Extensible Rule Markup Language (XRML) – Version 0.5

2001. 10.

IIS/ EC Lab

K A I S T

1. Introduction

1.1 Purpose

This document represents a specification for XRML language design to achieve the knowledge sharing between human and software agent. This language will be used as a basis for concrete implementations of the functionality described in the XRML paper.

1.2 Scope

It is the intention of this specification to describe a language that be used to represent the implicit rules in such a way as to allow software agents to process them as well as to be comprehensible to human.

1.3 Audience

This specification is intended for use by students, researchers or developers with Dr. Lee's permission. Furthermore, it will be value to those concerned with the develop XRML tool.

1.4 Document Status

This document is an official publication of the IIS/EC Laboratory in KAIST, representation version 0.5 of the XRML specification. It may be obtained via the Internet from: http://www.icec.net, or by e-mailing a request to: jykang@kgsm.kaist.ac.kr.

1.5 Document Conventions

- Extracts for DTD are provided in line with descriptive text. These extracts are intended to highlight the particular markup constructs used throughout this specification. They will inside a box as in the following example:

```
<!ELEMENT foo (bar)>
<!ELEMENT xxx (baz)>
```

1.6 Approach

At high level, these are the goal of this specification:

- Support knowledge sharing between human and Software agent
- Maintain consistency between natural language and structured rule
- Design a light, easy-to-understand tags

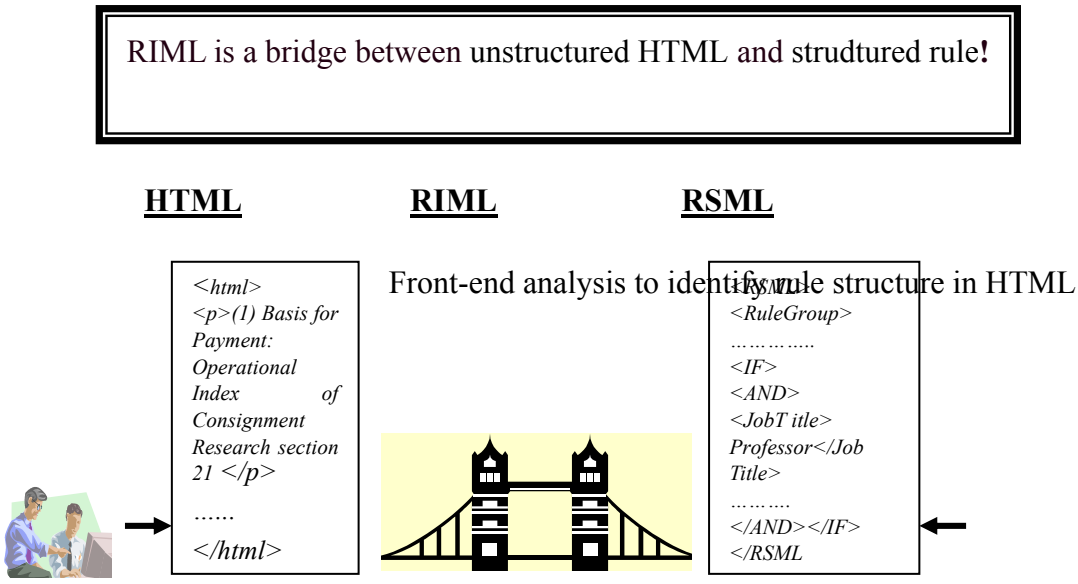
In order to achieve these goals, we propose a new language named Extensible Rule Markup Language (XRML) which is an extension of XML with additional capabilities of representation of rule structure, automatic knowledge processing by exchanged rule, and consistency maintenance of knowledge. To achieve the capabilities of XRML, we propose three components: the Rule Structure Markup Language (RSML), the Rule Identification Markup Language (RIML), and the Rule Triggering Markup Language (RTML).

2. Technical Specification

2.1 RIML

Software agent executes structured rule based knowledge processing, and human uses hypertext in natural language format for knowledge processing. However, it takes the circumstances into consideration that most of knowledge in Web is described in unstructured HTML files in a natural language, software agent can scarcely knowledge processing. It is an obstacle toward Intelligent Web.

To overcome this obstacle, we propose RIML. RIML is a bridge between unstructured HTML and strudtured rule



Identification of relevant aspect of rule structure
including titles, variables and values

2.1.1 DTD

DTDs provide critical information that allows XML processors to parse the coding and make certain that it contains all the information the application needs, in a form the application is prepared to accept. The DTD provide a critical link between the data files given to the XML processor and the data that are transmitted from the XML processor to the application. DTDs help computer understand structures that may seem obvious to human. The following DTD segment defines the top-level structure of RIML document.

```
<!ELEMENT RIML (Rule+)>
   <!ATTLIST RIML Version CDATA #Required)>

<!ELEMENT Rule (RuleTitle, (variablei*, valuei*)>

<!ELEMENT RuleTitle(#PCDATA)>
<!ELEMENT value(#PCDATA)>
<!ELEMENT variable(#PCDATA)>
```

2.1.2 Elements and Attributes of RIML

This element carries a required attribute (Version) that indicates version of the specification with which the document conforms. Most element declarations will contain list of element, setting rules for which elements are required, the sequence in which they appear, and how many times they may appear. Parameter entities may also appear in the list, making easier for developers to create multiple similar structures. Table 1 list a few symbols to provide rules and using elements.

<Table 1> Symbols for specifying element structure

Symbol	Symbol Type	Description	Example	Example Notes
|	Vertical bar	Indicates or condition	thisone|thatone	Either thisone or thatone must appear
,	Comma	Requires appearance in specified sequence	thisone, thatone	thisone must appear, followed by thatone
?	Question Mark	Makes optional, but only one may appear	thisone?	thisone may appear
	No symbol	One, and only one, must appear	thisone	thisone must appear
*	Asterisk	Allows any number to appear, even zero	thisone*	thisone may be present; multiple appearance of thisone are acceptable
+	Plus sign	Requires at least one to appear, but may appear multiple times	thisone+	thisone must be present; multiple appearance of thisone are acceptable
()	Parentheses	Group elements	(thisone|thatone) whichone	Either thisone or thatone must appear, followed by whichone

4.2.2 Elements and Attributes of RIML

The root element within a HTML/RIML document is named “RIML”. Each RIML document contains at least one of “Rule”. Another elements defined in RIML have the following meaning:

<Table 2> Meaning of elements and attributes in RIML

Element/Attribute	Description
Rule	The element "Rule" envelops the body of rule. The presence of this element indicates that this is contains at least one variable, value, and operator, etc. At least one must be present within RIML.
RuleTitle	It is the title of the Rule. It can be left out should a specific rule be not included in rule group.
variable*i*	The literal element "variable" represents variable names, where i=1, 2,….., n.
value*j*	It is the literal element that represents value names, where j=1, 2,….., m.

After the element named, an attribute definition or a list of attribute definitions may follow. A definition consists of the name of an attribute, its type, and its default value (or a specification for that value). Names of attributes must obey the same rules as names for entities and elements. Attribute types are quite unlike the structures explored so far and define the kinds of data permitted in attribute when used in an element source. In our example, attribute has RIML as an element name, Version as an attribute name, and CDATA as a type. The value, "#Required", indicates to the parser that attribute "Version" must have a value in all instances of the element.

2.1.3 An Example

```
<html>
<title>KAIST Research Info.</title>

<RIML Version="0.5">

  <p>Execution of Research Fund </p>
      <Rule>
        <p>A. <RuleTitle>Payment of Consignment Research Fund</RuleTitle>
        </p>
          <p>(1) Basis for Payment: Operational Index of Consignment Research
      section 21 </p>
……
          <p>(3)Payment Scope</p>
              <p>(A) <value1>Professor</value1>
               <ul><li>Less than <value3>standard payment *1.5</value3> with
                 regard to number of researches </ul>
        <p>* <variable2>Sponsored</variable2> project by <value2>Ministry of
              Science and Technology</value2> or < value2>Ministry of Information
              and Communication</value2> is exclude.</p>

              <p>(B) Student
                <ul type="circle">
                      <li><value1>Bachelor Course</value1>: Less than
                      <value3>$200</value3>
                      <li><value1>Master Course</value1>: Less than
                      <value3>$400</value3>
                      <li><value1>Ph. D. Course<value1>: Less than  <value3>
                      $600</value3>
                      <li><value1>Greater equal 5 years at Ph. D. Course<value1>
                      Less than <value3>$1,000</value3></ul>
      </Rule>
  </RIML>
</html>
```

2.2 RSML

2.2.1 DTD

What we need now is the intermediate representation of rules which can be automatically transformed to structured rules and easy to associate the RIML. The RSML in overall XRML architecture supports the software agent by passing the rule that KBS will use for inference. It performs an important role to maintain the consistency between web page and rulebase by tight coupling with the RIML type rules. To do so, the existing rule can be expressed in the form that can be exchanged on the Internet. In this paper suggests RSML as a schema that can present the structuredness of the rule on the Internet. RSML is defined as a XML that can present the structure of condition-action type rule. To maintain the rule structuredness, we suggested hierarchical DTD that is suitable for representation of RSML. The following DTD segments represent the structure of the RSML rules.

2.2.2 Elements and Attributes of RSML

An RSML rule is an XML document whose root element type is RSML. The meaning and hierarchy of elements and attributes defined in RSML have summarized Table 3 as follows.

<Table 3> Meaning of elements and attributes in RSML

Element/Attribute	Description
RSML	Root element of RSML
Rule	The element envelops the body of rule. At least one must be present within RSML. Rule encompasses elements such as RuleTitle, IF, and THEN
RuleTitle	It is the title of individual rule included in Rule group. It is the identifier of RSML rule.
IF/THEN	The element represents structure of Rule. It uses the condition-action type representation. It can either use or omit operators like AND and OR but it must have at least one variable, value and operator, which represents relationship between the variable and value.
AND/OR	These have the same operator as their names, and operator, which represents relationship between variable and value as low hierarchical elements.
variable	The literal element represents variable names.
value	It is the literal element that represents value names.

2.2.3 An Example

```
<RSML Version="0.5">
  <Rule>
      <IF>
        <AND>
          <Job Title>Professor</Job Title>
          <NOT>
            <OR>
              <Sponsord>Ministry of Science and Technology</Sponsord>
              <Sponsord>Ministry of Information and Communication
              </Sponsord>
            </OR>
          </NOT>
        </AND>
      </IF>

      <THEN>
        <LE/>
          <Payment Level>Standard Payment Level*1.5</Payment Level>
      </THEN>

          <!-- The other rules are represented in the Appendix    -->

    </Rule>
</RSML>
```

2.3 RTML

2.3.1 DTD

RTML is a language embedded in the application programs like forms in workflow management, software agents, and broadly speaking in any program. So what we have to prepare at this point is defining 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.

To fulfill the goals of the RTML, we have to prepare a set of standard statements about when to trigger the inference, which rule to use, how to use the obtained result, and so on. So we attempt to represent the attributes in grammatical form, which represents not only relevant rule, but also returned result that is selected among inference results. To do so, we suggest variables and values that describe attribute types. The DTD of RTML is represented as follows:

```
<!ELEMENT RTML (WhenToTrigger, Bring, Result)+>
        <!ATTLIST RTML Version CDATA #Required)>

<!ELEMENT WhenToTrigger (AND|OR|NOT)>
<!ELEMENT NOT (AND|OR)>
<!ELEMENT AND (budgetary_source, requisition)>
<!ELEMENT OR (budgetary_source, requisition)>
<!ELEMENT budgetary_source (#PCDATA)>
<!ELEMENT requisition (on|off)>
<!ELEMENT on (#PCDATA)>
<!ELEMENT off (#PCDATA)>

<!ELEMENT Bring (RuleTitle, DataFile)>
<!ELEMENT RuleTitle (#PCDATA)>
<!ELEMENT DataFile (#PCDATA)>

<!ELEMENT Result (expenditure)>
<!ELEMENT expenditure (permitted|denied)>
<!ELEMENT permitted (#PCDATA)>
```

2.3.2 Elements and Attributes of RTML

The root element within an RTML rule is named RTML. The meaning and hierarchy of elements and attributes defined in RTSML have summarized as follows.

- WhenToTrigger: This element specifies the inference triggering timing. It may contain following sub elements: AND, or OR.
- Bring: The element specifies the rule to be used for inference. It should contain following sub element: <RuleTitle> and <DataFile>.
- Result: It specifies how to use the obtained result. In this example, inference result is used salary expenditure.
- AND/OR: This element represents relationship between sub elements. This element should contain the following sub element: requisition and budgetary_source.
- requisition: This element represents statue of process. It has ON and OFF as a value.
- Budgetary_source: It is the literal element that represents type of budgetary to be expenditure.
- ON/OFF: It is the literal element that represents requisition status.
- RuleTitle: It is the title of individual rule to be used by intelligent agent.
- DataFile: It is the title of data file to be used by intelligent agent.
- expenditure/ budgetary_source: The literal element

2.3.3 An Example

```
<RTML Version="0.5">
  <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>
```