A New Paradigm of Distributed Problem Solving

Yila Su[1, 2], Chunnian Liu[1], and Limin Liu[2]
1 College of Computer Science and Technology, Beijing University of Technology, China
2 College of Information Engineering, Inner Mongolia University of Technology, China

Summary
Since the World Wide Web is enlarging its scale, users cannot find and utilize information easily. Hence problem solving systems in the Web environment are required. The core of such systems is the Problem Solver Markup Language (PSML) and PSML-based distributed Web inference engines. In this paper, we demonstrate a possible implementation of certain distributed reasoning capablities as required in the future PSML. In particular, our proposed implementation, called β-PSML, is based on the combination of OWL (Web Ontology Language) with Horn clauses. From the viewpoint of expressive power, the proposed β-PSML can represent multi-argument relation that is an extension of the OWL capability, and models domains with a rich hierarchical structure for Horn clauses. Furthermore, we discuss how to extend the β-PSML for solving problems in a large-scale distributed Web environment.
Key words:
Problem Solver Markup Language, Web Intelligence, Semantic Web

Introduction

One of the most significant applications on Web Intelligence (WI) technologies today is enterprise portals operating with state-of-the-art markup languages to search, retrieve and repackage knowledge and data. The enterprise portals are being developed into an even more powerful center based on component-based applications called Web Services [1, 20, 23, 24, 26].

WI researchers must study both portal-centralized and distributed information structures. Information on the Web can be either globally distributed throughout the Web within multi-layer over the infrastructure of Web protocols, or located locally, portal-centralized Web services (i.e., the intelligent service provider) that is integrated to its own cluster of specialized intelligent applications. However, each approach has a serious flaw. As pointed out by Alesso and Smith [1], the intelligent portal approach limits uniformity and access, while the global Semantic Web approach faces combinatory complexity limitations.

Previously, a way to address the above issues by developing and using a Web-enabled problem-solving system for portal-centralized, adaptable Web services has been proposed [18, 22, 26]. The core of such a system is the Problem Solver Markup Language (PSML) and PSML-based distributed Web inference engines, in which among others the following support functions should be provided since this is a must for developing intelligent portals based on WI technologies:

1. Providing the expressive power and functional support for complex adaptive, distributed problem solving;
2. Performing automatic reasoning on the Web by incorporating globally distributed contents and meta-knowledge automatically collected and transformed from the Semantic Web and social networks with locally operational knowledge-data bases;
3. Representing and organizing multiple, huge knowledge-data sources for distributed Web inference and reasoning.
4. Combining multiple reasoning methods in the method efficiently and effectively;
5. Modeling user behavior and representing/managing it as a personalized model dynamically.

A possible way as an immediate step to implement certain distributed reasoning capabilities of the future PSML is to make use of an existing logic language coupled with agent technologies. In this paper, we demonstrate one possible implementation of such capablities. In particular, our proposed implementation, called β-PSML, is based on the combination of OWL (Web Ontology Language) with Horn clauses. We consider two logic languages: the OWL (Web Ontology Language) and Prolog composed of Horn clauses. OWL is a new recommendation of the World Wide Web Consortium (W3C) for the Semantic Web [7]. Based on the description logic (DL), OWL has been designed especially to model domains with a rich hierarchical structure, but its number of arguments is limited to only one or two. On the other hand, Horn clauses can represent multi-argument relation.

This paper provides details on the β-PSML that integrates OWL with Horn clauses, and thus represents multi-argument relation – which cannot be represented by OWL. Another feature of the β-PSML is that it is expressive enough to model domains with a rich hierarchical structure that Horn clauses are not able to do. The current version of β-PSML presented in this paper is

Manuscript received June 5, 2006.
Manuscript revised June 25, 2006.

an extension of the OWL capability for complex adaptive, distributed problem solving, and can be easily used for automatic reasoning on the Web by incorporating global information sources from the Semantic Web and social networks with locally operational knowledge-data bases in an enterprise portal together for decision-making and e-business intelligence.

The remaining of this paper is as follows. Section 2 details how to construct the β-PSML. Section 3 presents the reasoning mechanism in β-PSML. Section 4 discusses how to extend the β-PSML for solving problems in a large-scale distributed Web environment. Section 5 surveys related work. Finally, Section 6 gives concluding remarks.

2. β-PSML

PSML is of great significance in the research of WI [18, 22]. In what follows, we demonstrate an implementation of some distributed reasoning capabilities, as required in the future PSML, that combines OWL and Horn clauses. The β-PSML has strict syntax and semantics as well as strong expressive power. In addition, the two corresponding inference engines can cooperate to solve some problems that cannot be solved by each of the single inference engine separately.

2.1 Constructing β-PSML

Horn clauses can represent multi-argument relation, and based on DL, OWL has been designed especially to model domains with a rich hierarchical structure. However, Horn clauses and DL are not reducible to each other [11]. On the one hand, Horn clauses lack the possibility to express existential quantification. For example, they do not allow to state that every person has a father, which can be easily done in DL:

$Person \subseteq \exists mother.T$

Also negation in the head is not allowed in Horn clauses. Hence it is impossible to state that a person is either a male or a female, while it can be easily expressed in DL:

$Person \subseteq Man \cup Woman$

$Man \subseteq \neg Woman$

On the other hand, DL is strict (decidable) subsets of First Order Logic (FOL) [4]. Based on DL, OWL can only represent predicates of one argument and some predicates of two arguments. For example, in DL one cannot state that individuals who work and live at the same location are home workers, while it can be easily stated with a Horn clause:

$live(X, Y), work(X, Z), loc(Y, L), loc(Z, L) \rightarrow HomeWorker(X)$

More generally, DL is unable to express chains of joins across different predicates.

Statements and constructors of OWL and Horn clauses (rules and facts) are basic elements of β-PSML.

The BNF definition of the rules and facts in Horn clauses is as follows:

```
<clause> ::= <head> | <body>→ <head>
<head> ::= <atom>
<atom> ::= <predicate> | <predicate> (<term list>)
<body> ::= <atom list>
<atom list>::= <atom> | <atom list>, <atom>
<term list> ::= <term> | <term list> , <term>
<term> ::= <constant> | <variable>
```

Because β-PSML is constructed on the XML and OWL, we show the XML concrete syntax based on the OWL-XML presentation syntax as follows.

In β-PSML, we use the same namespace names as those in OWL, which are listed in Table 1. We can propose the definition of every element of β-PSML by combining Horn clauses with OWL. The ontology root element of the β-PSML-XML presentation syntax is extended to include clause (Horn clauses) axioms and variable (variable declaration) axioms.

Table 1: Namespace names

Namespace name	Namespace
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
xsd	http://www.w3.org/2001/XMLSchema#
owl	http://www.w3.org/2002/07/owl#
owlx	http://www.w3.org/2003/05/owl-xml

The definition of element **ontology** is shown in Figure 1. The attribute **psmlx: name** refers to a name of this ontology, which is the base URI of this element. Note that element **ontology** is the root element of β-PSML documents in the XML presentation syntax. Thus we only need to add the relevant syntax for variables and clauses. Furthermore, variable axioms are statements about variables, indicating that the given string is used as a variable.

```
<psmlx:Ontology
  psmlx:name = xsd:anyURI
>
 Content: (owlx:VersionInfo | owlx:PriorVersion |
   owlx:BackwardCompatibleWith | owlx:IncompatibleWith |
   owlx:Imports | owlx:Annotation | owlx:Class[axiom] |
   owlx:EnumeratedClass(D,F) | owlx:SubClassOf(D,F) |
   owlx:EquivalentClasses | owlx:DisjointClasses(D,F) |
   owlx:DatatypeProperty | owlx:ObjectProperty | owlx:SubPropertyOf |
   owlx:EquivalentProperties | owlx:Individual[axiom] |
   owlx:SameIndividual | owlx:DifferentIndividuals |
   prolog:clause[axiom] | prolog:variable[axiom])*
```

```
</psmlx:Ontology>
```

Fig. 1 The definition of element **ontology**

The definition of element **prolog: variable** [axiom] is as follows:

```
<prolog:variable>xsd:string<prolog:variable>
```

The parent of the element is psmlx: ontology. A variable axiom simply defines the existence of a variable. This is taken from the Prolog namespace. For example:

```
<prolog:variable>x<prolog:variable>
```

Taken from the Prolog name space, Clause axioms (clause elements) are similar to SubClassOf axioms. Like SubClassOf axioms, clauses may include annotations.

The definition of element **prolog: clause** [axiom] is as follows:

```
<prolog:clause>
  Content: (owlx:Annotation*, head | owlx:Annotation*, body, head)
</prolog:clause>
```

The parent of the element is psmlx: ontology. Note that this element allows one to say that every binding that satisfies the body of the rule must also satisfy the head of the clause.Both body and head are lists of atoms and are read as the conjunction of the component atoms.

The definition of element **prolog: head** is as follows:

```
<prolog:head>
  Content: (psmls:atom)
</prolog:head>
```

The parent of the element is prolog: clause.

The definition of element **prolog: body** is as follows:

```
<prolog:body>
  Content: (psmls:atom*)
</prolog:body>
```

The parent of the element is prolog: clause.

Atoms can be formed from unary predicates (classes), binary predicates (properties), equalities or inequalities used in OWL or atoms in Prolog. In addition, we can use **predicate** to denote predicates of three or more arguments in β-PSML. Unless contradiction, other definitions of elements in β-PSML are the same as those in SWRL [12].

Translation from the XML Concrete Syntax to RDF Concrete Syntax could be easily accomplished by extending the XSLT transformation for the OWL-XML presentation syntax [6].

According to the construction of β-PSML, Horn clauses and OWL can be translated to the β-PSML easily. Thus, besides Horn clauses, the β-PSML can express ontologies and the knowledge with a rich hierarchical structure represented by OWL on the Semantic Web.

2.2 An Illustrative Example

We will use an example to illustrate how to express knowledge with β-PSML. Consider the following rule in a knowledge base:

made-by(x,y), no-fellow-company(y) → price(x, usa, high)

We can see that the clause cannot be represented with OWL since there are three arguments in a predicate of the clause. However, this rule can be represented by β-PSML as shown in Figure 2. Furthermore, the β-PSML is expressive enough to model domains with a rich hierarchical structure that Horn clauses are not able to do.

```
<prolog:clause>
  <prolog:body>
    <psmlx:individualPropertyAtom psmlx:property="made-by">
      <prolog:variable>x</prolog:variable>
      <prolog:variable>y</prolog:variable>
    </psmlx:individualPropertyAtom>
    <psmlx:classAtom>
      <owlx:Class owlx:name="no-fellow-company"/>
      <prolog:variable>y</prolog:variable>
    </psmlx:classAtom>
  </prolog:body>
  <prolog:head>
    <psmlx:predicatePropertyAtom psmlx:property="price">
        <prolog:variable>x</prolog:variable>
        <owlx:Individual owlx:name="usa"/>
        <owlx:Individual owlx:name="high"/>
    </psmlx:predicatePropertyAtom>
  </prolog:head>
</prolog:clause>
```

Fig. 2 An Illustrative Example of β-PSML

3. Reasoning of β-PSML

The reasoning process of β-PSML, as shown in Figure 3, can be divided into two parts: the reasoning of Horn clauses and the reasoning of DL. Although some queries may need to be completed by combining DL with Horn clauses, we just use one part of the reasoning of β-PSML if the queries on the Web can be completed by one of the two parts.

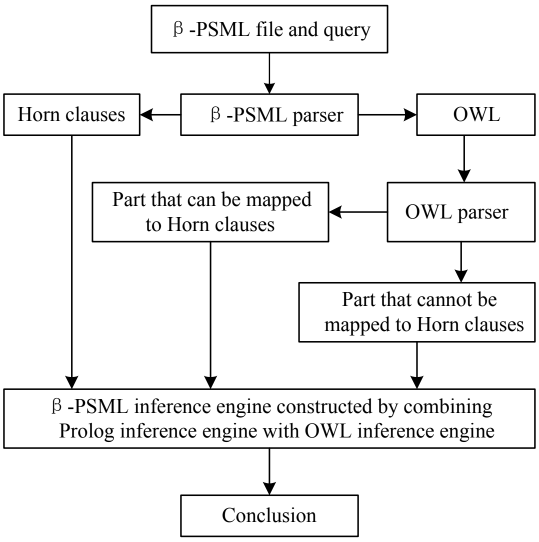

Fig. 3 The reasoning process of β-PSML

From Figure 3, we also can see that some part of OWL can be mapped into Horn clauses [11]. Thus, Horn clauses can be obtained from either the PSML program directly or from the OWL. Furthermore, the β-PSML includes all statements and constructors of OWL, and can express knowledge that cannot be expressed by Horn clauses.

The reasoning process of β-PSML by combining DL with Horn clauses can be divided into the following steps:

Step 1. Separate the program into Horn clauses and DL.

Step 2. Get new facts by using the reasoning algorithm of Horn clauses or DL.

Step 3. If no new facts, the algorithm failures and ends.

Step 4. If one of the new facts is the answer, then the system gives the answer and the algorithm ends.

Step 5. If the facts are got from Horn clauses last time, then using the reasoning algorithm of DL with the facts to get new facts. Go back to step 3.

Step 6. If the facts are got from DL last time, then using the reasoning algorithm of Horn clauses with the facts to get new facts. Go back to step 3.

As discussed above, the reasoning of β-PSML combines Horn clauses with DL loosely. Only facts are exchanged between the two parts. In the future work, we will consider combining them more closely and making reasoning of the β-PSML more efficiently and effectively.

We have constructed an initial prototype of β-PSML by combining BPU-Prolog [17] and OWLAPI [3]. This prototype can answer not only queries that can be completed by Horn clauses or DL, but also queries that cannot be completed only by Horn clauses or DL.

We illustrate the reasoning process of β-PSML with the following example. Given the following knowledge:

(1) hasMother(Sarah, Marie).
(2) hasMother(John, Sarah).
(3) hasMother(x,y), hasMother(y,z) $\rightarrow$ hasGrandmother(x,z).
(4) mother subClassOf (Person Intersection Woman).
(5) Person $\subseteq \exists$mother.T.

If the query is: who hasGrandmother Marie? The system gets the answer: John hasGrandmother Marie, by reasoning with {(1), (2), (3)}.

If the query is: who is a subClassOf Woman? First, the system maps (4) into the following Horn clauses:

(6) mother(x) $\rightarrow$ Person(x)
(7) mother(x) $\rightarrow$ Woman(x)

Then reasoning with (7), the system gets the answer: mother is a subClassOf Woman.

If the query is: does Marie have a mother? First, reasoning with (1), the system gets

(8) mother(Marie).

By mapping (4) to (6) and (7) and unifying (8) and (6), the system then generates

(9) mother(Marie) $\rightarrow$ Person(Marie)

Finally, the system gets the answer: Marie has a mother, by reasoning with (5) and (9).

Obviously, the third query discussed above cannot be answered only by Prolog or OWL inference engine. However, in our method, the β-PSML solved the problem successfully by cooperatively using both of the two inference engines.

4. Experiments

Experiments in this section show how the β-PSML can be used for automatic reasoning on the Web by incorporating global information sources from the Semantic Web with locally operational knowledge-data bases in an enterprise portal.

4.1 A Camera Ontology for a Camera-Sale Portal

We use a sample of camera ontology to demonstrate our approach as shown in Figure 4. In this figure, the plain line denotes that the subclass relation holds between the connected nodes; the dashed line denotes slot of attribute and set of attributes; the circle object denotes a class; and the rectangle denotes an attribute owned by the class.

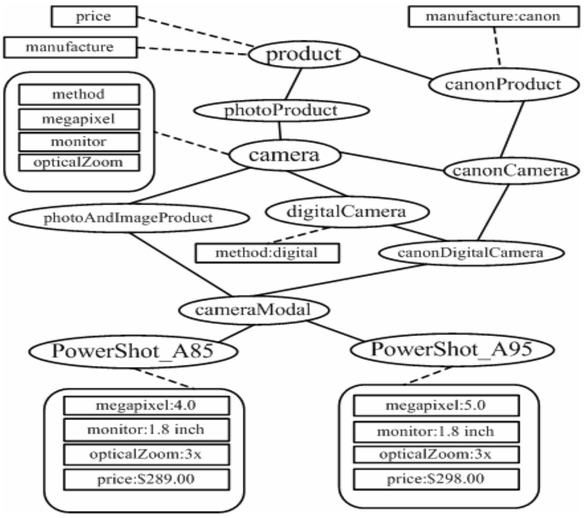

Fig. 4 A sample of camera Ontologies

Furthermore, the camera ontology is utilized to develop a camera-sale portal on the Semantic Web. Figure 5 shows the Web browser of this portal. In the camera-sale portal, we use RDF-Schema format for instances and the camera ontology in OWL [7] that builds upon the RDF and RDF Schema, so that XSLT can be employed to transform Web contents information on the Semantic Web (global information sources) [6].

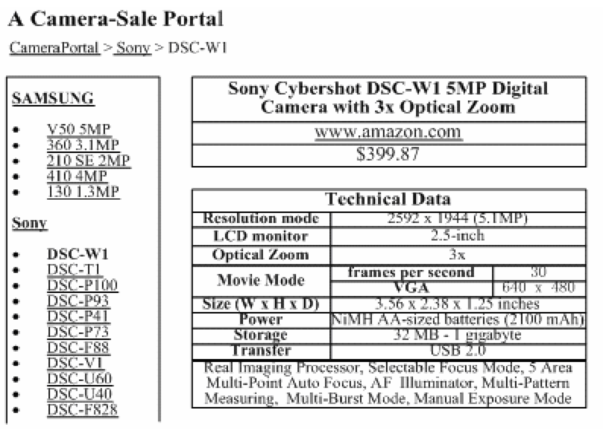

Fig. 5 The sample of camera-sale portals

4.2 Workflow of Transformation

Figure 6 shows a workflow of transformation from Web-contents information on the Semantic Web into the β-PSML. As discussed above, ontologies represented in OWL can be easily translated to β-PSML. The transformation process is the XSLT engine for analyzing an input source with ontologies (e.g. XML, RDF, and OWL) and for transforming it into the β-PSML as an output source. The examples of the input and output sources are shown in Figures 7-10, respectively.

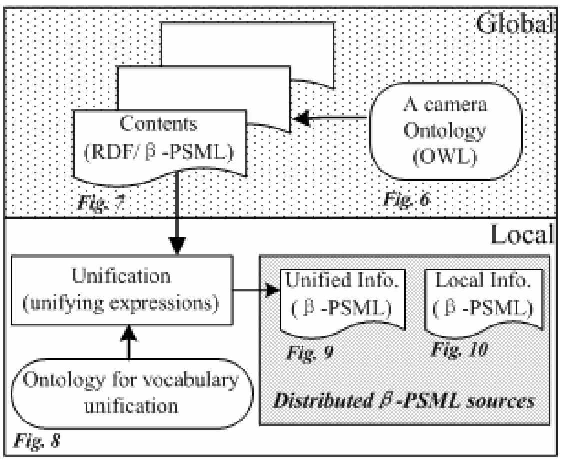

Fig. 6 Workflow of transformation

More specifically, the transformation collects Web contents information on the Semantic Web by Web information collecting agents. However, Web content information may not be afforded by camera manufacturer. For this reason, we utilized contents information (XML, OWL, the β-PSML) of our own which was drawn upon information of camera manufacturer's Web sites. These will be expected to be afforded by all manufacturers in the future.

The collected information is defined in the maverick style. Here are some problems: we must unify documentary form of the collected information because all manufacturers do not use the same format. Of course, if all manufacturers will use the unified format for contents, it will be easy to solve this problem. Unfortunately, it is difficult to ask for all manufacturers to use the same format.

Because the unification process of XML hierarchy is very difficult, we assumed that XML hierarchy is the same, but the used vocabularies may be different. We developed Unification for unifying vocabularies contained in Web contents information.

The workflow of Unification can be described as follows. First, Unification reads and analyzes the camera ontology denoted in OWL (see Figure 7) and instances denoted in RDF (see Figure 8). Secondly, Unification calls the XSLT engine and reads the ontology for vocabulary unification (see Figure 9). Finally, the output of Unification is the β-PSML file (see Figure 10) with a unified vocabulary.

4.3 Inference and Reasoning of the β-PSML in Experiments

In this subsection, we describe the inference and reasoning process of the β-PSML that are performed by transforming the global Semantic Web information sources and coupling with knowledge-data bases stored in the local-machine in advance. Part of the sample knowledge-data base in the local machine is shown in Figure 11. The knowledge-data base stores the information of the company itself and the old information of other companies.

We got ready for query by using the predicate find for the purpose of inferring with both local and global information sources. It can enumerate products by matching the criteria specified. Figure 12 shows the results with respect to query-answering by using find in the β-PSML.

In Figure 12, "Camera" is a variable of camera products. "megapixel (ge, 3.5)" is an assumption about 'megapixel is over 3.5', that means 'enumerate cameras that the megapixel is over 3.5'. When this query is given, the β-PSML reasons with global and local sources, and then gives answers about products that are acceptable. That is what the reasoning is carried out on the camera ontology and instances transformed from the Semantic Web with locally cameras-related operational knowledge-data bases.

If only local sources are employed for inference, the answers are incomplete (Figure 13). Hence, combining global and local information sources is very important for decision-making based on the Web inference engine.

```
<owl:Class rdf:ID="product">
  <rdfs:subClassOf>
    <rdfs:Restriction>
      <owl:onProperty rdf:resource="#price"/>
    </rdfs:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <rdfs:Restriction>
      <owl:onProperty
rdf:resource="#manufacturer"/>
    </rdfs:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="photoProduct">
  <rdfs:subClassOf rdf:resource="#product"/>
</owl:Class
```

Fig. 7 Part of the camera ontology in OWL

```
<rdf:Description rdf:ID="PowerShot A85">
   <rdf:typerdf:resource="http://www.w3.org/2000/0
1/rdf-schema#Class"/>
  <rdfs:subClassOf
rdf:resource="&camera;cameraModel"/>
  <camera:name>
     <xsd:string>
       <rdf:value>PowerShot A85</rdf:value>
     </xsd:string>
  </camera:name>
  <camera:megapixel>
     <xsd:string>
       <rdf:value>4.0</rdf:value>
     </xsd:string>
  </camera:megapixel>
  <camera:monitor>
     <xsd:string>
       <rdf:value>1.8 inch</rdf:value>
     </xsd:string>
  </camera:monitor >
</rdf:Description>
```

Fig. 8 Description of instances in RDF

```
<owlx :Class rdf:ID="name">
  <rdfs:seeAlso rdf:resource="#product"/>
  <rdfs:seeAlso rdf:resource="#product_name"/>
</owlx:Class>
<owlx:Class rdf:ID="megapixel">
  <rdfs:seeAlso rdf:resource="#mega_pixel"/>
  <rdfs:seeAlso rdf:resource="#resolution_mode"/>
  <rdfs:seeAlso rdf:resource="#pixel"/>
</owlx:Class>
<owl:Class rdf:ID="storage">
  <rdfs:seeAlso rdf:resource="#memory_stick"/>
  <rdfs:seeAlso rdf:resource="#memory"/>
</owlx:Class>
```

Fig. 9 A sample ontology for vocabulary unification in OWL

```
<prolog:clause>
  <prolog:head>
    <owls:Class rdf:ID="canonProduct">
       <rdfs:subClassOf>
         <owls:Restriction>
           <owls:hasValue
rdf:datatype="&xsd;string">canon</owls:hasValue>
           <owls:onProperty
rdf:resource="manufacturer"/>
         </owls:Restriction>
       </rdfs:subClassOf>
       <rdfs:subClassOf>
         <owls:Class rdf:ID="product"/>
       </rdfs:subClassOf>
    </owls:Class>
         …
      <cameraModel rdf:ID="PowerShot_A95">
        <price
rdf:datatype="&xsd;string">$398.00</price>
        <opticalZoom
rdf:datatype="&xsd;string">3x</opticalZoom>
        <megapixel
rdf:datatype="&xsd;string">5.0</megapixel>
        <monitor    rdf:datatype="&xsd;string">1.8
inch</monitor>
      </cameraModel>
   </prolog:head>
</prolog:clause>
```

Fig. 10 The sample of camera Ontologies transformed to the β-PSML

```
<cameraModel rdf:ID="DSC-W1">
  <price
rdf:datatype="&xsd;string">$387.00</price>
  <opticalZoom
rdf:datatype="&xsd;string">3x</opticalZoom>
  <megapixel
rdf:datatype="&xsd;string">5.26</megapixel>
  <monitor         rdf:datatype="&xsd;string">2.5
inch</monitor>
</cameraModel>
<cameraModel rdf:ID="DSC-T1">
  <price
rdf:datatype="&xsd;string">$589.00</price>
  <opticalZoom
rdf:datatype="&xsd;string">3x</opticalZoom>
  <megapixel
rdf:datatype="&xsd;string">5.1</megapixel>
```

```
  <monitor         rdf:datatype="&xsd;string">2.5
inch</monitor>
</cameraModel>
<cameraModel rdf:ID="DSC- P72">
  <price
rdf:datatype="&xsd;string">$254.00</price>
  <opticalZoom
rdf:datatype="&xsd;string">3x</opticalZoom>
  <megapixel
rdf:datatype="&xsd;string">3.2</megapixel>
  <monitor         rdf:datatype="&xsd;string">1.5
inch</monitor>
</cameraModel>
```


Fig. 11 Part of the local knowledge-data base

```
?- find camera megapixel (ge, 3.5)
camera PowerShot_A95
megapixel=5.0
------------------------------
camera PowerShot_A85
megapixel=4.0
------------------------------
camera DSC-W1 megapixel=5.26
------------------------------
camera DSC-T1 megapixel=5.1
```

Fig. 12 Query-answering in find by using both global and local information sources

```
?- find camera megapixel (ge, 3.5)
camera DSC-W1
megapixel=5.26
------------------------------
camera DSC-T1
megapixel=5.1
```

Fig. 13 Query-answering in find by only using local information source

5. Related Work

There have been several works addressing the combining OWL with Horn clauses or other logic systems for problem solving and rule markup languages on the Semantic Web. They include SWRL [12], XRML [14], Description Logic Programs [11], AL-log [8], CARIN [16], and the W4 project [2].

SWRL constructed a Semantic Web Rule Language by combining OWL and RuleML, but its number of predicate arguments was limited to only one or two and also its reasoning mechanism was not considered. In [14], Kang and Lee observed that there exist rules embedded implicitly in Web pages that cannot be processed in XML, and proposed XRML that supports the automatic processing of implicit rules embedded in the hypertexts and helps human's browsing for their comprehension. However, XRML only utilizes global information and does not consider the ontology. In [11], Grosof et al. studied the intersection of DL and Horn clauses. However, they did not consider other conditions. The combination of DL with Horn clauses was studied in AL-log and CARIN, but the ontology was not considered. The W4 (Well-Founded Semantics for the WWW) project aims at developing Standard Prolog interoperable tools for supporting distributed, secure, and integrated reasoning activities on the Semantic Web, but it did not also consider the ontology. Compared with other work, our research concentrates on the combination of OWL with Horn clauses, and the PSML provides capabilities to represent, transform and manage multiple, huge knowledge-data sources on the Web and the knowledge grid for distributed Web inference and reasoning as well as complex adaptive, distributed problem solving.

6. Conclusions

The β-PSML that we proposed is a new method of knowledge representation for the Semantic Web. The β-PSML can be easily used for inference and reasoning as well as transforming and managing both global and local knowledge-data base. The β-PSML can perform the function of logic layer for the Semantic Web. The experimental results show that our considerations are valid and our preliminary solution works well on the Semantic Web.

In the future work, we will consider how to use the knowledge with a hierarchical structure and combine various reasoning methods in PSML more efficiently and effectively for complex adaptive, distributed problem solving. Furthermore, distributed Web inference engines need to be implemented on the wisdom Web and knowledge grids (include Web service technology) to deal with huge and multiple distributed information sources [9, 10, 18, 25, 26].

Acknowledgments

This work is supported partially by the NSFC major research program: "Basic Theory and Core Techniques of Non-Canonical Knowledge" (60496322), the Natural Science Foundation of Beijing Municipality (4022003), and the Open Foundation of Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology.

References

[1] H.P. Alesso and C.F. Smith: *The Intelligent Wireless Web*, Addison-Wesley, 2002.

[2] J.J. Alferes, C.V. Damasio, and L.M. Pereira: Semantic Web Logic Programming Tools. F. Bry et al. (eds.) *Principles and Practice of SemanticWeb Reasoning*, Springer (2003) 16-32.

[3] S. Bechhofer, P. Lord, and R. Volz: Cooking the Semantic Web with the OWL API. *2nd International Semantic Web Conference*, ISWC, Sanibel Island, Florida, Oct. 2003. http://owl.man.ac.uk/api.shtml.

[4] A. Borgida: On the Relative Expressiveness of Description

Logics and Predicate Logics. *Artificial Intelligence*, 82(1-2): 353-367, 1996.

[5] W. Cheung, J. Liu, K. Tsang, and H. Wong: Goal-Directed Service Flow Planning Using Metaknowledge on Services, *Proc. Second International Workshop on Knowledge Grid and Grid Intelligence*, Beijing, China (2004) 123-130.

[6] J. Clark: *XSL Transformations (XSLT) Version 1.0*, http://www.w3.org/TR/xslt.

[7] M. Dean, G. Schreiber, S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, and L. Andrea Stein: *OWL Web Ontology Language Reference*, http://www.w3.org/TR/2004/RECowl-ref-20040210/.

[8] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf: AL-log: Integrating Datalog and Description Logics. Journal of Intelligent Information Systems, 10: 227-252, 1998.

[9] I. Foster and C. Kesselman: The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2004.

[10] I. Foster, N.R. Jennings, and C. Kesselman: Brain Meets Brawn: Why Grid and Agents Need Each Other. Proc. 3rd International Conference on Autonomous Agents and Multi-Agent Systems, New York, USA (2004) 8-15.

[11] B. Grosof, I. Horrocks, R. Volz, and S.Decker: Description Logic Programs: Combining Logic Programs with Description Logic, WWW2003 Foundations of the Semantic Web, ACM Press, NY, USA (2003) 48-57.

[12] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, Draft Version 0.7 of 21 Dec. 2004, http://www.daml.org/rules/proposal/.

[13] J. Hu, and N. Zhong: Organizing Dynamic Multi-level Workflows on Multi-layer Grids for e-Business Portals Development. Proc. Second International Workshop on Knowledge Grid and Grid Intelligence (2004) 48-56.

[14] J. Kang, and J.K. Lee: Extraction of Structured Rules from Web pages and Maintenance of Mutual Consistency: XRML Approach. Ontologies, Rules and Rule Markup Languages for the Semantic Web, LNAI 2876, Springer (2003) 150-163.

[15] M. Kifer, G. Lausen, and J. Wu: Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the ACM, 42: 741-843, 1995.

[16] A. Levy and M-C. Rousset: Combining Horn Rules and Description Logics in CARIN. Artificial Intelligence, 104(1-2): 165-209, 1998.

[17] C. Liu: A Partial Deduction Procedure Based on Abstract Interpretations. Chinese Journal of Software, 5(11): 31-37, 1994.

[18] J. Liu: Web Intelligence (WI): What Makes Wisdom Web? Proc. Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03) (2003) 1596-1601. [19] J. Liu, X. Jin, and Y. Tang: Multi-agent Collaborative Service and Distributed Problem Solving. Cognitive Systems Research, 5(3): 191-206, Elsevier, 2004.

[20] P. Raghavan: Social Networks. From the Web to the Enterprise. IEEE Internet Computing, 6(1): 91-94, 2002. [21] K. Tomita, N. Zhong, and H. Yamauchi: Coupling Global Semantic Web with Local Information Sources for Problem Solving, Proc. First International Workshop on Semantic Web Mining and Reasoning, Beijing, China (2004) 66-74.

[22] N. Zhong, J. Liu, and Y.Y. Yao: In Search of the Wisdom Web. IEEE Computer, 35(11): 27-31, 2002.

[23] N. Zhong: Toward Web Intelligence, E. Menasalvas et al (eds.) Advances in Web Intelligence, LNAI 2663, Springer (2003) 1-14.

[24] N. Zhong, J. Liu, and Y. Y. Yao (eds.) Web Intelligence, Springer, 2003.

[25] N. Zhong, J.L. Wu, and C. Liu: Building a Data Mining Grid for Multiple Human Brain Data Analysis. Proc. International Workshop on Knowledge Grid and Grid Intelligence, Halifax, Canada (2003) 29-35.

[26] N. Zhong: Developing Intelligent Portals by Using WI Technologies. J.P. Li et al (eds.) Wavelet Analysisand its Applications, and Active Media Technology, 2, World Scientific (2004) 555-567.

Yila Su received the B.S. and M.S. degrees from Tsinghua University in 1986 and 1989, respectively. He is currently an associate professor in Inner Mongolia University of Technology. His research interests are knowledge discovery and data mining, rough sets and granular-soft computing, Web intelligence, intelligent agents, and knowledge information systems.