tr>
   <td class="bg_color" align="right">*Year(s)</td>
   <td>
<input type="text" name="oyear" size="1" maxlength="2">
   </td>
</tr>
<tr>
<td> </td>
<td>
<input type="submit" name="mode" value="Submit">
<input type="reset">
</td>
</tr>
<tr>
<td>
<p>* <a href="http://dsl08.dsl.uow.edu.au/proj/phpboard/">
Project Status</a>
</td>
</tr>
</table>
</center>
</form>

<p>
<center>
<table border="0" width="100%">
<tr>
<td bgcolor="#000000">
<font color="#ffffff" size="2">&copy; 2002 Kang, S.
Information Systems. University of Wollongong.
All rights reserved</font>
</td>
</tr>
</table>
</center>
</body>
</html>
```

```
<!-- ocases.rdf →

<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF xmlns:admission="http://dsl08.dsl.uow.edu.au/is/admission-
rdf/1.0#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/1">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/2">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/3">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/4">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems
Engineer</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/5">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/6">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
```

```
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>1</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/7">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems
Engineer</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/8">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified
Professional</admission:certificate>
<admission:occupation>Non-IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/9">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>1</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/10">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/11">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems
Engineer</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
```

```
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/12">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified
Professional+Internet</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/13">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems
Engineer</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>5</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/14">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/15">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems
Engineer</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
</rdf:RDF>
```

```
<!-- ontolog.rdf -->

<?xml version="1.0" ?>
<rdf:RDF
       xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:AO  = "http://dsl08.dsl.uow.edu.au/is/admission-
rdf/AO#">

<!-- ontologies for information systems department -->
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/InformationSystem
s">
<AO:School>School of Economics and Information Systems</AO:School>
<AO:AboutUs>Information Systems is an important multidisciplinary
field concerned with the complex integration of information and
communication technology, data stores, people and procedures to
enable organisations to function efficiently and to capitalise on
their information resources for strategic advantage. The provision of
quality tertiary education in conjunction with the conduct of
leading-edge research in the area of Information Systems is of vital
concern to the future of all social organisations, both in business
and in not-for-profit institutions of all types and sizes.
      The Discipline of Information Systems in the School of
Economics and Information Systems within the Faculty of Commerce at
the University of Wollongong was formed in 1989 and is recognised an
effective and productive academic unit which serves the institution,
the region and the nation. The Discipline provides quality, up-to-
date undergraduate and postgraduate education in information systems
development and management. Two active research laboratories (the
Decision Systems Laboratory and the Activity Theory Usability
Laboratory) continue to be productive and successful in obtaining
external grants, supervising research students and in maintaining
active links with industry and international partners. The Australian
Journal of Information Systems is edited and produces within the
Discipline of Information Systems.
</AO:AboutUs>
<AO:OurStaff>
http://www.uow.edu.au/commerce/buss/staff.html
</AO:OurStaff>
<AO:OurResearch>
http://www.uow.edu.au/commerce/buss/research.html
</AO:OurResearch>
<AO:Activities>
http://www.uow.edu.au/commerce/buss/activities.html
</AO:Activities>
<AO:CurrentStudents>
http://www.uow.edu.au/commerce/buss/currstud.html
</AO:CurrentStudents>
<AO:NewStudents>
http://www.uow.edu.au/commerce/buss/newstud.html
</AO:NewStudents>
<AO:URL>
http://www.uow.edu.au/commerce/buss/
</AO:URL>
</rdf:Description>

<!-- ontologies for courses -->
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Courses/GraduateD
iploma">
<AO:FN>Graduate Diploma</AO:FN>
```

```
<AO:N rdf:parseType="Resource">
      <AO:SN>Grad Dipl</AO:SN>
</AO:N>
<AO:URL>http://www.uow.edu.au/commerce/buss/graddip.htm</AO:URL>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Courses/MasterInf
ormationSystems">
<AO:FN>Master Information Systems</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>MIS</AO:SN>
</AO:N>
<AO:URL>http://www.uow.edu.au/commerce/buss/moc.htm</AO:URL>
</rdf:Description>

<!-- ontologies for qualifications -->
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Qualification/Dip
loma">
<AO:FN>Diploma</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>Dipl</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Qualification/Bac
helor">
<AO:FN>Bachelor</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>Bach</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Qualification/Hon
ours">
<AO:FN>Bachelor of Honours</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>Honours Bachelor</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Qualification/Mas
ter">
<AO:FN>Master</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>Mast</AO:SN>
</AO:N>
</rdf:Description>

<!-- ontologies for professional certificates -->
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/SCJP
">
<AO:FN>Sun Certified Programmer for Java 2 Platform</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>SCJP</AO:SN>
</AO:N>
```

```
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/SCJD
">
<AO:FN>Sun Certified Developer for Java 2 Platform</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>SCJD</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/MCP"
>
<AO:FN>Microsoft Certified Professional</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>MCP</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/MCSA
">
<AO:FN>Microsoft Certified Systems Administrator</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>MCSA</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/MCSE
">
<AO:FN>Microsoft Certified Systems Engineer</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>MCSE</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/MCP+
I">
<AO:FN>Microsoft Certified Professional + Internet</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>MCP+I</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/MCSE
+I">
<AO:FN>Microsoft Certified Systems Engineer + Internet</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>MCSE+I</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/OCP"
>
<AO:FN>Oracle Certified Professional</AO:FN>
<AO:N rdf:parseType="Resource">
```

```
      <AO:SN>OCP</AO:SN>
</AO:N>
</rdf:Description>

<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/CNE"
>
<AO:FN>Certified Novell Engineer</AO:FN>
<AO:N rdf:parseType="Resource">
      <AO:SN>CNE</AO:SN>
</AO:N>
</rdf:Description>

<!-- a list of accepted professional certificates by IS at Year 2002
-->
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/Certificates/Cert
ified/2002">
<AO:APC>
      <rdf:Bag>
            <rdf:li
rdf:resource="http://dsl08.dsl.uow.edu.au/is/admission/AcceptedProfes
sionalCertificates/MCP+I" />
            <rdf:li
rdf:resource="http://dsl08.dsl.uow.edu.au/is/admission/AcceptedProfes
sionalCertificates/MCSE" />
            <rdf:li
rdf:resource="http://dsl08.dsl.uow.edu.au/is/admission/AcceptedProfes
sionalCertificates/MCSE+I" />
            <rdf:li
rdf:resource="http://dsl08.dsl.uow.edu.au/is/admission/AcceptedProfes
sionalCertificates/SCJP" />
      </rdf:Bag>
</AO:APC>
</rdf:Description>

</rdf:RDF>
```

```
/* C:\Tomcat\webapps\ROOT\proj\Rdql.java */

/*
 * RDQL (RDF data Query Language) of
 * Jena Semantic Web Toolkit (HPL 2002) to find the best or
 * the closest matched case in the knowledge repository
 *
 */

import com.hp.hpl.mesa.rdf.jena.model.*;
import com.hp.hpl.mesa.rdf.jena.mem.*;
import com.hp.hpl.mesa.rdf.jena.common.*;

import com.hp.hpl.jena.rdf.query.*;

import java.util.*;
import java.io.*;
import java.sql.*;
import java.sql.Statement;

public class Rdql
{
   static protected String COUR = "";  // Course
   static protected String QUAL = "";  // Qualification
   static protected int YEAR;    // A number of years
   static protected String ITCO = "";  // IT related
   static protected String CERT = "";  // Professional certificate
   static protected String OCCU = "";  // Occupation
   static protected int EXPE;    // Work experience
   static protected String RESP = "";  // Response

   // Constructor
    public void Rdql()
    {}

   // Constructor
   public void Rdql( String COUR, String QUAL, int YEAR,
      String ITCO, String CERT, String OCCU, int EXPE,
      String RESP)
   {
      this.COUR = COUR;
      this.QUAL = QUAL;
      this.YEAR = YEAR;
         this.ITCO = ITCO;
      this.CERT = CERT;
      this.OCCU = OCCU;
      this.EXPE = EXPE;
      this.RESP = RESP;
   }

/*
 * Adaptation Rule
 *
 * Retrieve attributes of the applicants'
 * that are stored in RDF model using RDQL.
 *
 * If all basic academic requirements for admission
 * then adapts the case
 * otherwise discards it
 */
```

```
synchronized public void runRdql()
 {
     Rdql rdql = new Rdql();

   try
   {
      Model model = new ModelMem();
      model.read(new
      FileReader("c:\\Tomcat\\webapps\\ROOT\\proj\\rcases.rdf"),
         "http://dsl08.dsl.uow.edu.au/is/admission-rdf/1.0#",
          "RDF/XML");

      String queryString =
      "SELECT ?x, ?course, ?qual, ?year, ?itco, ?certi, " +
      " ?occu, ?expe " +
      " WHERE (?x, <admission:course>, ?course), " +
      " (?x, <admission:qualification>, ?qual) " +
      " (?x, <admission:year>, ?year)" +
      " (?x, <admission:it>, ?itco)" +
      " (?x, <admission:certificate>, ?certi) " +
      " (?x, <admission:occupation>, ?occu) " +
      " (?x, <admission:experience>, ?expe) " +
      " USING admission for " +
      " <http://dsl08.dsl.uow.edu.au/is/admission-rdf/1.0#>";

      Query query = new Query(queryString);
      query.setSource(model);
      QueryExecution qe = new QueryEngine(query);

      QueryResults results = qe.exec();
      for (Iterator iter = results; iter.hasNext(); )
      {
         ResultBinding res = (ResultBinding)iter.next();
         Object x = res.get("x");
         Object cour = res.get("course");
         Object qual = res.get("qual");
         Object year = res.get("year");
         Object itco = res.get("itco");
         Object cert = res.get("certi");
         Object occu = res.get("occu");
         Object expe = res.get("expe");

         char qflag = ' ';
         char yflag = ' ';
         char iflag = ' ';
         char cflag = ' ';
         char oflag = ' ';
         char eflag = ' ';

         COUR = cour.toString();
         QUAL = qual.toString();
         YEAR = Integer.parseInt(year.toString());
         ITCO = itco.toString();
         CERT = cert.toString();
         OCCU = occu.toString();
         EXPE = Integer.parseInt(expe.toString());

          // 3 years diploma +
          // releant work experiences of at least 2 years
         if ( (COUR.equalsIgnoreCase("graduate diploma")) &&
              (QUAL.equalsIgnoreCase("diploma")) &&
```

```
      (YEAR >= 3) &&
      (OCCU.equalsIgnoreCase("IT related career")) &&
      (EXPE >= 2) )
{
    qflag = 'y';
    yflag = 'y';
    iflag = '-';
    cflag = '-';
    oflag = 'y';
    eflag = 'y';
    RESP = "APPROVED";
}
// 3 years diploma + professional certificates
// such as JAVA, MCSE, MCSD
if ((COUR.equalsIgnoreCase("graduate diploma")) &&
    (QUAL.equalsIgnoreCase("diploma") &&
    (YEAR >= 3) &&
    (CERT.equalsIgnoreCase("java") ||
     CERT.equalsIgnoreCase("mcse") ||
     CERT.equalsIgnoreCase("mcsd")) ))
{
     qflag = 'y';
     yflag = 'y';
     iflag = '-';
     cflag = 'y';
     oflag = '-';
     eflag = '-';
     RESP  = "APPROVED";
}
// other 3 conditions has to be added here.
if ((COUR.equalsIgnoreCase("graduate diploma")) &&
    (QUAL.equalsIgnoreCase("bachelor")) &&
    (YEAR >= 3) &&
    (ITCO.equalsIgnoreCase("no")))
{
   qflag = 'y';
   yflag = 'y';
   iflag = 'n';
   cflag = '-';
   oflag = '-';
   eflag = '-';
   RESP  = "APPROVED";
}
// 2 years bachelor/diploma will NOT be accepted to
// postgraduate courses.
if ((COUR.equalsIgnoreCase("graduate diploma")) &&
    (QUAL.equalsIgnoreCase("diploma") ||
    (QUAL.equalsIgnoreCase("bachelor"))) &&
    (YEAR <= 2))
{
     qflag = 'y';
     yflag = 'n';
     cflag = '-';
     oflag = '-';
     eflag = '-';
     RESP  = "REJECTED";
}
// admission to MIS-special cases
// 3 years diploma majoring in IS,IT,CS +
//relevant work experience of more than 2 years
if ((COUR.equalsIgnoreCase("mis")) &&
```

```
            (QUAL.equalsIgnoreCase("bachelor")) &&
            (YEAR >= 3) &&
            (ITCO.equalsIgnoreCase("yes")) &&
            (CERT.equalsIgnoreCase("java") ||
            CERT.equalsIgnoreCase("mcse") ||
            CERT.equalsIgnoreCase("mcsd")) )
        {
            qflag = 'y';
            yflag = 'y';
            iflag = 'y';
            cflag = 'y';
            oflag = '-';
            eflag = '-';
            RESP  = "APPROVED";
        }
        if (RESP.equalsIgnoreCase("approved"))
        {
           RESP = "APPROVED";
           setDB(COUR, QUAL, YEAR, ITCO, CERT,
           OCCU, EXPE, RESP, qflag, yflag,
           iflag, cflag, oflag, eflag);
        }
        RESP = "REJECTED";
     }
        results.close();
   } catch (Exception ex) {
     System.err.println("Exception: "+ex);
     ex.printStackTrace(System.err);
   }
}

// Method to store all confired RDF cases to DB
public static void setDB(String CO,
    String QU,
    int YE,
    String IT,
    String CE,
    String OC,
    int EX,
    String RE,
    char qflag,
    char yflag,
          char iflag,
    char cflag,
    char oflag,
    char eflag )
{
  String url = "jdbc:mysql://localhost:3306/test";
  String user = "sk33";
  String pass = "o151fs";
  Connection conn = null;
  Statement stmt = null;

   try
   {  // Load the MySQL driver
      // JDBC 3.0 driver
      Class.forName("com.mysql.jdbc.Driver").newInstance();
      // JDBC 2.x driver
      //Class.forName("org.gjt.mm.mysql.Driver").newInstance();

   }  // if the driver not loaded,
```

```
      //an exception would have been thrown
   catch (Exception e)
   {
      System.err.println("Unable to load driver.");
      e.printStackTrace();
      return;
   }
   try
   {  // setup a connection to the database
      conn = DriverManager.getConnection(url, user, pass);
      stmt = conn.createStatement();

      String SQLText =
      " INSERT INTO ccases (course, qualification, " +
      " year, it, certificate, occupation, " +
      " experience, response, published, " +
      " qflag, yflag, iflag, cflag, oflag, eflag) " +
      " VALUES(" + " '" + CO + "', '" + QU + "', " + YE +
      ", '" + IT + "', '" + CE + "', '" + OC + "', " + EX +
      ", '" + RE + "', now(), '" + qflag + "', '" + yflag +
      "', '" + iflag + "', '" + cflag + "', '" + oflag +
      "', '" + eflag + "' )";

      // Execute the query statement
      // and capture the results
      ResultSet rs = stmt.executeQuery(SQLText);
   }  // If the connection to the DB failed,
      // an exception would have been thrown
   catch (SQLException e)
   {
      e.printStackTrace();
   }
   finally {
      if (conn != null)
      {
         try
         {
            // Freeup any resources associated
            // with the statement and result set
            stmt.close();
            conn.close();
         }
         catch (SQLException e)
         {
            e.printStackTrace();
         }
      }
   }
}

synchronized public void cleanupSQL()
{
  String url = "jdbc:mysql://localhost:3306/test";
  String user = "sk33";
  String pass = "o151fs";
  Connection conn = null;
  Statement stmt = null;

   try
   {  // load the MySQL driver
      // JDBC 3.0 driver
```

```
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            // JDBC 2.x driver
            // Class.forName("org.gjt.mm.mysql.Driver").newInstance();

         }  // if the driver not loaded,
            // an exception would have been thrown
         catch (Exception e)
         {
            System.err.println("Unable to load driver.");
            e.printStackTrace();
            return;
         }
         try
         {  // setup a connection to the database
            conn = DriverManager.getConnection(url, user, pass);
            stmt = conn.createStatement();

            String SQLText = "DELETE FROM ccases";
            // execute the query statement
            // and capture the results
            ResultSet rs = stmt.executeQuery(SQLText);

         }  // if the connection to the DB failed,
            // an exception would have been thrown
         catch (SQLException e)
         {
            e.printStackTrace();
         }
         finally {
            try
            {
               // Freeup any resources associated
               // with the statement and result set
               stmt.close();
               conn.close();
            }
            catch (SQLException e)
            {
               e.printStackTrace();
            }
         }
      }
   }
```

```
<!-- rcases.rdf -->

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:admission="http://dsl08.dsl.uow.edu.au/is/admission-rdf/1.0#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/93">
<admission:course/>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>5</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/102">
<admission:course>bachelor</admission:course>
<admission:qualification>mis</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/76">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>2</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>mcse\=i</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/75">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>2</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/77">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>2</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer +
Internet</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/51">
```

```
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>No</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>Non-IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/21">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>mcp+internet</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/22">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Professional + Internet
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/46">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>1</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/4">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/27">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
```

```
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/49">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>5</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/59">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer +
Internet </admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/71">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/101">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/78">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/79">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
```

```
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/50">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>5</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/81">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/80">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>Non-IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/52">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>4</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/111">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Professional + Internet
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
```

```
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/5">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/19">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>4</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/24">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>10</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/17">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/57">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer +
Internet </admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>1</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/58">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
```

```
<admission:certificate>Microsoft Certified Systems Engineer +
Internet </admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/56">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer +
Internet </admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>6</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/94">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/25">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>10</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/16">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/54">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/20">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
```

```
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/108">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>5</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/55">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>6</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/109">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>10</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/82">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/83">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>3</admission:experience>
```

```
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/110">
<admission:course>Graduate Diploma</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>Non-IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/36">
<admission:course>Master Information Systems</admission:course>
<admission:qualification/>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/35">
<admission:course>Master Information Systems</admission:course>
<admission:qualification/>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/34">
<admission:course>Master Information Systems</admission:course>
<admission:qualification/>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/74">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>2</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/91">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
```

```
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/43">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>1</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/3">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/38">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/42">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>4</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/14">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>5</admission:experience>
</rdf:Description>
```

```
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/31">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/44">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer +
Internet </admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>1</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/70">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/26">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/103">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/2">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
```

```
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/39">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/41">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2
Platform</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>4</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/1">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Bachelor</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/15">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>dip</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Nil</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/68">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>1</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/64">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
```

```
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>2</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/11">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/63">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>4</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/30">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Microsoft Certified Systems Engineer
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>20</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/67">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>1</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/66">
<admission:course>Master Information Systems</admission:course>
<admission:qualification>Diploma</admission:qualification>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>Sun Certified Programmer for Java 2 Platform
</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
```

```
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/33">
<admission:course>mis</admission:course>
<admission:qualification/>
<admission:year>3</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>scjp</admission:certificate>
<admission:occupation>Nil</admission:occupation>
<admission:experience>0</admission:experience>
</rdf:Description>
<rdf:Description
rdf:about="http://dsl08.dsl.uow.edu.au/is/admission/case/32">
<admission:course>mis</admission:course>
<admission:qualification>bachelor</admission:qualification>
<admission:year>2</admission:year>
<admission:it>Yes</admission:it>
<admission:certificate>mcse</admission:certificate>
<admission:occupation>IT related career</admission:occupation>
<admission:experience>3</admission:experience>
</rdf:Description>
</rdf:RDF>
```

```
<!-- C:\Tomcat\webapps\ROOT\proj\search2.jsp -->

<!--

Search2.jsp runs RDQL against original cases (cases.rdf) initially.
If none of original cases found,
it continue searches in adapted cases (ccases.rdf).

-->

<%@ include file="header.html" %>

<%@ page
        import="com.hp.hpl.mesa.rdf.jena.model.*,
        com.hp.hpl.mesa.rdf.jena.mem.*,
        com.hp.hpl.mesa.rdf.jena.common.*,

        com.hp.hpl.jena.rdf.query.*,

        java.util.*,
        java.io.*,

        java.sql.*,
        java.sql.Statement
        "
%>

<jsp:useBean id="myObj" scope="page" class="proj.Cases" />
<jsp:setProperty name="myObj" property="course" param="course" />
<jsp:setProperty name="myObj" property="qualification" param="qual"
/>
<jsp:setProperty name="myObj" property="certificate" param="cert" />
<jsp:setProperty name="myObj" property="year" param="qyear" />
<jsp:setProperty name="myObj" property="it" param="itco" />
<jsp:setProperty name="myObj" property="occupation" param="occu" />
<jsp:setProperty name="myObj" property="experience" param="oyear" />

<!--
    Ontology looks up for commonly used list of words.
    eg. graduate disploma or grad. dipl and
    master information systems or mis for courses
    and so on.
-->

<%

    try
    {
        Model model = new ModelMem();
        model.read(
         new FileReader(
            "c:\\Tomcat\\webapps\\ROOT\\proj\\ontology1.rdf"),
                "http://dsl08.dsl.uow.edu.au/is/admission-rdf/AO#",
                "RDF/XML");

        String queryString = " SELECT ?x, ?z, ?fn, ?sn " +
        " WHERE (?x, <AO:FN>, ?fn), " +
        "       (?x, <AO:N>, ?z), " +
        "       (?z, <AO:SN>, ?sn) " +
        " USING AO for " +
        " <http://dsl08.dsl.uow.edu.au/is/admission-rdf/AO#>";
```

```
Query query = new Query(queryString);
query.setSource(model);
QueryExecution qe = new QueryEngine(query);

QueryResults results = qe.exec();
for (Iterator iter = results; iter.hasNext(); )
{
    ResultBinding res = (ResultBinding)iter.next();
    Object x = res.get("x");
    Object fn = res.get("fn");
    Object sn = res.get("sn");

    String X = x.toString();
    String FN = fn.toString();
    String SN = sn.toString();
```

if (X.equalsIgnoreCase(

"http://dsl08.dsl.uow.edu.au/is/admission/Courses/

```
 GraduateDiploma"))
 {
    if (myObj.getCourse().equalsIgnoreCase(FN) ||
       myObj.getCourse().equalsIgnoreCase(SN) )
    {
       myObj.setCourse("Graduate Diploma");
    }
 }

 if (X.equalsIgnoreCase(
 "http://dsl08.dsl.uow.edu.au/is/admission/
 Courses/MasterInformationSystems"))
 {
    if (myObj.getCourse().equalsIgnoreCase(FN) ||
       myObj.getCourse().equalsIgnoreCase(SN) )
    {
       myObj.setCourse("Master Information Systems");
    }
 }

 if (X.equalsIgnoreCase(
 "http://dsl08.dsl.uow.edu.au/is/admission/
 Qualification/Diploma"))
 {
    if ( myObj.getQualification().equalsIgnoreCase(FN) ||
       myObj.getQualification().equalsIgnoreCase(SN) )
    {
       myObj.setQualification("Diploma");
    }
 }

 if ( X.equalsIgnoreCase(
 "http://dsl08.dsl.uow.edu.au/is/admission/
 Qualification/Bachelor"))
 {
 if ( myObj.getQualification().equalsIgnoreCase(FN) ||
       myObj.getQualification().equalsIgnoreCase(SN) )
 {
       myObj.setQualification("Bachelor");
```

```
            }
         }

         if ( X.equalsIgnoreCase(
         "http://dsl08.dsl.uow.edu.au/is/admission/
         Qualification/Master"))
         {
            if ( myObj.getQualification().equalsIgnoreCase(FN) ||
               myObj.getQualification().equalsIgnoreCase(SN) )
            {
               myObj.setQualification("Master");
            }
         }

         if ( X.equalsIgnoreCase(
            "http://dsl08.dsl.uow.edu.au/is/admission/
            Qualification/Honours"))
         {
            if ( myObj.getQualification().equalsIgnoreCase(FN) ||
               myObj.getQualification().equalsIgnoreCase(SN) )
            {
               myObj.setQualification("Bachelor of Honours");
            }
         }

         // look for professional certificates on ontology

         if ( X.equalsIgnoreCase(

"http://dsl08.dsl.uow.edu.au/is/admission/Certificates/SCJP"))
         {
            if (myObj.getCertificate().equalsIgnoreCase(FN) ||
               myObj.getCertificate().equalsIgnoreCase(SN) )
            {
               myObj.setCertificate(
                  "Sun Certified Programmer for Java 2 Platform");
            }
         }

         if (X.equalsIgnoreCase(
            "http://dsl08.dsl.uow.edu.au/is/admission/
            Certificates/SCJD"))
         {
            if (myObj.getCertificate().equalsIgnoreCase(FN) ||
               myObj.getCertificate().equalsIgnoreCase(SN) )
            {
               myObj.setCertificate(
                  "Sun Certified Developer for Java 2 Platform");
            }
         }

         if (X.equalsIgnoreCase(

"http://dsl08.dsl.uow.edu.au/is/admission/Certificates/MCP"))
         {
            if (myObj.getCertificate().equalsIgnoreCase(FN) ||
               myObj.getCertificate().equalsIgnoreCase(SN) )
            {
               myObj.setCertificate(
                  "Microsoft Certified Professional");
            }
```

```
         }

         if (X.equalsIgnoreCase(

"http://dsl08.dsl.uow.edu.au/is/admission/Certificates/MCSA"))
         {
            if (myObj.getCertificate().equalsIgnoreCase(FN) ||
               myObj.getCertificate().equalsIgnoreCase(SN) )
            {
               myObj.setCertificate(
                  "Microsoft Certified Systems Administrator");
            }
         }


         if (X.equalsIgnoreCase(

"http://dsl08.dsl.uow.edu.au/is/admission/Certificates/MCSE"))
         {
            if (myObj.getCertificate().equalsIgnoreCase(FN) ||
               myObj.getCertificate().equalsIgnoreCase(SN) )
            {
               myObj.setCertificate(
                  "Microsoft Certified Systems Engineer");
            }
         }

         if (X.equalsIgnoreCase(
         "http://dsl08.dsl.uow.edu.au/is/admission/
         Certificates/MCP+I"))
         {
            if (myObj.getCertificate().equalsIgnoreCase(FN) ||
               myObj.getCertificate().equalsIgnoreCase(SN) )
            {
               myObj.setCertificate(
                  "Microsoft Certified Professional + Internet");
            }
         }

         if (X.equalsIgnoreCase(
         "http://dsl08.dsl.uow.edu.au/is/admission/
         Certificates/MCSE+I"))
         {
            if (myObj.getCertificate().equalsIgnoreCase(FN) ||
               myObj.getCertificate().equalsIgnoreCase(SN) )
            {
               myObj.setCertificate(
                  "Microsoft Certified Systems Engineer + Internet");
            }
         }

         if (X.equalsIgnoreCase(

"http://dsl08.dsl.uow.edu.au/is/admission/Certificates/OCP"))
         {
            if (myObj.getCertificate().equalsIgnoreCase(FN) ||
               myObj.getCertificate().equalsIgnoreCase(SN) )
            {
               myObj.setCertificate("Oracle Certified Professional");
            }
         }
```

```
         if (X.equalsIgnoreCase(

"http://dsl08.dsl.uow.edu.au/is/admission/Certificates/CNE"))
         {
            if (myObj.getCertificate().equalsIgnoreCase(FN) ||
               myObj.getCertificate().equalsIgnoreCase(SN) )
            {
               myObj.setCertificate("Certified Novell Engineer");
            }
         }

         if (myObj.getCertificate() == "")
         {
            myObj.setCertificate("Nil");
         }
            //System.out.println("x " + x + "\n FN: " +
                  FN + " \n SN: " + SN);
        }
        results.close();
    } catch (Exception ex) {
        System.err.println("Exception: " + ex);
        ex.printStackTrace(System.err);
    }

%>

<center>
<table width="100%" cellpadding="3" cellspacing="3" border="0">
<tr bgcolor="#004080">
<td colspan="9"><font color="#ffffff">Your query is...</font></td>
</tr>
<tr class="bg_color">
<td width="300">Resource</td>
<td>Course</td>
<td>Qualification</td>
<td>Year</td>
<td>IT related course</td>
<td>Professional Certificate</td>
<td>Occupation</td>
<td>Experience</td>
<td align="right"> </td>
</tr>
<tr>
   <td> </td>
    <td><% out.println(myObj.getCourse()); %></td>
    <td><% out.println(myObj.getQualification()); %></td>
    <td><% out.println(myObj.getYear()); %></td>
    <td><% out.println(myObj.getIt()); %></td>
    <td><% out.println(myObj.getCertificate()); %></td>
    <td><% out.println(myObj.getOccupation()); %></td>
    <td><% out.println(myObj.getExperience()); %></td>
   <td> </td>
</tr>
</table>
</center>

<%

boolean isOriginalExists = true;
boolean isAdaptedExists  = true;
```

```
if (isOriginalExists == true)
{
    String url      = "jdbc:mysql://localhost:3306/test";
    String user     = "sk33";
    String pass     = "o151fs";
    Connection conn = null;
    Statement stmt  = null;

    try
    {
        // Load the MySQL driver
        // JDBC 3.0 driver
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        // JDBC 2.x driver
        // Class.forName("org.gjt.mm.mysql.Driver").newInstance();
    }   // if the driver not loaded,
        // an exception would have been thrown
    catch (Exception e)
    {
        System.err.println("Unable to load driver.");
        e.printStackTrace();
        return;
    }
    try
    {   // Setup a connection to the database
        conn = DriverManager.getConnection(url, user, pass);
        stmt = conn.createStatement();

        String SQLText = "INSERT INTO icases " +
            " (course, qualification, year, it, " +
            " certificate, occupation, experience, published) " +
            " VALUES( '" + myObj.getCourse() + "', '" +
            myObj.getQualification() + "', " +
            myObj.getYear() + ", '" + myObj.getIt() + "', '" +
            myObj.getCertificate() + "', '" +
            myObj.getOccupation() + "', " +
            myObj.getExperience() + ", now() " + ")";

        //out.println(SQLText);

        // execute the query statement and capture the results
        ResultSet rs = stmt.executeQuery(SQLText);

    }   // If the connection to the DB failed,
        // an exception would have been thrown
    catch (SQLException e)
    {
        e.printStackTrace();
    }
    finally
    {
        try
        {
            // Freeup any resources associated
            // with the statement and result set
            stmt.close();
            conn.close();
        }
        catch (SQLException e)
        {
```

```
            e.printStackTrace();
        }
    }

%>

<center>
<table width="100%" cellpadding="3" cellspacing="3" border="0">
<tr bgcolor="#004080">
<td colspan="9"><font color="#ffffff">Original cases</font></td>
</tr>
<tr class="bg_color">
<td width="300">Resource</td>
<td>Course</td>
<td>Qualification</td>
<td>Year</td>
<td>IT related course</td>
<td>Professional Certificate</td>
<td>Occupation</td>
<td>Experience</td>
<td align="right"> </td>
</tr>
    <%
    try
    {
        Model model = new ModelMem();

        model.read(new
            FileReader(
                  "c:\\Tomcat\\webapps\\ROOT\\proj\\ocases.rdf"),
                  "http://dsl08.dsl.uow.edu.au/is/admission-
rdf/1.0#",
                  "RDF/XML") ;

        String queryString = " SELECT ?x " +
        " WHERE (?x, <admission:course>, \"" +
        myObj.getCourse() + "\"), " +
        "       (?x, <admission:course>, ?course), " +
        "       (?x, <admission:qualification>, \"" +
        myObj.getQualification() + "\"), " +
        "       (?x, <admission:qualification>, ?qualification), " +
        "       (?x, <admission:year>, ?year), " +
        "       (?x, <admission:it>, \"" + myObj.getIt() + "\"), " +
        "       (?x, <admission:it>, ?it), " +
        "       (?x, <admission:certificate>, \"" +
        myObj.getCertificate() + "\"), " +
        "       (?x, <admission:certificate>, ?certificate), " +
        "       (?x, <admission:occupation>, \"" +
        myObj.getOccupation() + "\"), " +
        "      (?x, <admission:occupation>, ?occupation), " +
        "      (?x, <admission:experience>, ?experience) " +
        " AND   ?year == " + myObj.getYear() +
        "       && ?experience == " + myObj.getExperience() +
        " USING admission for " +
        " <http://dsl08.dsl.uow.edu.au/is/admission-rdf/1.0#>";

        Query query = new Query(queryString);
        query.setSource(model);
        QueryExecution qe = new QueryEngine(query);

        QueryResults results = qe.exec();
```

```
        int j = 0;
        for (Iterator iter = results; iter.hasNext(); )
        {
            ResultBinding res = (ResultBinding)iter.next();
            Object x    = res.get("x");
            Object cour = res.get("course");
            Object qual = res.get("qualification");
            Object year = res.get("year");
            Object itco = res.get("it");
            Object cert = res.get("certificate");
            Object occu = res.get("occupation");
            Object expe = res.get("experience");

            out.println("<tr><td>" + x +
            "</td><td>" + cour + "</td><td>" + qual +
            "</td><td>" + year + "</td><td>" + itco +
            "</td><td>" + cert + "</td><td>" + occu +
            "</td><td>" + expe + "</td></tr>\n");
            j++;
        }
        results.close();
        if (j == 0)
        {
            isOriginalExists = false;
        }
      else
      {
         isOriginalExists = true;
         %>
         <tr class="bg_color">
         <td colspan="9">
         <b>Result:</b> </br>You have enquired about
         the eligibility for
         <b><% out.println(myObj.getCourse()); %></b> course.
         From the information you have provided,
         we advise that you are <b>eligible</b> for the course,
please
         submit a formal application to the University
         </td>
         </tr>
         <%
      }
    } catch (Exception e) {
        e.printStackTrace(System.err);
    }
}
if (isOriginalExists == false)
{
%>
<tr>
<td colspan="7">
<font color="red">No original case is found!</font>
</td>
</tr>
</table>
</center>

<center>
<table width="100%" cellpadding="3" cellspacing="3" border="0">
<tr bgcolor="#004080">
<td colspan="9"><font color="#ffffff">Adapted cases</font></td>
```

```
</tr>
<tr class="bg_color">
<td width="300">Resource</td>
<td>Course</td>
<td>Qualification</td>
<td>Year</td>
<td>IT related course</td>
<td>Professional Certificate</td>
<td>Occupation</td>
<td>Experience</td>
<td align="right"> </td>
</tr>
    <%
    try
    {
        Model model = new ModelMem();

        model.read(new
            FileReader(
                  "c:\\Tomcat\\webapps\\ROOT\\proj\\ccases.rdf"),
                  "http://dsl08.dsl.uow.edu.au/is/admission-
rdf/1.0#",
                  "RDF/XML") ;

        String queryString = " SELECT ?x "+
        " WHERE  (?x, <admission:course>, \""
        + myObj.getCourse() + "\"), " +
        "       (?x, <admission:course>, ?course), " +
        "       (?x, <admission:qualification>, \"" +
        myObj.getQualification() + "\"), " +
        "       (?x, <admission:qualification>, ?qualification), " +
        "       (?x, <admission:year>, ?year), " +
        "       (?x, <admission:it>, \"" + myObj.getIt() + "\"), " +
        "       (?x, <admission:it>, ?it), " +
        "       (?x, <admission:certificate>, \"" +
        myObj.getCertificate() + "\"), " +
        "       (?x, <admission:certificate>, ?certificate), " +
        "       (?x, <admission:occupation>, \"" +
        myObj.getOccupation() + "\"), " +
        "       (?x, <admission:occupation>, ?occupation), " +
        "       (?x, <admission:experience>, ?experience) " +
        " AND   ?year <= " + myObj.getYear() +
        "       && ?experience <= " + myObj.getExperience() +
        " USING admission for " +
        <http://dsl08.dsl.uow.edu.au/is/admission-rdf/1.0#>";

        Query query = new Query(queryString);
        query.setSource(model);
        QueryExecution qe = new QueryEngine(query);

        QueryResults results = qe.exec();
        int k = 0;
        for (Iterator iter = results; iter.hasNext(); )
        {
            ResultBinding res = (ResultBinding)iter.next();
            Object x = res.get("x");
            Object cour = res.get("course");
            Object qual = res.get("qualification");
            Object year = res.get("year");
            Object itco = res.get("it");
            Object cert = res.get("certificate");
```

```
            Object occu = res.get("occupation");
            Object expe = res.get("experience");

            out.println("<tr><td>" + x + "</td><td>" +
            cour + "</td><td>" + qual +
            "</td><td>" + year + "</td><td>" + itco +
            "</td><td>" + cert + "</td><td>" + occu +
            "</td><td>" + expe + "</td><td> </td></tr>\n");
            k++;
          }
          results.close();
          if (k <= 0)
          {
              isAdaptedExists = false;
          }
        else
        {
           isAdaptedExists = true;
           isOriginalExists = true;
           %>
           <tr class="bg_color">
           <td colspan="9"><b>Result:</b> </br>You have enquired about
           the eligibility for
           <b><% out.println(myObj.getCourse()); %></b> course.
           From the information you have provided, we advise that
           you are <b>eligible</b> for the course, please
           submit a formal application to the University
           </td>
           </tr>
           <%
          }
      } catch (Exception e) {
          e.printStackTrace(System.err);
      }
}

if (!isAdaptedExists)
{
%>
<tr>
<td colspan="7">
<font color="red">No adapted case is found!</font>
</td>
</tr>
</table>
</center>

   <%
   /*
    * Similarity Assement
   */

   %>
   <center>
   <table width="100%" cellpadding="3" cellspacing="3" border="0">
   <tr bgcolor="#004080">
   <td colspan="9">
   <font color="#ffffff">Similar original cases</font>
   </td>
   </tr>
   <tr class="bg_color">
```

```
   <td width="300">Resource</td>
   <td>Course</td>
   <td>Qualification</td>
   <td>Year</td>
   <td>IT related course</td>
   <td>Professional Certificate</td>
   <td>Occupation</td>
   <td>Experience</td>
   <td align="right">*Similarity</td>
   </tr>
      <%
      // none of orginal and adated cases are found!
      // do similarity assessment.

      try
      {
         Model mode2 = new ModelMem();

         mode2.read(new
            FileReader(
                  "c:\\Tomcat\\webapps\\ROOT\\proj\\ocases.rdf"),
                  "http://dsl08.dsl.uow.edu.au/is/admission-
rdf/1.0#",
                  "RDF/XML") ;

         String queryString2 = "SELECT ?x " +
         " WHERE (?x, <admission:course>, ?course), " +
         "      (?x, <admission:qualification>, ?qualification), " +
         "      (?x, <admission:year>, ?year), " +
         "       (?x, <admission:it>, ?it), " +
         "       (?x, <admission:certificate>, ?certificate), " +
         "      (?x, <admission:occupation>, ?occupation), " +
         "      (?x, <admission:experience>, ?experience) " +
         " USING admission for " +
         <http://dsl08.dsl.uow.edu.au/is/admission-rdf/1.0#>";

         Query query2 = new Query(queryString2);
         query2.setSource(mode2);
         QueryExecution qe2 = new QueryEngine(query2);

         QueryResults results2 = qe2.exec();

         for (Iterator iter2 = results2; iter2.hasNext(); )
         {
            ResultBinding res2 = (ResultBinding)iter2.next();

            Object x_O    = res2.get("x");
            Object cour_O = res2.get("course");
            Object qual_O = res2.get("qualification");
            Object year_O = res2.get("year");
            Object itco_O = res2.get("it");
            Object cert_O = res2.get("certificate");
            Object occu_O = res2.get("occupation");
            Object expe_O = res2.get("experience");

            String COUR = cour_O.toString();
            String QUAL = qual_O.toString();
            int YEAR = Integer.parseInt(year_O.toString());
            String ITCO = itco_O.toString();
            String CERT = cert_O.toString();
            String OCCU = occu_O.toString();
```

```
int EXPE = Integer.parseInt(expe_O.toString());

int intCOUR_O = 0;
int intQUAL_O = 0;
int intYEAR_O = 0;
int intITCO_O = 0;
int intCERT_O = 0;
int intOCCU_O = 0;
int intEXPE_O = 0;

// weighting factors
// weight 1 is less than than 6
int wCOUR = 1;
int wQUAL = 2;
int wYEAR = 2;
int wITCO = 1;
int wCERT = 1;
int wOCCU = 1;
int wEXPE = 1;

/* integer values of course, qualification, year,
 * certificate, occupation and experience
 * for ORIGINAL CASES
 */
if (COUR.equalsIgnoreCase("graduate diploma"))
{
   intCOUR_O = 1;
}

if (QUAL.equalsIgnoreCase("diploma"));
{
   intQUAL_O = 1;
}
if (QUAL.equalsIgnoreCase("bachelor"));
{
   intQUAL_O = 2;
}
if (QUAL.equalsIgnoreCase("master"));
{
   intQUAL_O = 3;
}
if (YEAR < 3)
{
  intYEAR_O = 0;
}

if (YEAR >= 3)
{
  intYEAR_O = 1;
}
if (ITCO.equalsIgnoreCase("yes"))
{
   intITCO_O = 1;
}
if (ITCO.equalsIgnoreCase("no"))
{
   intITCO_O = 0;
}
if (CERT.equalsIgnoreCase("nill"))
{
   intCERT_O = 0;
```

```
            }
            if (CERT.equalsIgnoreCase("java") ||
                CERT.equalsIgnoreCase("mcse") ||
                CERT.equalsIgnoreCase("mcsd"))
            {
               intCERT_O = 1;
            }
            if (OCCU.equalsIgnoreCase("nill"))
            {
               intOCCU_O = 0;
            }
            if (OCCU.equalsIgnoreCase("non-it related career"))
            {
               intOCCU_O = 1;
            }
            if (OCCU.equalsIgnoreCase("it related career"))
            {
               intOCCU_O = 2;
            }
            if (EXPE < 2)
            {
               intEXPE_O = 0;
            }
               if (EXPE >= 2)
            {
               intEXPE_O = 1;
            }

            int intCOUR_T = 0;
            int intQUAL_T = 0;
            int intYEAR_T = 0;
            int intITCO_T = 0;
            int intCERT_T = 0;
            int intOCCU_T = 0;
            int intEXPE_T = 0;

            /* integer values of course, qualification,
             * year, certificate,
             * occupation and experience for TARGET CASES
             */
            if (myObj.getCourse().equalsIgnoreCase("graduate
diploma"))
            {
               intCOUR_T = 1;
            }
            if
(myObj.getQualification().equalsIgnoreCase("diploma"));
            {
               intQUAL_T = 1;
            }
            if
(myObj.getQualification().equalsIgnoreCase("bachelor"));
            {
               intQUAL_T = 2;
            }
            if (myObj.getQualification().equalsIgnoreCase("mis"));
            {
               intQUAL_T = 3;
            }
            if (myObj.getYear() < 3)
            {
```

```
                intYEAR_T = 0;
              }
              if (myObj.getYear() >= 3)
              {
                intYEAR_T = 1;
              }
              if (myObj.getIt().equalsIgnoreCase("yes"))
              {
                 intITCO_T = 1;
              }
              if (myObj.getIt().equalsIgnoreCase("no"))
              {
                 intITCO_T = 0;
              }
              if (myObj.getCertificate().equalsIgnoreCase("nill"))
              {
                 intCERT_T = 0;
              }
              if (myObj.getCertificate().equalsIgnoreCase("java") ||
                 myObj.getCertificate().equalsIgnoreCase("mcse") ||
                 myObj.getCertificate().equalsIgnoreCase("mcsd"))
              {
                 intCERT_T = 1;
              }
              if (myObj.getOccupation().equalsIgnoreCase("nill"))
              {
                 intOCCU_T = 0;
              }
              if (myObj.getOccupation().equalsIgnoreCase(
                    "non-it related career"))
              {
                 intOCCU_T = 1;
              }
              if (myObj.getOccupation().equalsIgnoreCase(
                    "it related career"))
              {
                 intOCCU_T = 2;
              }
              if (myObj.getExperience() < 2)
              {
                intEXPE_T = 0;
              }
              if (myObj.getExperience() >= 2)
              {
                intEXPE_T = 1;
              }

              int similarity = 0;

              similarity = (
                 (intCOUR_O - intCOUR_T) +
                 (intQUAL_O - intQUAL_T) +
                 (intYEAR_O - intYEAR_T) +
                 (intCERT_O - intCERT_T) +
                 (intOCCU_O - intOCCU_T) +
                 (intEXPE_O - intEXPE_T) );

              out.println("<tr><td>" + x_O + "</td><td>" + cour_O +
              "</td><td>" + qual_O + "</td><td>" + year_O + "</td><td>"
+
              itco_O + "</td><td>" + cert_O + "</td><td>" + occu_O +
```

```
            "</td><td>" + expe_O + "</td><td align=\"right\">" +
            similarity + "</td></tr>\n");
          }
          results2.close();
       } catch (Exception e) {
          e.printStackTrace(System.err);
       }
   //} // end of similarity assessment
   %>
   <tr class="bg_color">
   <td colspan="9">
   <b>Result:</b> Whichever gives the smallest value (or closest to
0)
   in "*Similarity" on the last column of the table is
   the nearest nearghboring case to your inputed case.
   It means that if there is anything close to your case you might be
   eligible to apply your desired course -
   <b><% out.println(myObj.getCourse()); %></b>.
   </td>
   </tr>
   </table>
   </center>
<%
}
%>
<%@ include file="footer.html" %>
```

```
/* C:\Tomcat\webapps\ROOT\proj\SimpleTransform1.java */

/*
 * the XML file (rcases.xml) to newly retrived RDF file (rcases.rdf)
 */

// Imported TraX classes
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.stream.StreamSource;
import javax.xml.transform.stream.StreamResult;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerConfigurationException;

// Imported java classes
import java.io.FileOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

/*
 *  Use the TraX interface to perform a transformation
 */
public class SimpleTransform1
{
    public void xsl_transform()
throws TransformerException, TransformerConfigurationException,
      FileNotFoundException, IOException
    {
      // Use the static TransformerFactory.newInstance() method
      // to instantiate a TransformerFactory.
      // The javax.xml.transform.TransformerFactory
      // system property setting determines the actual class
      // to instantiate
      // -- org.apache.xalan.transformer.TransformerImpl.
      TransformerFactory tFactory =
      TransformerFactory.newInstance();

      // Use the TransformerFactory to instantiate a Transformer
      // that will work with the stylesheet you specify.
      // This method call also processes the stylesheet
      // into a compiled Templates object.
      Transformer transformer =
      tFactory.newTransformer(new StreamSource("rcases.xsl"));

      // Use the Transformer to apply
      // the associated Templates object to an XML document
      // (rcases.xml) and write the output to a file (rcases.rdf).
      transformer.transform(
         new StreamSource("rcases.xml"), new StreamResult(
         new FileOutputStream("rcases.rdf")));

        System.out.println("The result is in rcases.rdf!");
    }
}
```

```
/* C:\Tomcat\webapps\ROOT\proj\SimpleTransform2.java */

/*
 * Transforms cases.xml to cases.rdf
 */

// Imported TraX classes
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.stream.StreamSource;
import javax.xml.transform.stream.StreamResult;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerConfigurationException;

// Imported java classes
import java.io.FileOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

/*
 *  Use the TraX interface to perform a transformation
 */
public class SimpleTransform2
{
   public void xsl_transform() throws TransformerException,
      TransformerConfigurationException,
      FileNotFoundException, IOException
   {
      // Use the static TransformerFactory.newInstance() method
      // to instantiate a TransformerFactory.
      // The javax.xml.transform.TransformerFactory
      // system property setting determines the actual class
      // to instantiate --
      // org.apache.xalan.transformer.TransformerImpl.
      TransformerFactory tFactory = TransformerFactory.newInstance();

      // Use the TransformerFactory to instantiate
      // a Transformer that will work with
      // the stylesheet you specify. This method call
      // also processes the stylesheet
      // into a compiled Templates object.
      Transformer transformer =
      tFactory.newTransformer(new StreamSource("ccases.xsl"));

      // Use the Transformer to apply the associated
      // Templates object to an XML document
      // (foo.xml) and write the output to a file (foo.out).
      transformer.transform(new StreamSource("ccases.xml"),
      new StreamResult(new FileOutputStream("ccases.rdf")));

      System.out.println("The result is in ccases.rdf!");
      }
}
```