

December 2002

Intelligent Knowledge Acquisition with Case-Based Reasoning Techniques

Seung Hwan Kang
University of Wollongong

Sim Lau
University of Wollongong

Follow this and additional works at: <http://aisel.aisnet.org/acis2002>

Recommended Citation

Kang, Seung Hwan and Lau, Sim, "Intelligent Knowledge Acquisition with Case-Based Reasoning Techniques" (2002). *ACIS 2002 Proceedings*. 45.
<http://aisel.aisnet.org/acis2002/45>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISEL). It has been accepted for inclusion in ACIS 2002 Proceedings by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact elibrary@aisnet.org.

Intelligent Knowledge Acquisition with Case-Based Reasoning Techniques

Seung Hwan Kang

Dr. Sim Kim Lau

School of Economics and Information Systems
University of Wollongong
Wollongong, NSW, Australia
sk33uow.edu.au

Abstract

Knowledge management systems are an emerging area gaining interest in organisations. This paper discusses the application of case based reasoning techniques and intelligent agents in the knowledge acquisition phase of knowledge management systems so that an intelligent knowledge acquisition process is possible.

Keywords

Knowledge Acquisition, Knowledge Management Systems

INTRODUCTION

Knowledge Management Systems (KMS) are an emerging area gaining interest among organisations and the purpose of building a KMS is to share corporate knowledge in the organisation (Walsham, 2001). It is a computerised system that has been developed with a human-centered view of knowledge. Organisations have also begun to realise that valuable knowledge is often being taken away from people as a result of staff resignation, retirement, downsizing or outsourcing. At the same time, organisations have also found that knowledge sharing and reuse have been inefficient in KMS. One of the problems that have been identified is the inefficient acquisition and adaptation process of new knowledge in the KMS cycle, and current development of KMS is weak in terms of facilitating the concept of knowledge sharing and reuse (Dubitzky *et al.*, 1999). The concept of reusing knowledge refers to reusing old solutions in current problem when similar problems arise.

This paper discusses the application of Case-Based Reasoning (CBR) techniques in the knowledge acquisition (KA) phase of KMS development cycle. A conceptualised part of knowledge representing past experience can be represented as cases (Kolodner, 1993). The basic idea of adapting CBR in KMS is to re-apply retrieved and retained knowledge based on past cases and apply it in a new set of knowledge domain that is similar to the existing ones. The sharing and reuse of knowledge are made possible through the adaptation of new knowledge using new case behaviour and reusing of knowledge from existing cases. This paper also investigates the feasibility of deploying intelligent agents using techniques from CBR during the KA phase of KMS development cycle.

This paper is organised as follows. The next section discusses techniques used in the CBR systems, followed by discussion on the role of intelligent agents in the KA process for a networked environment, and discussion of how agents can use CBR techniques to achieve an intelligent KA process. Finally, proposed future research directions are highlighted.

CASE-BASED REASONING SYSTEMS TECHNIQUES

According to Kolodner (1993:13), “a case is a conceptualised piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoned.” The reason of representing knowledge as a case is because it supplies a wide range of content in the form of problem and solution descriptions. When new problem issue arises, the retrieval process identifies the case with the most similar problem description based on past cases. If there is any stored problem description that is the same as the case under investigation, then detailed solution of that stored case can be reused. If necessary, the adaptation process can occur and a new solution can be created for the new problem.

In the following section, we will discuss commonly used techniques in the CBR cycle, consisting of the following phases: retrieve, reuse, review and retain (Kolodner, 1993; Aamodt and Plaza, 1994; Watson, 1997).

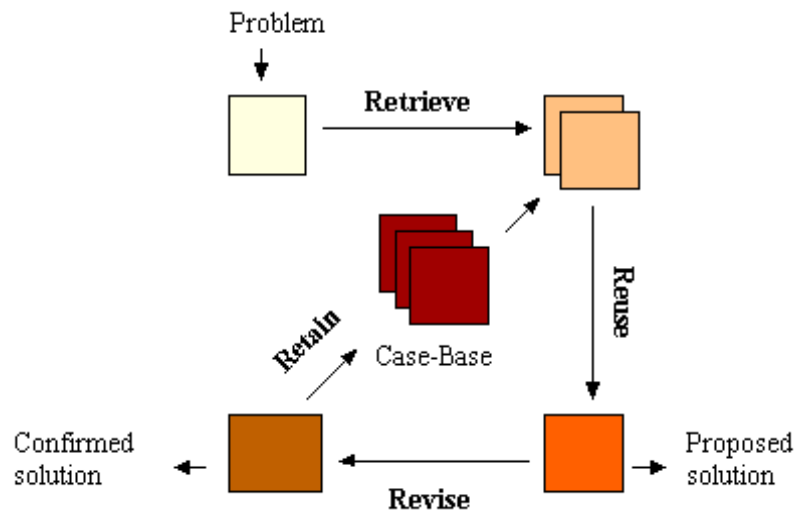


Figure 1: The CBR Cycle (Watson, 1997:17)

Case retrieval is a technique to decide which case is similar to a source case. When the case that is the most similar is found, then the CBR system retrieves that case that can provide a detailed solved problem description to a new problem. The two most widely used techniques of case retrieval are: Nearest-Neighbour retrieval and inductive retrieval.

Nearest-Neighbour Retrieval (NNR) is a technique to measure how similar the target case is to a source case (Watson, 1997). It processes retrieval of cases by comparison 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 case match can be represented by the following equation (Watson, 1997:28):

$$\text{Similarity}(T, S) = \sum_{i=1}^n f(T_i, S_i) \times w_i$$

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

The equation of the NNR represents the sum of similarity of the target case to the source case for all attributes multiplied by the importance weighting of individual attributes. However, the NNR technique is not efficient. This is because whenever new cases are introduced, indexing needs to be performed and it could affect efficiency.

Inductive retrieval is a technique 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 a target case is not found in the decision tree at runtime, then the CBR system does 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 indexing is used to retrieve a set of matching cases and then the nearest-

neighbour retrieval is used to rank the cases in the set according to their similarity to the target case.

Case storage, which is often referred to as case-memory or memory organisation, is used in the case reuse phase of the CBR cycle. It replicates the conceptual view of case representation in most storage devices. Cases are often reproduced to increase the quality of the solutions of the already solved problems in a given set of circumstances. Stored cases are also used for future reference (Leake, 1995). A reusable case is more user-acceptable because its solution has already been accepted by the previous user.

Indexes are commonly used in file and database systems to speed up retrieval and optimise accessibility of data. Indexing is also commonly used in the case retention 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; 2001). Watson (1997) suggests that the case representation should be characterised using indexes. The intention of characterised case representation is to balance between the storing methods and their indexes in order to simplify accessibility and retrieval of relevant cases.

Adaptation is used in the case revision phase of the CBR cycle. It is a technique to alter the retrieved case to reproduce a new solution to a new problem. The retrieved case can be changed so that it can be presented to suit the new use. The purpose of case adaptation is to improve the CBR system's overall problem solving ability using newly introduced cases for future use. The two most widely used techniques of case adaptation are structural adaptation and derivational adaptation.

Structural adaptation is a technique to apply adaptation rules or formulas directly to the stored solution in the CBR library. Once a case has been applied by adaptation rules or formulas, the CBR system adapts the case as a match for the new problem. On the other hand, derivational adaptation is a technique 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.

THE ROLE OF INTELLIGENT AGENTS IN THE KNOWLEDGE ACQUISITION PROCESS

KA is the process of translating implicit knowledge into explicit form, and it is the most crucial part of KMS development life cycle (Brule and Blount, 1989). During the 1980s, KA methods were designed to construct explicit knowledge from implicit knowledge using manual methods such as interviewing, tracking the reasoning process, and observing documented and undocumented knowledge. This process is used to find what knowledge is being used and how it is being used. However, manual methods are often costly, slow, and prone to error. Semi-automatic and automatic methods are then introduced by automating the KA process using artificial intelligence techniques with the aim to overcome poor productivity of manual KA approach (Liebowitz, 2001; Tsui *et al.*, 2000).

The popularity of networked infrastructure such as the World Wide Web, the Internet, intranets and extranets, have enabled the KA process to speed up. For example, a knowledge engineer can interview experts via electronic interviewing. Then documented knowledge can be submitted via electronic forms. The web search engine also can be used to integrate distributed knowledge during the KA phase. For example, during the KA phase, huge amounts of web links can be collected by web Spiders and placed in the storage devices. The following are some of the commonly used search engines available and their methods of search on the WWW:

- Keyword search using indexed words.
- Concept-based search using meaning.
- Refining search using user-defined options.
- Relevancy ranking based on search term frequency.
- Meta tags based on information about a document.

Keyword search using “indexed words” are popular. An example of keyword searching engine is AltaVista (<http://www.altavista.com>). The disadvantage of this method is it may produce 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 subjects in relation to the context.

Another method is refining searching, which uses user-defined options, can differ from one search engine to another. Generally, this method of searching is capable to include and/ or exclude more than one word in the search terms. An example of inclusive and exclusive refining searching is achieved by specifying the logical terms such as AND, OR, NOT, +, -, and quotation marks in the query. At the present time, most of the search engines include the refining searching 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, 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 (W3C, 1999). Some search engines such as AltaVista use Meta tags to index web documents. For example, to specify the author of a document, the Meta element “author” can be used as follows: <META name=“author” content=“Albert Einstein”>. Table 1 summarises the features of these major search engines.

	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 1: Summary of major search engines

This research proposes the use of intelligent agents in the KA process of KMS development in a networked environment such as the WWW. There are a number of definitions for intelligent 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” (Wooldridge and Jennings, 1995; Hayes-Roth, 1995).

Agents can be software, intelligent, or learning agents. Software agents are a software application that acts like human. It can be used to assist the KA process such as locating and filtering new knowledge. It is possible to use software agents to manipulate distributed

data on 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. On the other hand, an intelligent agent has human like characteristics such as autonomy, temporal continuity, reactivity, and is goal driven (Wooldridge and Jennings, 1995; Maes, 1995). A learning agent has an inherent characteristic of human beings to adapt its behaviour in order to improve its performance (Konar, 2000). With learning capability, the learning agent adapts the abstract patterns of relationship in the domain autonomously.

When software agents travel through the WWW, they are often restricted by the limited knowledge domain. For example, the user needs can be a combination of keywords or a description of user-defined software agents' tasks. Very often, software agents need to compare or merge information so that two or more terms can be used to refer to the same thing. Because of the importance of knowledge sharing and reuse in KMS, better support for standardised domains is needed. To achieve this, ontology can be applied to explicitly formalise the specification of a shared conceptualisation (Gruber, 1993a; Boicu et al., 2001). Ontology aims at capturing domain knowledge in a generic way so that it provides a commonly agreed understanding of a domain, which can be reused and shared (Gruber, 1993b; Gomez-Perez, 1999). This way, ontology can be applied to capture common interest of knowledge in the KA process and knowledge sharing is achievable. It can be used to conceptualise representation of a community of interest for human or software agents (Gruber, 1993a; Gomez-Perez, 1999; Sowa, 2000; Hendler, 2001; Berners-Lee et al., 2001). There are several ontology frameworks, which include inspirational, inductive, deductive, synthetic, and collaborative approaches (Holsapple and Joshi, 2002; Clyde and Joshi, 2002; Gruninger and Lee, 2002; Guarino, 2002). These approaches are briefly described as follows (Gruninger and Lee, 2002):

- Inspirational approach – is focus on the need of an ontology using individual imagination, creativity, and personal views on the domain of interest.
- Inductive approach – is concentrate 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.
- Deductive approach – is concerned with the general adoptive principles using example of filtering and distilling of the general notions.
- Synthetic approach – is method to characterise a relatively widespread use of ontologies, rather than separate ontologies, to express syntactic relationships of multiple ontologies.
- Collaborative approach – is focus on cooperative method of ontology design and development using multiple viewpoints such as iterative improvement of ontology development, and its consensus-building mechanism by relatively wide acceptance of interests.

DISCUSSION

This research aims to investigate the feasibility of deploying intelligent agents using CBR techniques during the KA phase of KMS development cycle. We aim to develop an intelligent KA process by employing software and learning agents using techniques in CBR. A typical KMS development cycle consists of: create knowledge, capture knowledge, refine knowledge, store knowledge, manage knowledge and disseminate knowledge (Turban and Aronson, 2001). There are similarities between the phases of CBR and the KMS cycle. Table 2 shows the similarity between the phases of CBR and the KMS cycle. The Retrieve phase in the CBR cycle corresponds 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 is similar to the Store, Manage and Disseminate knowledge phases in KMS.

CBR Cycle	KMS Cycle
Retrieve	Create knowledge, Capture knowledge
Reuse	Refine knowledge

CBR Cycle	KMS Cycle
Revise	
Retain	Store knowledge, Manage knowledge, Disseminate knowledge

Table 2: CBR Cycle vs. KMS Cycle

Figure 2 shows the phases in which CBR techniques can be applied in the KMS development cycle.

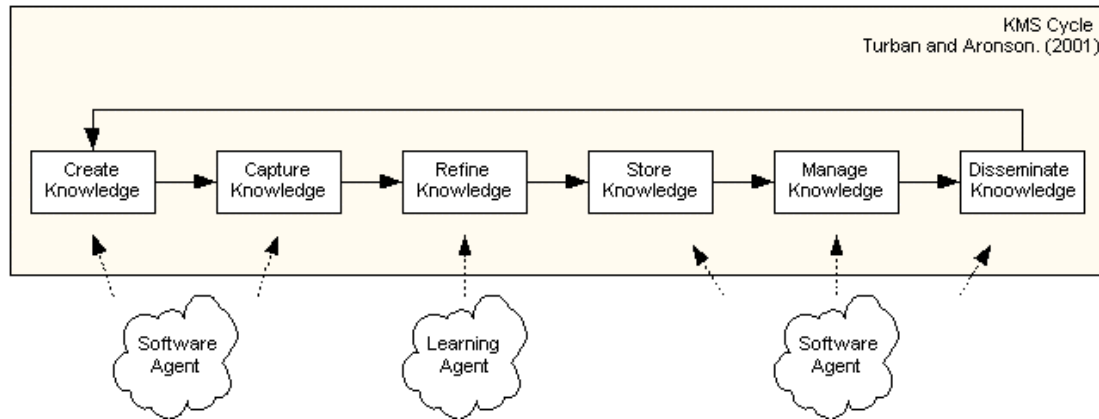


Figure 2: Application of CBR techniques in KMS cycle

To facilitate the KA process, ontology can be applied to the KMS as a form of meta-knowledge. The use of ontology allows consistent conceptualisation to be referenced in KMS. It explicitly formalises the specification of a shared conceptualisation during the KA phase. The knowledge in this case are conceptualised as cases that contain problem and solution descriptions. The use of ontology also minimises redundancies in the knowledge repository and agents can be deployed to allow sharing and reuse of existing knowledge, and to ensure new knowledge can be added to the knowledge repository. The inductive ontology approach will be used to allow consistent conceptualisation to be referenced in the KMS development life cycle.

A software agent will be added to the create knowledge and capture knowledge phases of the KMS cycle (see Figure 2). It is proposed that the software agent will use techniques based on the retrieval phase of the CBR cycle. We propose to use the approach proposed by Aamodt and Plaza (1994) and Watson (1997), in which a combination of inductive retrieval and nearest-neighbour retrieval are used to allow pattern to be matched and similar cases to be compared respectively. Thus the NNR technique is adopted for the similarity assessment and inductive retrieval is used for pattern matching. If similar cases are found in the KMS, then the software agent will retrieve existing cases from the KMS. This way existing knowledge is retrieved, shared and reused. On the other hand, if similar cases are not found, then a learning agent will be deployed. The learning agent allows new knowledge to be learned based on new case behaviour. The learning agent will use the derivational adaptation technique of the revision phase of CBR to adapt new knowledge. This way new knowledge represented in the form of new cases with new problem and solution descriptions will be added to the knowledge repository. This allows knowledge in the KMS to be refined. Therefore a learning agent will be added to the refine knowledge phase of KMS as shown in Figure 2. However, before new knowledge is stored, 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 phase.

FURTHER RESEARCH

The next step of our research will be focusing on the implementation issues of the above intelligent KA framework so that intelligent KA can be achieved through the deployment of intelligent agents and ontology. To develop the proposed intelligent framework, we propose

to use the semantic web features for knowledge representation. Semantic web allows information to reuse and share over the Web (Berners-Lee *et al.*, 2001; W3C, 2002a; 2002b). Semantic web also enables software agents to communicate with other software agents. The agents share terms of mapped or merged ontologies, which are usually defined using a language of Extensible Markup Language (XML) and the Resource Description Framework (RDF) (W3C, 2000; Goldfarb, 2001; W3C, 1999). This way, software agents share terms of mapped or merged ontologies defined in the language of XML and RDF over standard platform. With XML and RDF model, it is possible to offer a structure to assist with interoperability between software agents and web applications. XML and RDF can provide simple and effective solution to facilitate interoperability in the KMS based on metadata models.

REFERENCES

- Altavista, URL <http://www.altavista.com>
- Aamodt, A. and Plaza E. (1994). "Case-Based Reasoning: Foundational Issues, Methodological Variables, and System Approaches", *Artificial Intelligence Communication*, Vol. 7, No. 1.
- Barlow, L. (2002). "How Search Engines Work", URL: <http://www.monash.com/spidap4.html>
- Brule F. J. and Blount A. (1989). "Knowledge Acquisition", McGraw-Hill.
- Berners-Lee, T., Hendler, J. and Lassila, O. (2001). "The Semantic Web", May 2001, *Scientific American*, URL <http://www.sciam.com/2001/0501issue/0501berners-lee.html>
- Boicu, M., Tecuci, G., Stanescu, Balan, C. G. and Popovici E. (2001). "Ontologies and the Knowledge Acquisition Bottleneck", in *Proceedings of IJCAI-2001 Workshop on Ontologies and Information Sharing*, Seattle, Washington
- Clyde, H. W. and Joshi D.K. (2002). "A Collaborative Approach to Ontology Design", *Communications of the ACM*, Vol. 45, No. 2, pp.39-41
- Dubitzky, W., Buchner, G.A. and Azuaje, J.F. (1999). "Viewing Knowledge Management as a Case-Based Reasoning Application", the *AAAI-99 KM/CBR Workshop*, Munich, Germany, URL <http://www.aic.nrl.navy.mil/~aha/aaai99-kmcbw/papers/>
- Excite, URL: <http://www.excite.com>
- Goldfarb, F. C. (2001). "The XML Handbook", 3rd Ed., Prentice Hall
- Gomez-Perez, A., (1999). "Applications of Ontologies and Problem-Solving Methods", in *Workshop at Thirteenth Biennial European Conference on Artificial Intelligence*
- Gruber, R. T. (1993a). "A Translation Approach to Portable Ontology Specifications", *Knowledge Systems Laboratory Technical Report KSL 92-71*, Revised Apr. 1993
- Gruber, R. T. (1993b). "Toward Principles for the Design for Ontologies used for Knowledge Sharing", *The International Workshop on Formal Ontology*, March 1993
- Gruninger, M. and Lee, J. (2002). "Ontology Applications and Design", *Communications of the ACM*, Vol. 45, No. 2, pp.39-41
- Guarino, N. (2002). "Evaluating Ontological Decisions with OntoClean", *Communications of the ACM*, Vol. 45, No. 2, pp.61-65
- Hayes-Roth, B. (1995). "An Architecture for Adaptive Intelligent Systems", *Artificial Intelligence: Special Issue on Agents and Interactivity*, Vol. 72, pp.329-365
- Hendler, J. (2001). "Agents and the Semantic Web", *The IEEE Intelligent Systems Journal*, Mar./Apr. 2001, URL <http://www.cs.umd.edu/users/hendler/AgentWeb.html>
- Holsapple, W. C. and Joshi, D. K. (2002). "A Collaborative Approach to Ontology Design", *Communications of the ACM*, Vol. 45, No. 2, pp.42-47
- Kolodner, J. (1993), "Case-Based Reasoning", Morgan Kaufman Publishers, Inc., San Mateo, CA

- Konar, A. (2000). "Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain", CRC Press
- Leake, B. D. (1995). "Combining Rules and Cases to Learn Case Adaptation", in Proceeding of the Seventeenth Annual Conference of the Cognitive Science Society
- Liebowitz, J. (2001). "Knowledge management and its link to artificial intelligence", Expert Systems with Applications, Vol. 20, pp.1-6
- Lycos, URL: <http://www.lycos.com>
- Maes, P. (1995). "Artificial Life Meets Entertainment: Life like Autonomous Agents", Communications of the ACM, Vol. 38, No. 11, pp.108-114
- Russell, J. S. and Norvig, P. (1995). "Artificial Intelligence: A Modern Approach", Englewood Cliffs, NJ: Prentice Hall
- Smith, D. C., Cypher A. and Spohrer. (1994). "KidSim: Programming Agents Without a Programming Language", Communications of the ACM, Vol. 37, No. 7, pp.55-67
- Sowa F. J. (2000). "Ontology, Metadata, and Semiotics", Lecture Notes in AI #1867, Springer-Verlag, Berlin, pp.55-81
- Tsui, E., Garner, B. J. and Staab, S. (2000). "The role of artificial intelligence in knowledge management", Knowledge-Based Systems, Vol. 13, pp.235-239
- Turban, E. and Aronson E. J. (2001). "Decision Support Systems and Intelligent Systems", 6th Ed., Prentice Hall
- Walsham, G. (2001). "Knowledge Management: The Benefits and Limitations of Computer Systems", European Management Journal, Vol. 19, No. 6, pp. 559-608
- Watson, I. (1997). "Applying Case-Based Reasoning: Techniques for Enterprise Systems", Morgan Kaufmann Publishers, Inc.
- Watson, I. (2001). "Knowledge Management and Case-Reasoning: a Perfect Match?", in 14th International FLAIRS Conference, AAAI Press, pp.118-123
- W3C. (1999). "Resource Description Framework", URL <http://www.w3c.org/RDF/>
- W3C. (2000). "Extensible Markup Language (XML) 1.0 (Second Edition)", URL <http://www.w3.org/TR/REC-xml>
- W3C. (2002a). "Requirements for a Web Ontology Language", URL <http://www.w3.org/TR/2002/WD-webont-req-20020307/>
- W3C. (2002b). "Semantic Web", URL <http://www.w3.org/2001/sw/>
- Wooldridge, M. and Jennings R. N. (1995). "Agent Theories, Architectures, and Languages: a Survey", in Wooldridge and Jennings Eds., Intelligent Agents, Berlin, Springer-Verlag, pp.1-22
- Yahoo, URL <http://www.yahoo.com>

COPYRIGHT

Seung Hwan Kang and Dr. Sim Kim Lau © 2002. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.