# Enhanced Knowledge Management with eXtensible Rule Markup Language


Jae Kyu Lee and Mye M. Sohn
*Graduate School of Management*
*Korea Advanced Institute of Science and Technology*
*207-43 Cheongryang, Seoul 130-012, Korea*
*E-mail: jklee@kgsm.kaist.ac.kr; miae@kida.re.kr*



## Abstract

*XML has become the standard platform for structured data exchange on the Web. Next concern of Semantic Web is the exchange of rules in markup language form. The rules should be represented in such a way as to allow software agents to process and browse them for human comprehension. For this purpose, we propose a language* ***eXtensible Rule Markup Language (XRML)****. XRML is composed of rule identification, rule structure, and rule triggering markup languages. In XRML, a critical concern is how to extract the structured rules implicitly embedded in the Web pages and keep consistency between the two. By using the XRML, the Web based Knowledge Management Systems (KMS) can be integrated with rule-based expert systems. The advanced architecture with XRML can extend the application of KMS to automated form processing, preventive auditing, rule exchange and integration, and agent-based e-commerce.*


## 1. Introduction

HTML (Hypertext Markup Language) has made Web technology possible by providing the ability of browsing for human comprehension. However, software agents cannot understand the HTML files because general-purpose natural language processing capability is still very limited. To overcome the limit, Extensible Markup Language (XML) has been devised to explicate the implicitly embedded data in a formal structure with mutually agreed upon semantic definitions. A large number of industrial standard initiatives using the XML are under way including ebXML (Electronic Business XML Initiative). Other XML initiatives include OTP (Open Trading Protocol), OBI (Open Business on the Internet), CBL (Common Business Language), RosettaNet, eBis-XML, BizTalk, xCBL, and so on [21]. Some of these are expected to make up the standards of message exchanges for Business-to-Business (B2B) EC. The knowledge representation for Agent Communication also adopts the XML platform [9].

Our next concern is to process the rules implicitly embedded in the Web pages, which cannot be processed even with XML. So we need to represent the implicit rules in such a way as to allow software agents to process and browse them for human comprehension. We propose a language for such a purpose ***eXtensible Rule Markup Language (XRML)***. The topology of XRML can be contrasted with XML and HTML as shown in Figure 1 [16]. XRML should be transformable to XML and eventually to HTML.

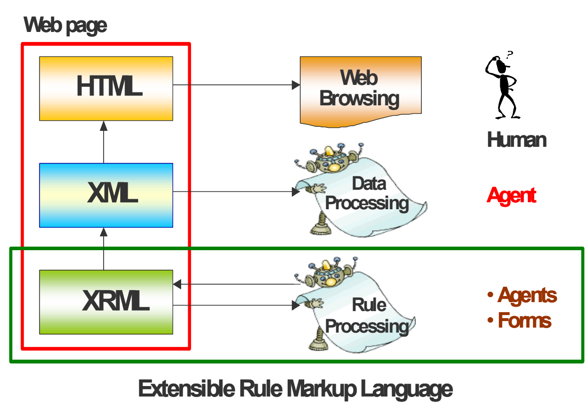


Figure 1. Topology of XRML

The remainder of this paper is organized as follows: Section 2 identifies components and objectives of XRML. Section 3 reviews the relevant researches in Rule Markup Language. Section 4 contrasts Knowledge Based System (KBS) with Knowledge Management System (KMS) and seeks an architecture of integrating them. Section 5 explains the syntax of Rule Identification Markup Language (RIML), Rule Structure Markup Language (RSML) and Rule Triggering Markup Language (RTML). Potential application areas of XRML are identified in Section 6, and Section 7 concludes with a summary and the potential contributions.


Acknowledgement: Earlier version of this paper is accepted to publish in the Communications of the ACM, forthcoming

## 2. Components and Objectives of XRML

### 2.1 Components of XRML

In order to fulfill the fore mentioned goal, the implicit rules embedded in the Web pages should be identifiable, interchangeable with structured rule format, and finally accessible by various applications. In this light, XRML requires the following three components:

(1) **Rule Identification Markup Language (RIML)**: The natural-language-like hypertexts specify the knowledge to browse to human. However the hypertexts need a meta-language that identifies the existence of implicit rules and their associations with the explicitly represented structured rules. RIML is a language that expresses such meta-knowledge about the rule existence and association with the formal representations.

(2) **Rule Structure Markup Language (RSML)**: The rules in the expert systems have to be represented in a formal structure to process with inference engines. Thus the implicit rules in the hypertexts should be transformed to the formal rule structure for automated inference. Since there is no formal clue for linking two representations directly, we need an intermediate representation which can be automatically transformable to structured rules and support easy generation and maintenance from the hypertexts. A language for such intermediate representation is called **Rule Structure Markup Language.**

(3) **Rule Triggering Markup Language (RTML):** RTML is a language that defines the conditions when certain rules will be triggered. So RTML is embedded in the applications such as Knowledge Based System (or expert systems), software agents, and forms in workflow management systems.

An illustrative architecture applying XRML to workflow management is depicted in Figure 2. In this application, the RTML embedded in forms can trigger the inference engine to use the rules generated from RSML. Note that humans can read hypertext on the browser, that XML statements are transformed to the database, and that RSML statements are transformed to rules in the rule base. The inference engine in the Knowledge Based System calls rules and data, and returns the inference results back to the inquiring software agents (workflow management system in this case). A challenging issue here is how to assist the extraction of RSML from hypertexts while maintaining consistency with RIML.

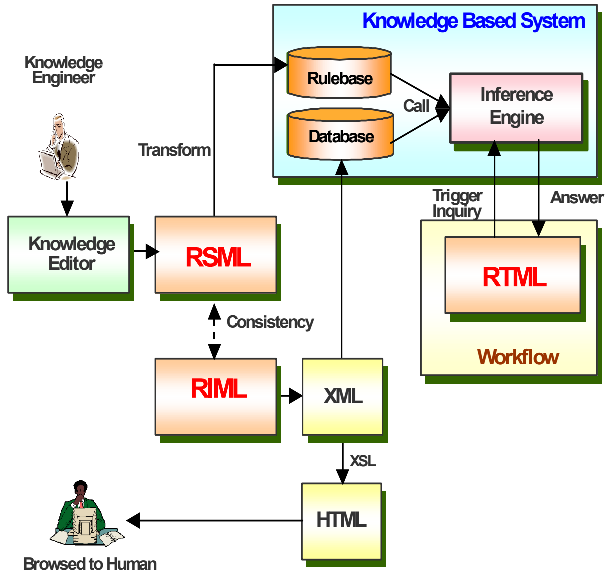


Figure 2. An Illustrative Architecture of XRML

### 2.2 Design Criteria of XRML

While we design the XRML, we need to pursue six ideal goals:

(1) **Expressional Completeness**: RSML should be completely transformable to a canonical syntax of structured rules. A syntax of a commercial tool may be regarded as the canonical rule. The rules in RSML form can be transformed to canonical form by mainly changing the semantic tags to explicit variable names (see Section 5.2).

(2) **Relevance Linkability**: Linkages of the relevance between *hypertexts with RIML,* and *rules in RSML* syntax *(*called *RSML rules),* should be completely expressed. Such linkages are possible because RIML identifies the rule title, variables and values, and the inverse relationship can be automatically generated.

(3) **Polymorphous Consistency**: Consistency should be maintained for knowledge expressed in different types of expressions, such as RSML rules and hypertext with RIML. Knowledge consistency between RIML and RSML can be semi-automatically assured by assisting the identification of their relationships through relevance linkages.

(4) **Applicative Universality**: The rule expressions in RSML should be able to support multiple applications which embeds RTML within the domain universe. This can be realized by transforming the RSML to various rule syntaxes in applications.

(5) **Knowledge Integrability**: Structured rules collected from multiple sources including the rules from RSML should be integrated uniformly. The Integrability is essential for rule exchanges.

(6) **Interoperability**: Rules in RSML should be exchangeable and sharable among multiple commercial solutions. This can be realized if RSML can be transformed to the rules in major expert system tools.

## 3. Initiatives in Rule Markup Language Research

There is emerging research in rule markup languages although the research focus varies. Rule Markup Languages are described in [3], but the goals of XRML have evolved further to attain the design goals described above. An early version of XRML was presented in [13].

- **Business Rules Markup Language (BRML)**: It specifies a common rule structure to exchange rules between heterogeneous rule-based systems.
- **Agent-Object-Relationship Markup Language (AORML)**: It describes business rules to process with software agents, including the business process, interaction process, sequence of events, actions, activities, and control.
- **Universal Rule Markup Language (URML)**: It represents the input/output data of AI applications in XML for reducing conversion efforts and time.
- **Artificial Intelligence Markup Language (AIML)**: It is XML specification for Artificial Linguistic Internet Computer Entity (ALICE) using a simple pattern-matching technique.
- **Case Based Markup Language (CBML)**: This is an XML-based case representation language to achieve interoperability and flexiblilty of case reuse.
- **Relational-Functional Markup Language (RFML)**: This is an XML version of Relfun, which is a logic programming language that uses call-by-value expressions.

Semantic Web community pursues to express various knowledge representations to comply with XML such as RDF and OIL [5, 10, 12]. XRML however not only expresses and exchanges rules but also identifies the existence of rules and provides a basis to keep consistency between human and software agents.

## 4. Knowledge Based Systems and Knowledge Management Systems

To explain the necessity of XRML, let us review the concept and status of two relevant disciplines: Knowledge Based Systems and Knowledge Management System. They are similarly termed, but have different roots in practice. The distinction of these two systems are contrasted in Table 1.

Table 1. Knowledge Based Systems and Knowledge Management Systems

| System Title | KBS | KMS |
|---|---|---|
| **Knowledge Processor** | Inference Engine | Human |
| **Inference** | Forward/Backward Chaining, Approximate Reasoning | HyperLink, Keyword Search |
| **Knowledge Representation** | Rule, Predicate Calculus, Object (Frames) | HTML, XML, VRML |
| **Scope of Knowledge** | Specific Knowledge Base | Global Knowledge Portal; Corporate Portal |
| **Tools** | Domain Specific | General Purpose |
| **Popularity** | Limited to the Application Users | Everybody, Everyday |

### 4.1 Knowledge Based System (KBS)

The KBS, also called the Expert System, stems from Artificial Intelligence, and has been popular since early 1980s [17]. The main goal of KBS is the automatic inference of coded knowledge. The natural language understanding is a key part of knowledge processing, but its success is limited yet. Therefore, for knowledge processing, practical KBSs use only structured knowledge representations such as rules, predicate calculus, and objects (or frames), and tailored inference engines. Because of its limited capability in common sense reasoning, the applications are developed for specific domains such as diagnosis, configuration, manufacturing planning, and managerial decision aids. Knowledge acquisition and maintenance have been the hurdles for justifying the implementation.

Recently, KBS has expanded its platform in the Web environment. Typical applications include intelligent e-mail interpretation and classification, smart advisory about products for customer services and training, online configuration, and help desk for technical support [6]. Knowledge acquisition is explored from Web pages [8], and a tool has been developed to automatically generate the hypertext structure, which works like a rule-based system [25]. The tool market for KBS is coming back with a new term *Business Rule Engine* [23].

### 4.2 Knowledge Management System (KMS)

On the contrary, the KMS refers to a class of information systems applied to creating, transferring and managing organizational knowledge [1]. With a wider use of Internet and Web, knowledge management has become an increasingly important issues. Two primary approaches to capture the organizationl knowledge are the *network model* and the *repository model* [26]. A comprehensive list of knowledge management frameworks are summarized in [7, 22].

Although there are various features in knowledge management systems, what we focus here is the knowledge repositary model - the Web based KMS whose primary goal is retrieving from the large scale Web based documents and databases in contrast with the KBS that infers using rules [18]. In this architecture, the primary targeted users of KMS are not software agents, but humans who desire comprehension with interactive search. Technically speaking, any knowledge on the Web (or any other storage structures such as a database) can be within the scope of a KMS application. The application area of KMS is thus very general, and can be used widely. So far knowledge management research has exploited the issues of knowledge sharing and reusing mainly from the managerial and motivational point of view.

Corporate portal, under a bright spotlight nowadays, is a new trial to accomplish knowledge sharing and reusing. Corporate portal makes it possible for inner and outer users of corporation to search, manipulate and share electronic resources including documents, enterprise application, e-business services and information from the Internet, stored in the corporate database.

### 4.3 Convergence of KBS and KMS with XRML

Convergence of KBS and KMS is inevitable because knowledge should be shared by both humans and software agents [17, 20]. In fact, this is exactly the goal that XRML is pursuing. To meet this end, it is necessary to keep consistency between the hypertext knowledge in KMS and the structured rules in KBS. Thus, this is one of the key research issues in XRML. In this regard, *XRML is a framework for integrating KBS and KMS.*

Generating RSML rules from hypertext can be regarded as a process of *knowledge extraction*, and generating meta-knowledge on the relationships between the hypertexts and RSML rules (regarding which hypertext is related to which RSML rules, and vice versa) as a process of *meta-knowledge extraction*. As mentioned earlier, knowledge acquisition from a variety of sources is in general very expensive. But knowledge extraction from existing hypertexts is a not so much a social issue as a technical one, and thus can be cost effective.

A sea of the hypertext knowledge is already coded in Markup Language form on the Internet. This means that the cost for application of XRML can be easily justifiable and that its impact can be enormous. XRML can be not only the next step for KBS and KMS, but also a rule markup language for the Semantic Web, which the Web community is pursuing [2, 4].

## 5. Syntax of RIML, RSML and RTML

Let us investigate the syntax of RIML, RSML, and RTML with examples. Full syntax of XRML 0.5 can be found in [24].

### 5.1 Rule Identification Markup Language

Suppose there are two browsed paragraphs in HTML that describe the regulations about research budget expenditure.

*<HTML>*
*<p>A research account can be spent only within the limit of the contract budget, according to the following restrictions.</p>*
*<p>If the budgetary source is the type-P research fund, the spendable items are limited to on student's salary and expenses for data collection.</p>*
*</HTML>*

The second paragraph includes an implicit rule which can be explicitly expressed as follows:

*Rule Title: Restriction of Type-P Research Fund Expenditure*

*IF ((budgetary source IS type-P research fund)*
*AND ((spendable item IS student's salary)*
*OR (spendable item IS expense for data collection)))*
*THEN expenditure IS permitted*

Even though the two types of expressions imply same regulation, it is not easy to identify the relationship between them. So we need to add meta-knowledge on how the hypertext is related with the structured rule, as underlined in the following example.

```
<HTML>
<p>A research account can be spent only within the limit of the contract budget, according to the restrictions.
</p>

<RIML>
<Rule>
<RuleTitle> Restriction of Type-P Research Fund Expenditure </RuleTitle>
<p>If the <variable1>budgetary source</variable1> is the <value1>type-P research fund</value1>, the <variable2>spendable items</variable2> are limited to on <value2>student's salary</value2> and <value2>expenses for data collection</value2>.</p>
</Rule>
</RIML>
</HTML>
```

In this simple example, the section related to the structured rules is delineated by <RIML>…</RIML>. The rule and its title are identified by <Rule>…</Rule> and <RuleTitle>…</RuleTitle>. The tags <variable#> and <value#> identify the variables and values used in the structured rule. The same numbers in the tags imply the association between a variable and a value. The HTML/RIML can be transformed to the original HTML file by eliminating RIML statements in this simple example. The transformation process can become complex as we employ more RIML commands.

Such tags are extensible if we need to identify further detail. For instance, simple algebraic operators such as GreaterThan (GT) or LessThan (LT) can be added. More sophisticated and domain specific tags allow easier comprehension of the relationships, but take more knowledge editorial efforts. So we need to balance the sophistication of RIML with the effort of transforming to RSML. The tags may be abbreviated to shorten terms such as *vr* (for **v**a**r**iable) and *vl* (for **v**a**l**ue). The HTML/RIML editor should help the process of editing the hypertext along with its meta-knowledge.

### 5.2 Rule Structure Markup Language

What we need now is the intermediate representation of rules specified in RSML which can be easily associated with the RIML. Note that the variables are transformed into tags in XML syntax with their values within the paired tags. Rules in this syntax can be directly matched with the data in the XML file. This is the big advantage of RSML.

```
<RSML>
<Rule>
<RuleTitle> Restriction of Type-P Research Fund
Expenditure </RuleTitle>
<IF>
  <AND>
    <budgetary source>type-P research fund
    </budgetary source>
    <OR>
        <spendable item>student's salary
        </spendable item>
        <spendable item>expense for data collection
        </spendable item>
    </OR>
  </AND>
</IF>
<THEN>
 <expenditure>permitted</expenditure>
</THEN>
</Rule>
</RSML>
```

Note that the variables (in the tags) and their values in RSML are the same as the words identified in RIML. With the definitions in RIML, the RSML editor can generate a crude shape of the rules by assigning the key words to corresponding slots of variables and values. When we need to revise rules in the RSML, the same relationship can be traced in the reverse direction identifying which paragraphs and words are associated with these rules. In this manner, we can assist the consistency maintenance between RIML and RSML. A thesaurus of synonyms, the plausibility of association among variables, and that between variables and values about the application domains, will ensure that knowledge edition and maintenance are easier and more accurate. Developing the consistency maintenance aids is a challenging opportunity. Details on this issue are under research [15].

RSML statements can be transformed to canonical rules by modifying the reserved words of RSML. To make editing RSML more automated, a knowledge engineer needs to specify more meta-knowledge in the RIML stage. For instance, if the association knowledge between a variable and a value is specified in RIML, a statement with the variable and the value can be automatically generated as mentioned earlier. So the total amount of effort needed for knowledge management will be decided by how both RIML and RSML are generated and maintained.

### 5.3 Rule Triggering Markup Language

RTML is a language embedded in the application programs such as forms in workflow management, software agents, and broadly speaking in any program. So what we have to define at this point is a set of standard statements about when to trigger the inference, which rules to use, how to use the obtained result, and so on. The RTML tags here are useful to identify the relevant tags in RIML, RSML, and data files in the XML format.

See the following example that is assoicated with the Research Fund Account.

```
<RTML>
<WhenTrigger>
   <AND>
      < requisition>on</requisition>
      <budgetary source>type-P research fund
      </budgetary source>
  </AND>
</WhenTrigger>
<Bring>
    <RuleTitle>Restriction of Type-P Research Fund
   Expenditure</RuleTitle>
   <DataFile>Research Fund Accounts</DataFile>
</Bring>
<Result>
    <expenditure>permitted</expenditure >
</Result>
</RTML>
```

The tag *<WhenTrigger>* specifies the condition of rule triggering; *<Bring>* brings the relevant rules and data to the inference engine; *<Result>* returns the inference result as the value of the tag. The application programs probably written in Java could call up the returned results.

## 6. Applications of XRML

XRML can be applied to a broad spectrum of Knowledge Based Systems on the Web, and it can make Knowledge Management Systems more intelligent. Typical examples include automated form processing, preventive auditing, knowledge exchange and integration, and agent-based intra-/inter-organizational e-commerce. An application of XRML in the workflow environment was depicted in Figure 2.

- ***Automated Form Processing*: Object-oriented forms equipped with the RTML can trigger inquiries for automatic approval of routine and frequent tasks such as business trip reimbursement and small acquisitions. This function can be effectively integrated with the emerging desk-top purchase in which the requisitioner bypasses the approval process and procurement department [14]. The hypertext used by forms can also be visualized for the requisitioners on the Web.
- ***Preventive Auditing***: Certain activities need auditing by authorities. If audit knowledge is implemented in XRML, the knowledge can be visually displayed to the inquirers, and can automatically stamp the approval of auditors. Auditors may focus on knowledge maintenance rather than audit transactions.
- ***Knowledge Exchange and Integration:*** The rules in different organization can be shared by exchanging the XRML rules. The transmitted rules can be displayed to receivers, integrated with the existing rules in the receiver's site, and processed by receiver's agents. Regulations by government and corporates could be integrated with the architecture depicted in Figure 3 [11].

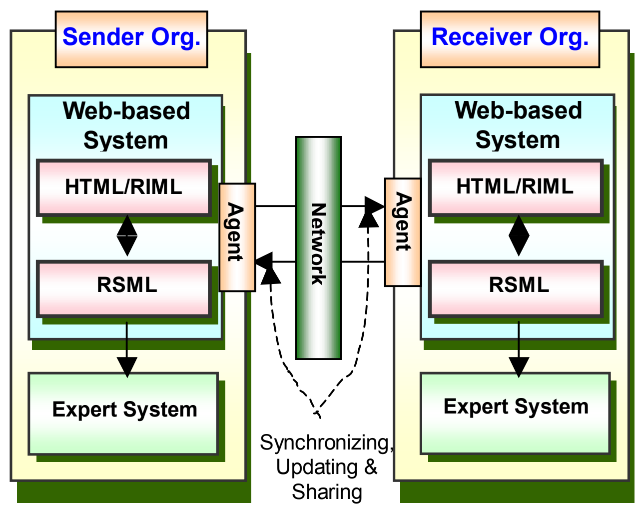


Figure 3. Inter-organizational Knowledge Sharing with XRML

- ***Agent based Intra-/Inter-Organizational EC*:** During B2B transactions and collaboration, knowledge about products and services, and contract terms and conditions can also be requested by software agents. XRML is particularly necessary at the call centers because the human agents in the call center are not really experts to the inquiries. However, they can be trained to introduce the relevant knowledge based system to the customers. Synchronized Web browsers between the human agents and customers can help the communication more effectively [19].

## 7. Challenges – The Opportunity

To implement XRML in the real world applications, there are several challenges that we have to overcome, which are more an opportunity than a hurdle.

- ***Consistency Maintenance of Polymorphic Knowledge Representations:*** The same data and rules may exist in a relational database, HTML and XML files, RSML, structured rules and even program codes. So when one type of knowledge or data changes, consistency among them should be maintained. Consistency in XRML between HTML/RIML, RSML, and XML is particularly important. Meta-knowledge can support this process.
- ***Domain-Specific Thesaurus:*** RIML can start with a natural-language-independent syntax. Understanding the syntax of a particular natural language such as English or Korean will be helpful in identifying the relationship between variables and values. In addition, to support the frequently used domains such as online customer supports for electronic products, we can *a priori* define the relationship among the vocabulary items.
- ***Multi-URL Based Inference*:** In the earlier example, we have used one Web page to infer a certain issue. However, an inference may require more than one Web page. To handle this issue, the RSML rules need to keep the information about the URL as well as the rules in each URL. This requires an extension of the tags in XRML.
- ***Integration of Rules from RSML with Other Sources:*** When inquires by RTML require knowledge not only from RSML but from other sources, the RTML should be able to identify the rules from the all necessary sources. If all types of rules are transformed to a canonical structure in advance whatever the initial form was, the problems at execution time can be avoided easily. However, this approach shifts the integration effort to the knowledge maintenance stage.

## 8. Concluding Remarks

We have designed the XRML version 0.5 as illustrated above, and developed its prototype named Form/XRML which is an automated form processing for disbursement of the research fund in the Korea Advanced Institute of Science and Technology (KAIST). Since XRML allows both human and software agent to use the rules, there is huge application potential. We expect that XRML can contribute to the progress of Semantic Web platforms making knowledge management and e-commerce more intelligent. Since there are many emerging research groups and vendors who investigate this issue, it will not take long to see XRML commercial products. Matured XRML applications may change the way of designing information and knowledge systems in the near future.

## References


[1] Alavi, Marayam and Dorothy E. Leidner, “Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues,” *MIS Quarterly,* Vol. 25, No. 1, March 2001.

[2] Berners-Lee, T. Hendler, J. and Lassila, O., “The Semantic Web,” *Scientific American*, May 2001.

[3] Boley, H. and Tabet, S. “Rule Markup Langugae,” *http://www.dfki.uni-kl.de/ruleml*, 2001.

[4] Boley, H., Tabet, S. and Gerd Wagner, “Design Rationale of RuleML: A Markup Language for Semantic Web Rules,” *Proc. SWWWS ’*01, Stanford, July/August 2001.

[5] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L. and Patel-Schneider, P.F., “OIL: An Ontology Infrastructure for the Semantic Web, ” *IEEE Intelligent Systems*, Vol. 16, No. 2, March-April 2001, pp. 38 –45.

[6] Firepond, “eServicePerformer Concierge,” *http://www.firepond.com/SPconcierge.pdf*, 2001.

[7] Fischer Gerhard and Jonathan Ostwald, “Knowledge Management: Problems, Promises, Realities, and Challenges,” *IEEE Intelligent Systems*, Jan/Feb. 2001, pp. 60-72.

[8] Gaines, R.B. and Shaw, L.G.M,. “Knowledge Acquisition, Modeling and Inference through the World Wide Web,” *CPCS/KSI/KSS Report*, 1997.

[9] Grosof, N. B., “Standardzing XML Rules: Rule for E-Business on the Semantic Web (Invied Talks),” *Workshop on E-Business and the Intelligent Web*, IJCAI-01, Aug 2001.

[10] Hendler, J. “Agents and the Semantic Web”, *IEEE Intelligent Systems*, Vol. 16, No. 2, March-April 2001, pp. 30 – 37.

[11] Kim W. T, J. K. Lee and J. Y. Kang, “Knowledge Sharing System between Government and Company using XRML,” *The Korea Society of Management Information System*, 2002, pp. 706-715 (Korean).

[12] Klein, M. “XML, RDF, and Relatives”*, IEEE Intelligent Systems*, Vol. 16, No. 2, March-April 2001, pp. 26 –28.

[13] Lee, J. K., "Artificial Iintelligence in Electronic Commerce,.". *Plenary Speech at Pacific Rim International Conference on Artificial Intelligence,* Melbourne, http://www.icec.net, August 2000.
[14] Lee, J. K. and Efraim, T. Planning B2B E-Procurement Marketplaces. eds. Lowry, P.B., Cherrington, J.O. and Watson, R.R., **"**Handbook of Electronic Commerce in Business and Society," *CRC Press LLC*, 2001.
[15] Lee, J. K. and Kang J., "Knowledge Acquisition from Web Pages using XRML," *Working Paper (KAIST),* 2002.
[16] Lee, J. K. and Shon M., "eXtensible Rule Markup Language - Toward the Intelligent Web Platform," *Communicatiuons of the ACM*, 2002 (forthcoming).
[17] Liebowitz, J., "Knowledge Management: Learning from knowledge Engineering," *CRC Press LLC*, 2001.
[18] Martin Phillippe and Peter W. Eklund, "Knowledge Retrieval and the World Wide Web," *IEEE Intelligent Systems*, May/June 2000, pp. 18-25.
[19] Moon B. K. and J. K. Lee, "A Next Generation Multimedia Call Center for Internet Commerce: IMC," *Journal of Organizational Computing and Electronic Commerce*, vol. 10, no.4, 2000, pp. 227-240.
[20] O'Leary, E. D., "Using AI in Knowledge Management: Knowledge Management and Ontologies," *IEEE Intelligent Systems*, May/June 1998, pp. 34-39.
[21] Robin Cover, "The XML Cover Pages–XML Articles and Papers," *http://xml.coverpages.org/XMLArticles.html,* 2002.
[22] Rubenstein-Montano, B., J. Liebowitz, J. Buchwalter, D. McCaw, B. Newman, K. Rebeck, and The Knowledge Management Methodollogy Team, "A System Thinking Framework for Knowledge Management,", *Decision Support System*, 2001, pp. 5-16.
[23] Sinur J., "The 2002 Business Rule Engine Market Magic Quadrant," *Gartner Research*, Feb. 2002, pp. 1-5.
[24] Sohn, M. "XRML version 0.5," *http://xrml.kaist.ac.kr*, 2001.
[25] Song, Y. U. and Lee, J. K., "Automatic Generation of Web-based Expert Systems," *Journal of Intelligent Information Systems*, 2000, pp. 1-16 (Korean).
[26] Turbn Efraim, E. McLean and J. Wetherbe, "Information Technology for Management," *John Wiley & Sons, Inc.*, 2002, p. 369.