Semantic Web with XRML

한국지능정보시스템 학회

May 31, 2003

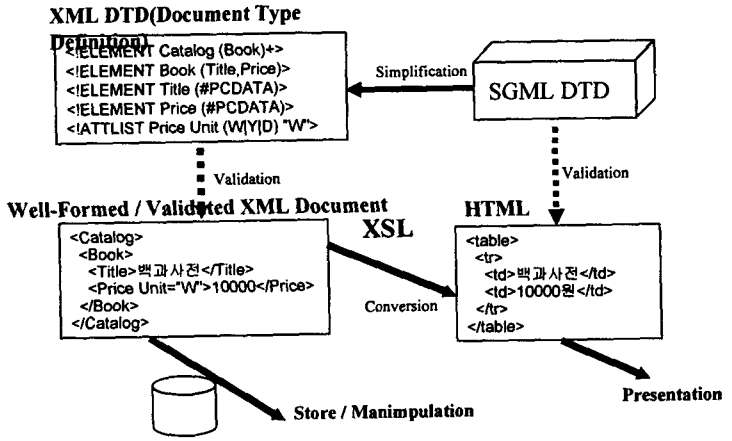
이 재규; **jklee@kgsm.kaist.ac.kr**

Graduate School of Management

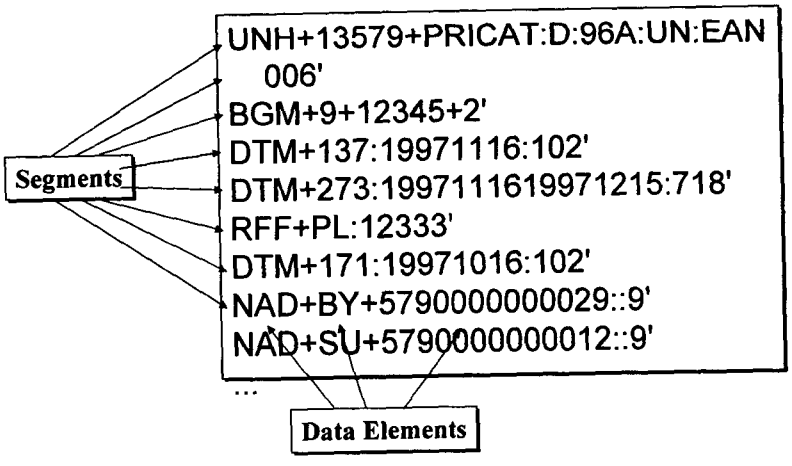
Korea Advanced Institute of Science and Technology

Table of Contents

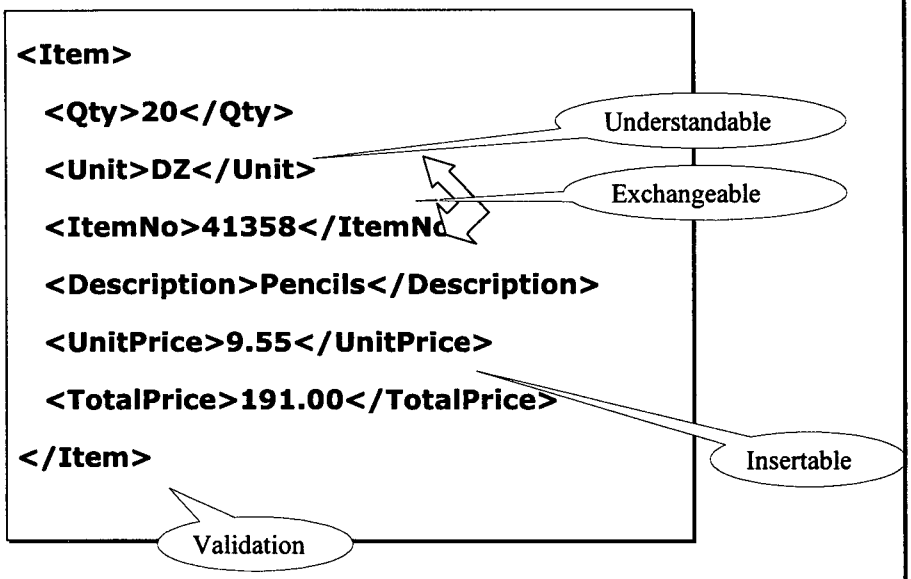
I. HTML과 XML



EDI Message



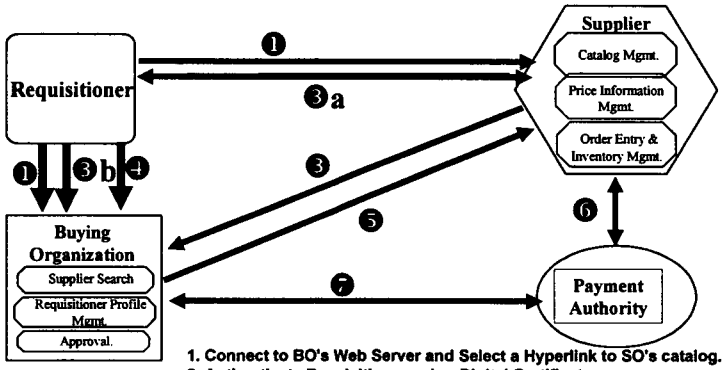
XML/EDI 메시지



E-Commerce Protocols with XML

- ACL in XML Form
- XML/EDI (www.xmledi.com)
- Biz Talk (Microsoft)
- OBI (Open Business Interface)
- CBL (Common Business Language)
- OTP (Open Business Protocol)
- ACML (Agent Communication Markup Language)
- ebXML: EDIFACT Initiative

OBI (Open Buying on the Internet)



1. Connect to BO's Web Server and Select a Hyperlink to SO's catalog.
2. Authenticate Requisitioner using Digital Certificate
3. OBI Order Request
4. Add Administrative Information
5. OBI Order
6. Obtain Credit Authorization
7. Issue Invoice and Receive Payment

XML-based B2B Standards

- Business Interface Definitions (BIDs)
- Common Business Library (CBL) includes XML message templates for the basic business forms used in ANSI X12 EDI (www.xmledi.com) transactions as well as the emerging Internet specifications as OPT and OBI.
- CBL facilitates spontaneous commerce between trading partners without custom integration or prior agreement on specific industrywide standards.
- OTP (Open Trading Protocol): Payment, receipt, delivery, and customer support. www.otp.org
- OBI (Open Buying on the Internet): By Americal Express and major buying and selling organization (www.openbuy.com)

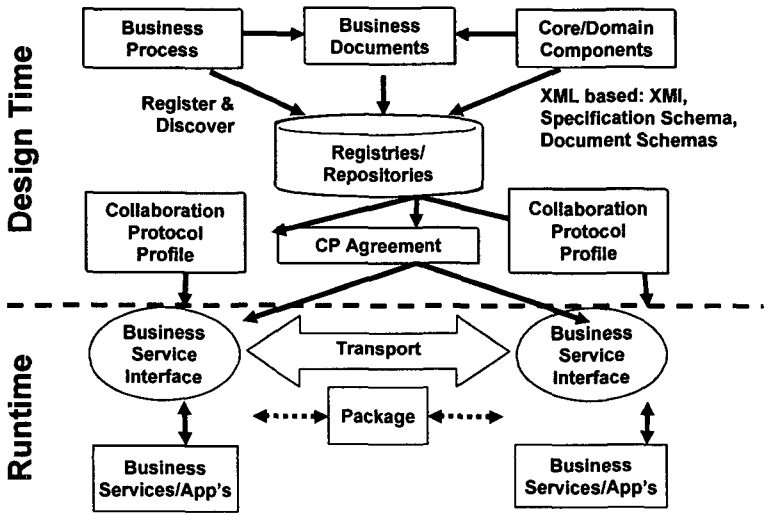
XML-Based B2B Standards

- These specifications are mapped to each other using a dictionary of common business terms and data elements.
- Ontology standard necessary.
- Advantages of XML: Extensibility, less sophisticated than logic but easy to use and ubiquity.

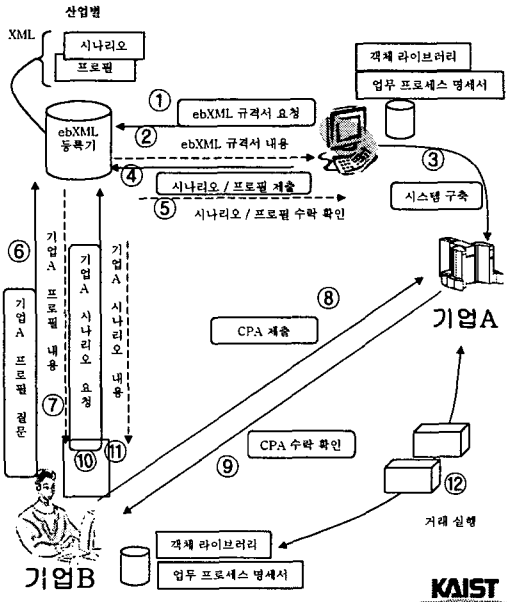
- Domain Specific Commerce Languages:

- ❖ RossetaNet(www.rosettanet.org) for PC industry;
- ❖ Information and Content Exchange (www.w3.org/TR/1998/Note-ice-19981026, CNET initiative);
- ❖ Open Financial Exchange (www.ofx.net)

ebXML Architecture



ebXML의 개념적 처리 절차



Web Service 의 정의

➢ 언제, 어디서나, 어떤 기기에서나, 원하는 정보 (예: 일반 Contents),응용 기능 (예: SAP PDM 모듈) 또는 서비스 자체 (예: 세금납부)를 제공해 주는 총체적 서비스
(자료:삼성 정보기술 연구소 기반기술팀 웹 서비스 센터)

➢ 공개적인 네트워크 및 관련 표준을 통해 단일한 기업 내부 또는 다수의 기업간에 기존의 어플리케이션을 OS 및 프로그램 언어에 상관없이 상호 운영이 가능 하도록 해주는 표준화된 소프트웨어 기술로서 거래 업체간의 필요한 서비스를 발견, 제공하여 다양한 비즈니스를 가능케 해준다.
(자료:정보통신정책 제 14권 15호-웹 서비스의 현황 및 비즈니스 모델의 변화, 정부연)

➢ 사내 또는 외부에서 인터넷을 통해 제공되는 컴포넌트(Components)들을 기계-대-기계 (Machine-to-Machine)간에 자동으로 연결시켜 가상의 어플리케이션을 구축할 수 있도록 하는 것으로 이상적으로는 전세계 컴퓨터를 하나도 통합하기 위한 움직임이라고 할 수 있다.
(자료:ie매거진의 ie포럼-소프트웨어 개발 및 배포방식의 변화, 웹 서비스, 박범대)

Web Service의 정의

발표기관	정의
W3C	하나의 URI(Uniform Resources Identifier)에 의해 정의된 소프트웨어 어플리케이션으로, XML 구조에 의해 어플리케이션 인터페이스 및 결합의 정의, 서술, 발견이 가능해지고, 인터넷기반의 프로토콜을 경유한 XML 기반의 메시지를 사용하는 다른 소프트웨어 어플리케이션과 직접적인 상호운영이 가능하도록 지원해 주는 것
IDC	웹 서비스 아키텍처는 인터넷 프로토콜(IP), SOAP, WSDL 등을 포함한 공개적인 접속 표준과 독자적으로 기술된 구성요소에 의해서 다이나믹한 구성요소 간의 연결성 및 상호운영성을 가능하게 하는 표준화된 접근 방식
Barland	웹 서비스는 인터넷 상에서 표준화된 기술을 사용하여 운영되고 발견될 수 있는 비즈니스 프로세스
Gartner	웹 서비스는 e-비즈니스 관련 표준을 기반으로 하고 있으며, 인터넷을 통해 제공되는 비즈니스 로직을 갖는 소프트웨어 컴포넌트
Aankee Group	1)분산된 어플리케이션을 구축하기 위한 개발 툴 2)전체 어플리케이션 또는 어플리케이션 조립을 위한 구성요소를 전달하는 소프트웨어로서의 ASP모델

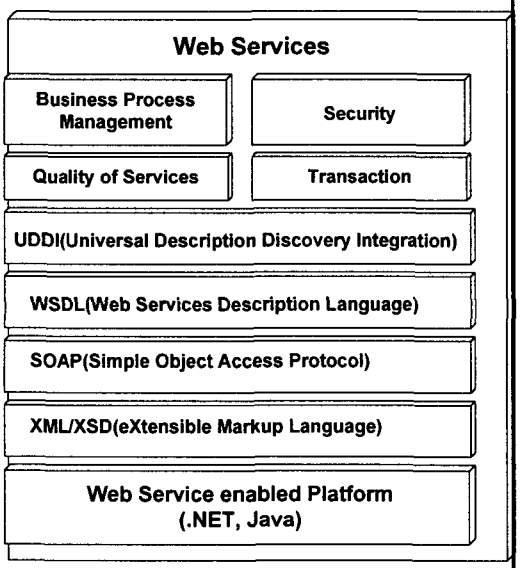
Web Service 기본구조

XML
- 데이터를 표현하기 위한 표준방식

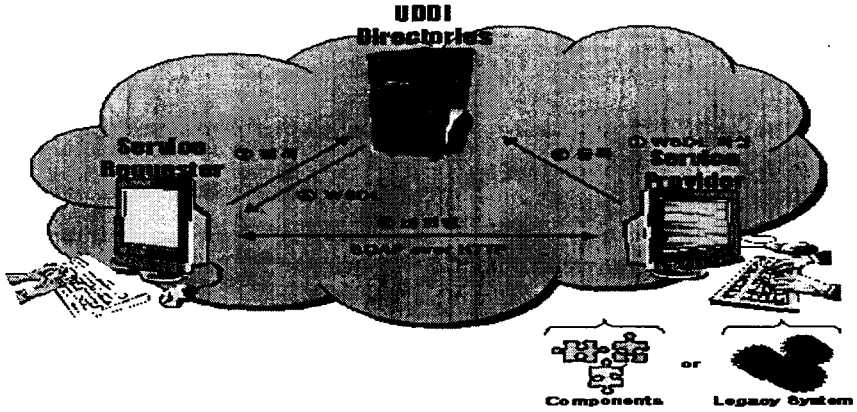
SOAP
- 웹 서비스에서 사용되는 보편적이며 확장성이 있는 메시지 포맷

WSDL
- 웹 서비스를 표현하고 기술하는 언어

UDDI
- 웹 서비스를 제공하는 Provider를 검색 가능하게 하는 기능



Web Service Scenario



① Service Provider : 비즈니스 정보 및 서비스 정보를 WSDL 파일에 작성

③ UDDI 레지스트리에 등록, 비즈니스 개체나 서비스는 고유 ID를 부여 받음

③ Service Requester는 UDDI에서 분류 및 식별체계 조건 등으로 검색

④ 해당 서비스를 이용하기 위한 WSDL 파일을 받음

⑤ 이 정보를 이용해 서비스 제공자와의 연결을 통해 서비스를 제공 받는데 실행 결과값만을 XML 문서 형태로 받게 된다.

SOAP Sample

```
POST /BookStore HTTP/1.1
Host: www.amazon.com
Content-Length: 162
Content-Type: text/xml; charset="utf-8"
SOAPAction:
   "http://www.amazon.com/BookService#OrderBook"

<Envelope>
    <Body>
        <OrderBook>
            <title>B2B Electronic
              Commerce</title>
            <author>HDKIM</author>
        </OrderBook>
    </Body>
</Envelope>
```

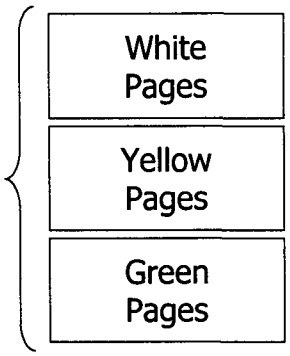
HTTP Request

SOAP HTTP Binding

SOAP Payload

Registry Data

- Businesses register public information about themselves



- Standards bodies, Programmers, Businesses register information about their Service Types

White / Yellow / Green Pages

- White Page Information
  - Business Name
  - Text Description
    - list of multi-language text strings
  - Contact Info
    - names, phone numbers, fax numbers, web sites...
  - Known Identifiers
    - list of identifiers that a business may be known by - DUNS and others
- Yellow Page Information
  - 3 standard taxonomies in V1
    - Industry: NAICS (Industry codes - US Govt.)
    - Product/Services: UN/SPSC (ECMA)
    - Location: Geographical taxonomy
  - Implemented as name-value pairs to allow any valid taxonomy identifier to be attached to the business white page
- Green Page Information
  - Technical information on how to "do e-commerce" with businesses
  - Nested model
    - Business processes
    - Service descriptions
    - Binding information
  - Programming/platform/implementation agnostic
  - Services can also be categorized

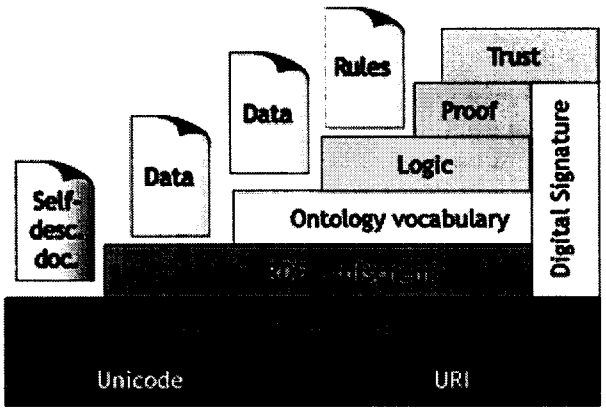
[II] Semantic Web and Ontology

1. Semantic Web
2. Ontology Languages
   - 2.1 RDF and RDFS
   - 2.2 DAML+OIL
   - 2.3 OIL
3. Sources for Ontology
   - 3.1 UNSPSC
   - 3.2 OpenCyc
   - 3.3 MINDSWAP

1. Semantic Web

- Definition (http://www.w3c.org/2001/sw)
  - The Semantic Web is the representation of data on the World Wide Web. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming.
- "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." -- Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001
- The Semantic Web is a vision: the idea of having data on the web defined and linked in a way, that it can be used by machines - not just for display purposes, but for using it in various applications. (http://www.semanticweb.org/)
- The Semantic Web is based on languages that make more of the semantic content of the page available in machine-readable formats for agent-based computing
  - A "semantic" language that ties the information on a page to machine readable semantics (ontology)

Semantic Web Architecture



Designed by Tim Beners-Lee

2. Ontology Languages

- Ontology
  - A formal explicit specification of a shared conceptualization [Gruber]
- Resource Description Framework (RDF) [RDF]
  - Represent metadata with resource, property, statement triple
- RDF Schema (RDFS) [RDFS]
  - Describe RDF vocabularies
- DARPA Agent Markup Language (DAML) [DAML]
  - More expressive schema language
- Ontology Inference Layer (OIL) [Dieter]
  - European lead initiative for frame based representation
- DAML+OIL [Debora]
  - Integration of DAML and OIL, Latest version of DAML
- Web Ontology Language (OWL) [OWL]
  - W3C standard ontology language

2.1 RDF and RDFS

- RDF (Resource Description Framework) [RDF]
  - Directed labeled graphs representation system
  - Suitable for describing any Web resource
    - Use metadata to describe the data contained on the Web
  - Goal
    - Add formal semantics to the Web
  - RDF description model elements
    - Resources : All things being described by RDF expressions
    - Properties : A specific aspect, characteristics, attribute, or relation used to describe a resource
    - Statements
      - A specific resource together with a named property plus the value of that property for that resource

- RDFS (RDF Schema) [RDFS]
  - Describes one set of properties and classes for describing RDF schemas
  - A set of ontological modeling primitives on top of RDF
  - Provide mechanisms for declaring properties, relationships between these properties and other resources

2.2 DAML+OIL

- **DAML[DAML]**
  - DARPA Agent Markup Language – U.S. government sponsored research looking at creating tools to facilitate the semantic web
  - Motivation
    - Agent-based computing can potentially help us recognize complex patterns in widely distributed, heterogeneous, uncertain information environment
    - Enable agents to understand and process concepts on a Web page
    - Develop a language aimed at representing semantic relations in machine-readable ways
  - Description
    - Tie a page's information to machine-readable semantics
    - Build upon RDF and RDFS
    - Create greater support for enumerations and extend the concept of classes to include greater cardinality

2.3 OIL

- Ontology Inference Layer – European lead initiative [Dieter]
- Designed to represent machine-accessible semantics of information on the Web
- Features
  - Frame-based languages
    - Provide modeling with classes (frames) and properties (slots)
  - Description logics
    - Enables reasoning with concept descriptions and the automatic derivation of classification taxonomies
  - Web standards
    - Has a well defined syntax in XML based on a document type definition (DTD) and an XML schema definition
    - OIL is defined as an extension of the RDF and its schema definition language RDFS
- Applications
  - Search engines
    - Search the semantic concepts underlying the information in Web pages rather than searching for matching keywords
  - E-commerce
    - Comparison shopping using machine-processable catalogs
  - Knowledge management
    - Provide means to structure and access the knowledge

Language Feature Comparison

(http://www.daml.org/language/features.thml)

	XML DTD	XML Schema	DAML+ OIL	RDF(S)	OWL
Bounded lists			o	o	o
Cardinality constraints	o	o	o		o
Class expressions			o		o
Data types		o	o	?	o
Defined classes			o		o
Enumerations	o	o	o		o
Equivalence			o		o
Extensibility			o	o	o
Formal semantics			o	o	o
Inheritance			o	o	o
Inference			o		o
Local restrictions			o		o
Qualified constraints			o		
Reification			o	o	o

3. Sources for Ontology

- DAML Ontology Library (http://www.daml.org/ontologies)
  - Provide summaries and query engine for 208 ontologies
  - Provide statistics for number of classes, properties, and instances of each ontology
    - Total classes: 56326
    - Total properties: 8602
    - Total instances: 64822
- Universal Standard Products and Services Classification Code (UNSPSC)
  - http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml
  - 9795 classes are available
- OPENCYC
  - http://opencyc.sourceforge.net/daml/cyc.daml
  - 1740 classes, 859 properties, and 2572 instances are available
- Maryland Information and Network Dynamics Lab Semantic Web Agents Project (MINDSWAP)
  - http://www.mindswap.org/2003/CancerOntology/nciOncology.owl
  - 25,762 classes, 96 properties, and 25851 instances are available

3.1 UNSPSC

- Universal Standard Products and Services Classification Code
  - Result of a merger of the United Nations' Common Coding System (UNCCS) and Dun & Bradstreet's Standard Product and Services Codes (SPSC).
- Features
  - First coding system to classify both products and services for use throughout the global marketplace
  - Enables users to consistently classify the products and services they buy and sell
  - UNSPSC is considered an open standard
  - Encouraged to apply the codes in the business systems
- Ontology source
  - http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml
- Statistics
  - 9795 classes are available

3.2 OpenCyc

- Open source version of the Cyc technology
  - Cyc: general knowledge base and commonsense reasoning engine
- Support
  - speech understanding
  - database integration
  - rapid development of an ontology in a vertical area
  - email prioritizing, routing, summarization, and annotating
- Release 1.0 of OpenCyc
  - 6,000 concepts: an upper ontology for all of human consensus reality.
  - 60,000 assertions about the 6,000 concepts, interrelating them, constraining them, in effect (partially) defining them.
  - Cyc Inference Engine, Cyc Knowledge Base Browser.
  - A specification of CycL, the language in which Cyc is written
  - A specification of the Cyc API
- Ontology source
  - http://opencyc.sourceforge.net/daml/cyc.daml
  - 1740 classes, 859 properties, and 2572 instances are available

3.3 MINDSWAP

- Maryland Information and Network Dynamics Lab Semantic Web Agents Project
  - Working with Semantic Web technology inside the MIND LAB University of Maryland Institute for Advanced Computer Studies
- The National Cancer Institute (NCI) Cancer Ontology
  - Public domain description logic-based terminology
  - Produced by the National Cancer Institute
  - Intended to facilitate translational research and to support the bioinformatics infrastructure of the Institute
  - Contents
    - Broad clinical vocabularies
    - Topics : diseases, drugs, chemicals, diagnoses, genes, treatments, anatomy, organisms, and proteins.
- Statistics
  - 500,000 triples.
  - 25,762 classes, 96 properties, and 25851 instances are available
- Ontology source
  - http://www.mindswap.org/2003/CancerOntology/nciOncology.owl
  - 46MB

References

- Semantic Web, http://www.w3.org/2001/sw/
- RDF, Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://www.w3.org/TR/REC-rdf-syntax/.
- RDFS, RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft 23 January 2003, http://www.w3.org/TR/rdf-schema/.
- DAML, DAML Language (DAML+OIL), http://www.daml.org/language/
- OWL, OWL Web Ontology Language Overview, http://www.w3.org/TR/owl-features/
- DAML Ontology Library, http://www.daml.org/ontologies/
- Dieter Fensel, Frank van Harmelen, Ian Horrocks, Debora L. McGuinness, Peter F. Patel-Schneider, "OIL: An Ontology Infra-structure for the Semantic Web", IEEE Intelligent Systems, vol.16, no.2, March/April, 2001, pp.38-45.
- Debora L. McGuinness et al., "DAML-ONT: An Ontology Language for the Semantic Web", http://www.daml.org/2000/10/daml-ont.html.
- Debora L. McGuinness et al., "DAML+OIL: An Ontology Language for the Semantic Web", IEEE Intelligent Systems, vol.17, no.5, September/October, 2002, pp.72-80.
- UNSPSC, http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml
- OpenCyc, http://opencyc.sourceforge.net/daml/cyc.daml
- MINDSWAP, http://www.mindswap.org/2003/CancerOntology/nciOncology.owl

[III] Rule Markup Languages

- **Introduction**
- **Literature Review**
  - **RuleML**
  - **AIML**
  - **CBML**
  - **RFML**
  - **AORML**
  - **BRML**
  - **Mapping XML to Java**
- **Possibility of Study and Further Study**
- **References**

RuleML 1/2

- http://www.dfki.uni-kl.de/ruleml/
- Open network of individuals and groups from both industry and academia
- Goal
  - Provide a basis for an integrated rule-markup approach
    - Will be beneficial to all involved and to the rule community at large
  - Support collaboration among participants
    - Establish translations between existing tag sets
    - Converge on a shared rule-markup vocabulary
  - RuleML kernel language can serve as a specification for immediate rule interchange

RuleML 2/2

- Application domains in which rules are being used for interconnected purposes
  - Engineering : Diagnosis rules
  - Commerce : Business rules
  - Law : Legal reasoning
  - Internet : Access authentication
- Scope
  - In a combination of natural language rules and rules of some formal notation
  - Working towards an XML-based markup language that permits Web-based rule storage, interchange, retrieval, firing application
- Participants

RuleML Example

```
<?xml version="1.0"
   standalone="no"?>
<!DOCTYPE rulebase
   SYSTEM "http://www.dfki.de/
   ruleml/dtd/ruleml-datalog-
   standalone.dtd">

<rulebase>

<!-- This example rulebase
   contains four rules. The third
   and
fourth rules are actually facts.

In English:
The first rule says that a person
   owns an object if that person
   buys
the object from a merchant and
   the person keeps the object. --
   >

<if>
 <atom>
  <rel>own</rel>
  <var>person</var>
  <var>object</var>
 </atom>
 <!-- explicit 'and' -->
 <and>
  <atom>
   <rel>buy</rel>
   <var>person</var>
   <var>merchant</var>
   <var>object</var>
  </atom>
  <atom>
   <rel>keep</rel>
   <var>person</var>
   <var>object</var>
  </atom>
 </and>
</if>

</rulebase>
```


CBML

- Distributed CBR using XML
  - By Conor Hayes and P.Cunningham(Trinity College, Dublin)
  - Motive
    - Client-server interaction is long lived in Web-based CBR
    - Most Web-based CBR is thin client applications
    - Response time may be poor
  - Objective
    - Perform some of the case-base processing on the client side
    - Improve overall response times
  - CBML (Case Based Markup Language)
    - Use CBML for exchange of case between client and server
    - Has advantages of interoperability and ease of reuse

RFML

- Relfun (Relational-Functional language)
  - By Harold Boley(Deutsches Froschungszentrum fuer Kuenstliche Inteligenz GmbH)
  - Logic-programming language with call-by-value (eager) expressions
  - Used for definitions of relations and functions
    - Much of the knowledge on the Web constitutes definitions of relations and functions
  - Relfun has two syntax : Prolog-like, Lisp-like
- RFML (Relational-Functional Markup language)
  - XML version of Relfun
  - Converted from Relfun
    - Prolog-like syntax is the source for conversion and Lisp-like syntax is shown as the intermediary form

Examples of RFML

```
Likes(john, mary)        //Prolog-like syntax

(hn (likes john mary))   //Lisp-like syntax

<hn>                     //RFML
<pattop>
<con> likes </con>
<con> john </con>
<con> mary </con>
</pattop>
</hn>


likes(john,X) :- female(X),
    likes(X,wine).
    //Prolog-like syntax

(hn (likes john _x) (female _x)
    (likes _x wine))
    //Lisp-like syntax
```

```
// RFML
<hn>
<pattop>
<con> likes </con>
<con> john </con>
<var> x </var>
</pattop>
<callop>
<con> female </con>
<var> x </var>
</callop>
<callop>
<con> likes </con>
<var> x </var>
<con> wine </con>
</callop>
</hn>
```

CommonRules and BRML

- **CommonRules**
  - By IBM Research's Business Rules for E-Commerce
  - Web communication of executable business rules
    - between enterprises using heterogeneous rule systems
    - Communicate detailed conditions and policies expressed in rules between Buyers, sellers and suppliers
  - Specification of executable business rules
    - Help non-programmer business domain experts
  - BRML
    - Common "interlingua" rule representation for exchange of rules between heterogeneous rule systems
    - XML rule interchange format for rules
  - Java library

BRML Example

Let C_1 be a simple example CLP ruleset that contains single ruls
giveDiscount(percet5, ?Cust) <-shopper(?Cust) and loyalCustomer(?Cust),

→ This rule says to give 5% discount to loyal customers.

```
<?xml version="1.0"?>
<!DOCTYPE brml system "brml.dtd">
<clp>
   <erule rulelabel="emptyLabel">
     <head>
       <cliteral predicate="giveDiscount">
        predicate="loyalCustomer">
          <function name="percent5"/>
         <variable name="?Cust"/>
       </cliteral>
     </head>
     <body>
      <and>
      <fcliteral predicate="shopper">
        <variable=name="?Cust"/>
      </fcliteral>
      <fcliteral
        <variable=name="?Cust"/>
      </and>
     </body>
   </erule>
</clp>
```

Mapping XML to Java

- **Mapping XML to Java**
  - **By Robert Hustead (Javaworld)**
  - **Challenges of using XML**
    - **Generating XML is straightforward, but the inverse is not**
    - **Heavy memory requirements and slow speeds can be a problem**
  - **SAX, DOM APIs**
    - **Popular parsing standards**
- **ART*Enterprise**
  - By Asid Tabet, etc. (MindBox)
  - Integrated knowledge-based application development environment from MindBox Inc.
    - Supports rule-based, case-based, object-oriented and procedural representation and reasoning of domain knowledge
  - Use XML as input and output of rule-based application
  - **XML → Java objects (preprocess and validate)→ A*E objects**

References

- RuleML, http://www.dfki.uni-kl.de/ruleml/
- AIML, http://www.alicebot.org/
- CBML, http://boffin.ucd.ie/Ulysses/CBR/CBR_CBML.htm
- AORML, http://tmitwww.tm.tue.nl/staff/gwagner/AORML
- BRML, http://www.research.ibm.com/rules/
- Mapping XML to Java, http://www.javaworld.com/javaworld/jw-08-2000/jw-0804-sax_p.html
- ART*Enterprise, http://people.ne.mediaone.net/stabet/urml-pricai2000.html

Display for Human's Comprehension and Representation for Agents

Research budgets is organized within limits of the contract deposit and distributed to each accounts. The use of research budget is as follows:

If the budgetary source is the type-P research fund, the spendable items are student salary and expenses for data collection.

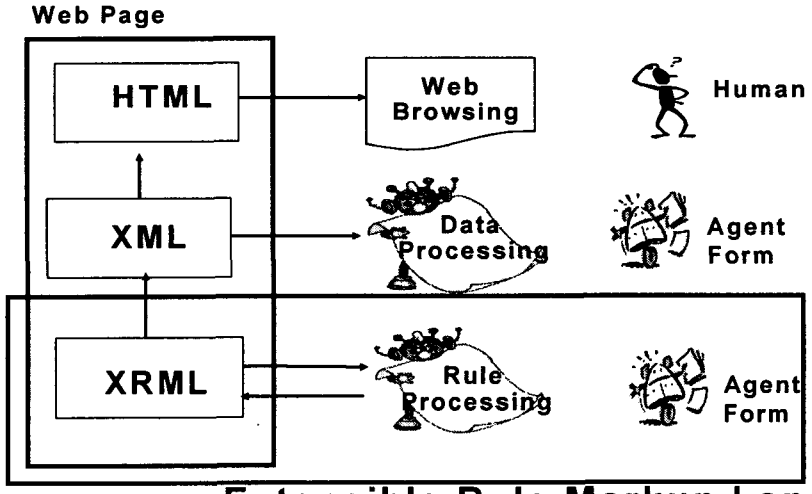
The use of research budget could be changed by the decision of board of directors. Only a person who possess the right of modification can change the regulation and the changed regulation is enforced after the president's approval.

Rule Group: Budgetary Type Constraints

If the residual of budgetary source less than purchase amount, then purchase is halted

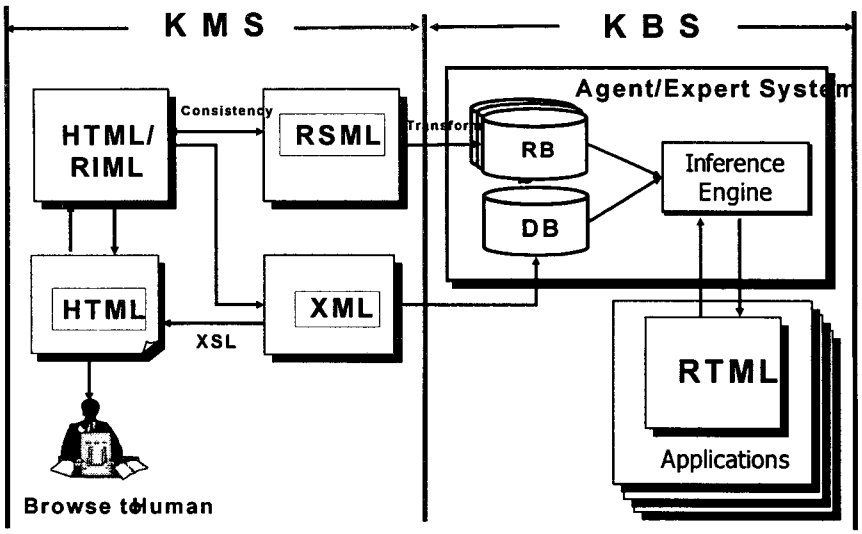
Rule Title: Budgetary Appropriateness
IF [budgetary-source IS type-P research fund]
AND [[item IS student's-salary
OR [item IS data-collection-expenses]]
THEN expenditure IS permitted

[IV] Topology of XRML



Extensible Rule Markup Language

Architecture of XRML with Components



eXensible Rule Markup Language

- Rule Identification Markup Language
  - Identification of relevant rules in paragraphs
  - Representation of existence of rules and its variables/values and other rule structures
- Rule Structure Markup Language
  - Representation of rules in Markup Syntax
  - Rule extraction and keep consistency between the two
- Rule Triggering Markup Language
  - Trigger an inference engine against the rule base

Advantages of XRML

- ❖ Aid Human's Knowledge Processing:
  - Conversion of XRML (including Rule Identification Language) to XML/HTML
- ❖ Aid Agent's Knowledge Processing against Web page
  - Triggering rule-based inferences in the agents, possibly from the Workflow Management System
- ❖ Maintain Consistency between Rule Base and Web page
  - Aid the extraction(semi-automatically) of rules from XML maintaining consistency between them
- ❖ Integration of KBS and KMS

HTML Documents and Canonical Rule - Assumption: a complete rule from a page

<HTML>

A research account can be spent only within the limit of the contract budget, according to the following restrictions.

<h2>**Restriction of Type-P Research Fund Expenditure**</h2>

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.

</HTML>

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

Illustrative HTML/RIML

Browsed Display

A research account can be spent only within the limit of the contract budget, according to the restrictions. The use of research budget is as follows:

Restriction of Type-P Research Fund Expenditure

If the budgetary source is the type-P research fund, the spendable items are student salary and expenses for data collection.

HTML/RIML

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

<RIML Version = "0.5">
<Rule>
<URL>http://xrml.kaist.ac.kr/XRML/RSML/sp.rsml</URL>
<h2><RuleTitle> Restriction of Type-P Research Fund
Expenditure</RuleTitle></h2>

<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>
```


Rule Structure Markup Language

Rule

Rule Group: Budgetary Type Constraints

Rule Title: Budgetary Appropriateness
IF *(budgetary-source IS type-P research fund)*
AND *((item IS student's-salary*
OR *(item IS data-collection-expenses))*
THEN *expenditure IS permitted*

RSML

```
<RSML Version="0.5">
<Rule>
<RuleTitle> Restriction of Type-P Research Fund Expenditure </RuleTitle>
<URL>http://xrml.kaist.ac.kr/XRML/RIML/sp.riml</URL>
<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>
```

Association of RIML with RSML

RIML

```
<HTML>
<p>A research account can be spent only within the limit of the contract budget, according to
   the restrictions.</p>
<RIML Version = "0.5">
<Rule>
<URL>http://xrml.kaist.ac.kr/XRML/RSML/sp.rsml</URL>
<h2><RuleTitle> Restriction of Type-P Research Fund Expenditure</RuleTitle></h2>

<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>
```

RSML

```
<RSML Version="0.5">
<Rule>
<RuleTitle> Restriction of Type-P Research Fund Expenditure </RuleTitle>
<URL>http://xrml.kaist.ac.kr/XRML/RIML/sp.riml</URL>
<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>
```


Generation of RSML rule

❖ Mapping of the keywords and corresponding slots of variables and values

<RIML Version="0.5" >	↔	<RSML Version="0.5">
<RuleGroup>	↔	<RuleGroup>
<RuleGroupTitle>text</RuleGroupTitle>	↔	<RuleGroup Title>text</RuleGroupTitle>
<Rule>	↔	<Rule>
<URL>URL of RSML file</URL>	↔	<URL> URL of RIML file </URL>
<RuleTitle>title of rule</RuleTitle>	↔	<RuleTitle> title of rule </RuleTitle>
<variable1>data of vr1</variable1> text<value1>data of vl1</value1>text <variable2>data of vr2</variable2> text <valuei2>data of vr2</value2> <valuei2>data of vr2</value2>	↔	<IF> <data of vr1>data of value1</data of vr1> </IF> <THEN> <data of vr2>data of value2</data of vr2> </THEN>
</Rule>	↔	</Rule>
</RuleGroup>	↔	</RuleGroup>
</RIML>	↔	</RSML>

Identification of Table

❖ **Define new tags for table**

IF part **THEN part**

variablei1, variablei2 ...	variablet1, variablet2 ...
valuei11, valuei12...	valuet11, valuet12...
valuei21, valuei22...	valuet21, valuet22...

- **Generated rule**

IF variablei1=valuei11
AND
variablei2=valuei2 ...
THEN
variablet1=valuet11

IF variablet2=valuet2
...

- **RIML representation of tables with new tags**

```
<RuleTable>
<RuleTitle>Title of rule</RuleTitle>
  <Tvariable1>
    <TIF1>variablei1</TIF1>
    <TIF2>variablei2</TIF2> ...
    <TTHEN1>variablet1</TTHEN1>
    <TTHEN2>variablet2</TTHEN2>
  ...
  </Tvariable1>
  <Tvalue1>
    <TIF1>valuei11</TIF1>
    <TIF2>valuei12</TIF2> ...
    <TTHEN1>valuet11</TTHEN1>
    <TTHEN2>valuet12</TTHEN2>
  ...
  </Tvalue1>
  ...
</RuleTable>
```


Identification of Operator

❖ Example

HTML: *The package's value is more than $150*

Rule: *(package's value > $150)*

❖ Various representation of "more than" in HTML

- >, greater than, more than

❖ Comparison operators are represented in RIML

Operator in HTML	Operator in RIML
more than, greater than, >	**<operator type="GT">**
less than, <	**<operator type="LT">**
less than or equal to, <=	**<operator type="LE">**
greater than or equal to, >=	**<operator type="GE">**
is not	**<operator type="NOT">**

Identification of Function

- **Function types are basic mathematical operators in RIML:**
  +, -, *, /

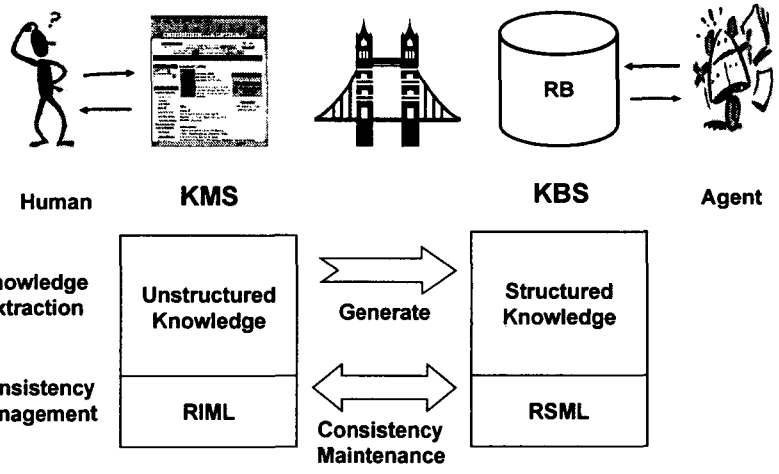
- **Example**
  HTML: Total Delivery Time =
  Available-to-Ship Time + Delivery Method Ship Time

  Rule: (Total_Delivery_Time IS
  (+ Available-to-Ship_Time Delivery_Method_Ship_Time))

- **Represent mathematical operator as function in RIML**

```
<variable1> Total Delivery Time </variable1>
<value1>
    <function type="+">
        <arg1> Available-to-Ship Time </arg1>
        <arg2> Delivery Method Ship Time </arg2>
    </function>
</value1>
```


Knowledge Extraction and Consistency Management with XRML



Balance the Sophistication of RIML

- **Tradeoff between RIML sophistication and RSML automation**
  - **Criteria**
    - **Effort of knowledge edition**
    - **Degree of relationship comprehension**
    - **Degree of automatic generation of RSML**
- **Candidate tags**
  - **<Rule>, <RuleTitle>, <variable>, <value>, <IF>, <THEN>, <AND>, <OR>, <operator>, <function>**
- **Minimum tags for identification**
  - **<Rule>, <RuleTitle>, <variable>, <value>**

Challenges – The Opportunity

- **Challenges**
  - Consistency maintenance of polymorphic knowledge representations
  - Domain specific thesaurus – Ontology
    - Ontology Building with XRML
    - Using Ontology to Identify XRML
  - Multi-URL based rule extraction and inference
  - Integration of rules from RSML with other sources
- To aid the rule and meta-rule extraction
  - **XRML Knowledge Editor**

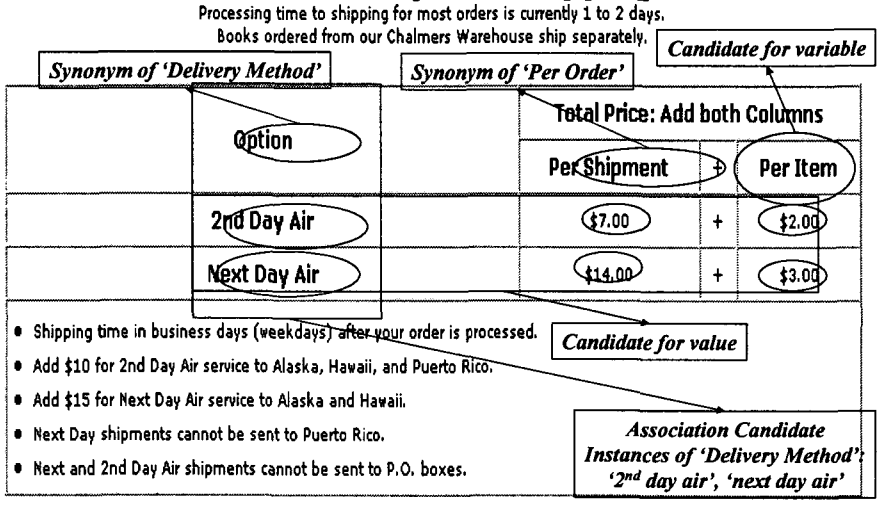
Building RIML with Ontology

- **Advantage of using ontology**
  - **Help knowledge engineer to find variable and value**
  - **Recommend candidates for variable and value → Highlight candidates**
  - **Recommend candidates for association of variable and value → Highlight candidates for association or show list of values for selected variable**
- **Identification of variable with ontology**
  - **Every property can be any type of variable**
    - **Ex) 'per Item', 'per Order', 'trackable', 'ship Time'**
  - **Class can be an OAV type variable**
    - **Ex) 'Delivery Method'**
- **Identification of value with ontology**
  - **Numeric value**
    - **Ex) 1.99, 0.99**
  - **Fact type value**
    - **Instance can be a variable**
    - **Ex) 'Yes', 'No', 'true', 'false'**
  - **OAV type value**
    - **Instance can be a value**
    - **Ex) 'Standard Delivery', '1 to 2 business days'**
    - **Subclass can be a value**
    - **Ex) 'Electronics', 'Computer'**

Association of Variable and Value

- Numeric variable and value
  - Associate identified numeric variable with adjacent identified numeric value
    - The value must be the same type of variable
- Fact type variable and value
  - Associate identified fact type variable with adjacent identified fact type value
- OAV type variable and value
  - Associate identified OAV type variable with adjacent identified OAV type value
    - The value must be the instance of the class which is the domain of identified variable property

Example of Identification Using Ontology



KAIST Jae Kyu Lee - 61 -

Procedure of Building RIML with Ontology

Step	Manual Step by Knowledge Engineer	Automated Step with intelligent XRML Editor
A. Symbol Identification with ontology		1. Recommend candidates for variable and value
	2. Determine variable and value among candidates	3. Assign <variable>, </variable>, <value>, </value>
B. Symbol Identification with Synonym		4. Recommend synonym as candidate for variable and value - Decide variable/value with the type of standard word
	5. Determine synonym, variable and value among candidates	6. Assign <variable>, </variable>, <value>, </value>
C. New Symbol Identification	7. Identify new variable, value, synonym in the HTML	8. Assign <variable>, </variable>, <value>, </value>
	9. Update ontology with new variable, value and synonym	10. Register variable and value to ontology as proper class, property and instance
D. Association with ontology		11. Recommend candidates for association of variable and value
	12. Associate recommended candidates and new pair of variable and value	13. Assign variable ID to variable and value

KAIST

Building RSML Using Ontology

- Specialization of Rule
- Generalization of Rule
- Rule Refinement using Ontology

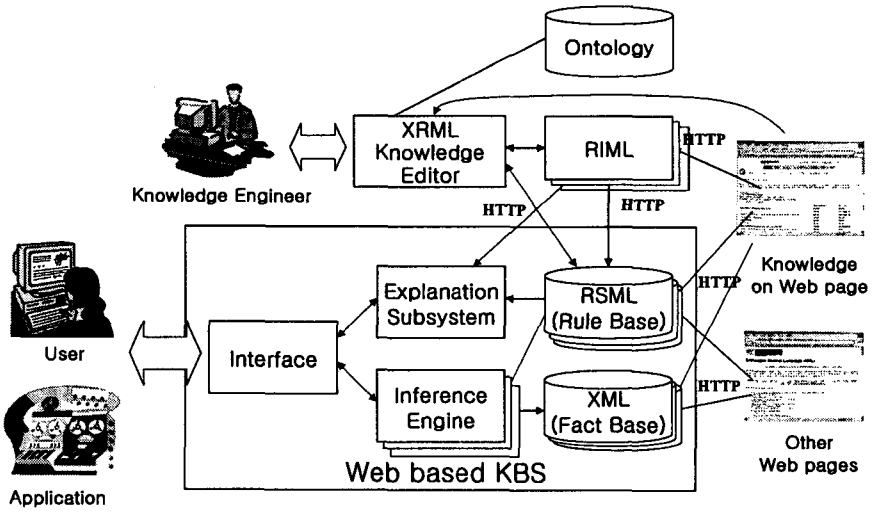
Specialization of Rules

- Concept of specialization
  - If a new rule have condition a1 which is the sub-class of class A that is included in other existing rule as a condition, then the condition part of existing rule must be modified with other sub-classes of the class A.
  - Graph-based editor can automatically generate this modification and recommend it
- Existing rule
  - A → B
- Existing ontology
  - A = (a1, a2)
- New rule
  - a2 → C
- Recommendation for existing rule
  - a1 → B
  - A and NOT a2 → B

Generalization of Rules

- **Concept of Generalization**
  - **If a new rule have condition a2 which is the sub-class of class A and other existing rule have condition a1 which is the sub-class of same class A, then the new rule and the existing rule can be merged into one rule which uses condition A, *super-class* of a1 and a2.**
  - **Graph-based editor can automatically generate and recommend this merge**
- **Existing rule**
  - **a1 → B**
- **Existing ontology**
  - **A = (a1, a2)**
- **New rule**
  - **a2 → B**
- **Merge existing rule and new rule**
  - **A → B**

XRML Knowledge Editor



Rule Processing by Forms with RTML

❖ Automatic consulting by forms

- To perform the automated rule processing, form of workflow or other application can trigger the agents to generate adequate inference results.

❖ Definition of a set of standard statements

- When to trigger the inference
- Which rules to use
- How to use the obtained result

Illustrative Example of RTML

```
<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>
<UseResult>
  <expenditure>permitted</expenditure >
</UseResult>
</RTML>
```

Triggering Condition

Relevant Rule

Returning Result

Design Criteria of XRML (1)

- Expressional Completeness:

  RSML should be completely transformable to a canonical syntax of structured rules

  XRML-XML based KBS

- Relevance Linkability

  Linkages of the relevance between hypertexts with RIML and RSML rules syntax should be completely expressed

- Polymorphous Consistency:

  Consistency should be maintained for knowledge expressed in different types of expressions, such as RSML rules and hypertext with RIML.

  Ontology about variable, values, and synonyms necessary

Design Criteria of XRML (2)

- Applicative Universality:

  The rule expressions in RSML should be able to support multiple applications which embeds RTML within the domain universe

  Scope of target representations

- Knowledge Integrability:

  Structured rules collected from multiple sources Including the rules from RSML should be integrated uniformly.

- Interoperability :

  Rules in RSML should be exchangeable and sharable among multiple commercial solutions

Application Development Procedure Using XRML

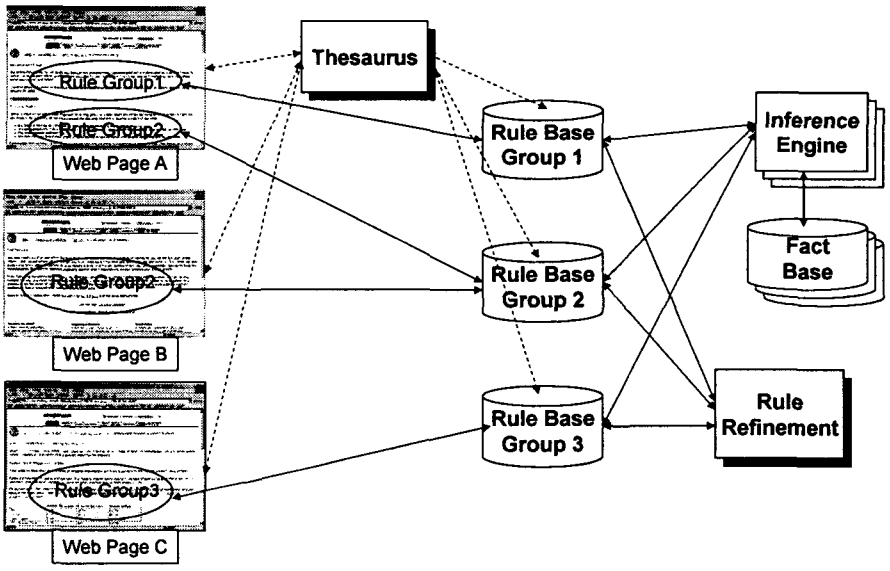
- Define application using knowledge from Web pages
- Design Rule Base in high level
- Associate Rule Groups in the Web Pages with Rule Base Groups

- Rule Identification from HTML using RIML tags: Generate RIML document
- Rule Structuring from RIML document: Generate RSML

- Integrate RSML with Rule Base, and add the External Knowledge Sources
- Maintain Consistency between Web Pages and Rules using the Knowledge Editor

- Deploy the Web based KBS using XRML

Knowledge Extraction and Integration from Multiple URLs



Issues with Multiple URLs

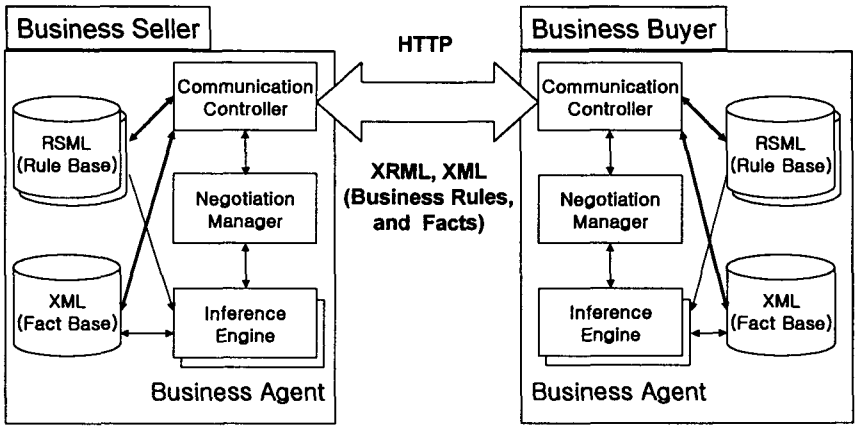
- Association of Rule Group in the Web page with existing Rule Base
- Association of Rule Group in the Web page with new Rule Base
- *How to refine rules from multiple URLs*
- *How to share the symbols of multiple URLs which are associated with the same Rule Base*
  - *Need thesaurus for each Rule Base - Ontology*

Applications of XRML

- Architecture of Web Based KBS Using XRML
  - Configurations
  - Regulations (eGovernment): 약관심사
  - Insurance Underwriting
  - Delivery Cost Comparison Site
- Contact Center
  - Web + Call Center
- Intelligent Workflow
- Multi-Agent based EC for B2B
- Intelligent Knowledge Repository
- Assist Knowledge Collaboration
- XRML at Web Service

Agent based Commerce with XRML

- **Agents exchange not only data or fact, but also business rules represented by XRML through HTTP**
  - **Contract terms and conditions, Bidding rules or Auction rules, Negotiation rules for CPFR**



Concluding Remark

- XRML identifies rules in the Web, and process both for human and agents
- XRML support extracting rules from the Web pages
- XRML can help building ontology
- XRML identification can be assisted by ontology
- XRML supports consistency between multiple representations
- XRML can be a bridge between KBS and KMS
- The notion of XRML can extend its target representation to Cases, Optimization Models, Constraints, Components: X*ML
- There are tons of application potential.

References

- J.K.Lee, M.Sohn,"Extensible Rule Markup Language toward Intelligent Web Platform", Communications of the ACM, Vol 46, pp59-64, 2003
- J.K.Lee, J.Y.Kang, "Rule Acquisition and Mutual Consistency Maintenance with XRML", 2003, Working Paper
- Gabriele Schmidt, Thomas Wetter, "Using Natural Language Sources in Model-Based Knowledge Acquisition", Data & Knowledge Engineering, v.26, no.3, pp 327-356, 1998.
- Dieter Fensel, Karsten Poeck, "A Comparison of Two Approaches to Model-based Knowledge Acquisition", LNCS, no.867, 1994.
- Ajay Hemmnani, Stephane Bressan, "Extracting Information from Semi-structured Web Documents", LNCS 2426, pp. 166-175, 2002.
- Anette Hulth et. al., "Automatic Keyword Extraction Using Domain Knowledge", LNCS 2004, pp. 472-482, 2001.
- Mario Lenz, Andre Hubner, Mirjam Kunze, "Textual CBR", LNCS 1400, pp. 115-138, 1998.
- Linda L. Rodi, Jonathan A. Pierce, Robert E. Dalton, "Graphical Knowledge Acquisition for Fault Diagnosis and Repair", Tool-Based Knowledge Acquisition.
- RDF, Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://www.w3.org/TR/REC-rdf-syntax/.
- RDFS, RDF Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft 23 January 2003, http://www.w3.org/TR/rdf-schema/.

References

- Tin A. Nguyen, Walton A. Perkins, Thomas J. Laffey, Deanne Pecora, "Knowledge Base Verification", AI Magazine, Summer, 1987.
- Motoi Suwa, A. Carlisle Scott, Edward H. Shortliffe, "An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System", AI Magazine, Fall, 1982.
- T.R. Gruber, "A Translation Approach to Portable Ontologies", Knowledge Acquisition, Vol. 5, 1993, pp. 199-220.
- DAML, Debora L. McGuinness et al., "DAML-ONT: An Ontology Language for the Semantic Web", http://www.daml.org/2000/10/daml-ont.html.
- Debora L. McGuinness et al., "DAML+OIL: An Ontology Language for the Semantic Web", IEEE Intelligent Systems, vol.17, no.5, September/October, 2002, pp.72-80.
- Dieter Fensel, Frank van Harmelen, Ian Horrocks, Debora L. McGuinness, Peter F. Patel-Schneider, "OIL: An Ontology Infra-structure for the Semantic Web", IEEE Intelligent Systems, vol.16, no.2, March/April, 2001, pp.38-45.
- John McCarthy, "Circumscription - A Form of Nonmonotonic Reasoning", Artificial Intelligence 13(1,2):27-39, 1980, pp. 171-172.
- Mario Lenz, Andre Hubner, Mirjam Kunze, "Textual CBR", Lecture Notes in Computer Science, issue. 1400, 1998, pp. 115-138.

Special Thanks

❖Special Thanks to:

- 손미애
- 강주영
- 박승룡
- 양성병
- 김우택
- 최규성
- 방홍희
- 윤미중

❖Invitation to XRML Working Group

- **Web Korea Forum**
- **www.webkorea.or.kr**