

Schema Theory for Semantic Link Network

Hai Zhuge^{#1}, Yunchuan Sun^{#2*}, and Junsheng Zhang^{#3}

[#]Knowledge Grid Research Group, Key Lab of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

¹zhuge@ict.ac.cn, ²sun@kg.ict.ac.cn, ³junsheng@kg.ict.ac.cn

^{*}School of Economics & Business Administration, Beijing Normal University, Beijing 100875, China
yunch@bnu.edu.cn

Abstract—Semantic Link Network (SLN) is a loosely coupled semantic data model for managing Web resources. Its nodes can be any types of resources. Its edges can be any semantic relations. Potential semantic links can be derived out according to reasoning rules on semantic relations. This paper proposes the schema theory for SLN including the concepts, rule-constraint normal forms, and relevant algorithms. The theory provides the basis for normalized management of SLN and its applications. A case study demonstrates the proposed theory.

I. INTRODUCTION

The schema of relational database defines the structure of database. It defines a set of relations with attributes and the dependencies among attributes. The normalized theory of the relational schema is to ensure high consistency, low redundancy and better efficiency [12, 13]. Relational data model is limited in representing rich semantic relationships between resources and supporting reasoning on semantic relations.

The Semantic Web aims at making Web resources machine-understandable by enriching semantics in the resources [9, 15, 16]. XML (eXtensible Markup Language) is to describe the structure in Web resources for cross-platform machine-understandability [4]. XML schema defines a set of syntaxes and rules to express the shared vocabularies [2]. It provides a means for defining the structure, content and semantics of XML documents in more detail. Based on XML, many markup languages have been proposed. RDF (Resource Description Framework) focuses on describing the universal resources on the Web by an object-attribute-value triple [5, 20]. RDF Schema (RDFS) defines a set of syntaxes to store the metadata of resources with XML syntax and provides basic RDF vocabularies for structuring RDF resources [10, 19]. RDFS is still weak in expressing rich semantic relationships and supporting relational reasoning. OWL (Web Ontology Language) is designed to describe the semantics of the resources themselves with ontologies and semantic relationships with roles [3, 8, 21, 22]. It can represent the meaning of terms in vocabularies explicitly and the relationships between those terms. Its logical foundation is description logics which has the decidability of ontology consistency [6, 7, 17]. The Rule Markup Language (RuleML) is to express rules in XML for deduction, rewriting, and further inferential-transformational tasks [1, 11]. The Semantic Web Rule Language (SWRL) is based on a combination of the OWL with RuleML [18]. Compared with the relational data model, the Semantic Web

is still weak in theory.

Semantic Link Network (SLN) is a loosely coupled semantic data model for managing Web resources with the following main features of the Web:

- Easy to build and easy to use; and,
- Any semantic node can semantically link to any other semantic node.

An SLN instance is a graph denoted as $(ResourceSet, LinkSet)$, where $ResourceSet$ is a set of resources, and $LinkSet$ is a set of semantic links in form of $R_1 \xrightarrow{\alpha} R_2$, where $R_1, R_2 \in ResourceSet$, and α is a semantic factor representing a semantic relation between R_1 and R_2 .

A set of reasoning rules on semantic links enables SLN to derive out potential semantic links. The basic concepts and model of SLN have been introduced in [25-31]. More references are available at www.knowledgegrid.net/~h.zhuge/SLN.htm.

A schema of SLN can be defined to specify resource types, semantic link types, and reasoning rules on semantic links in a domain. Fig.1 describes the scenario of developing domain SLNs. A global SLN schema reflecting the consensus on the basic semantics of the domain should be defined first. Users can define SLN instances by instantiating the SLN schema or define a sub-schema first and then instantiating the sub-schema.

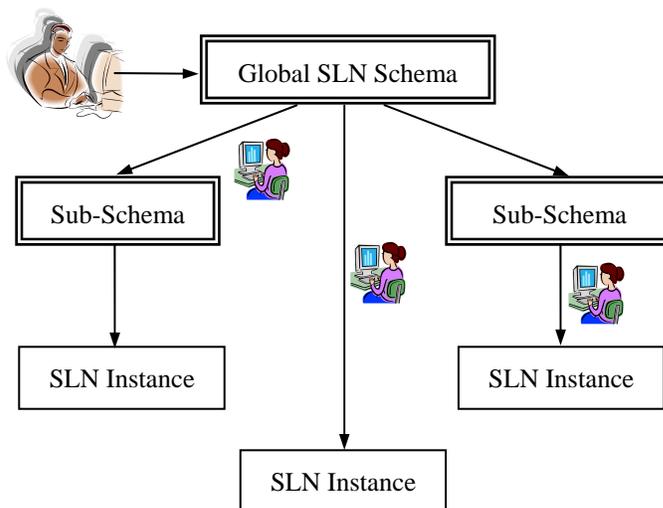


Fig.1 The role of SLN schema.

II. SCHEMA FOR SEMANTIC LINK NETWORK

The schema of SLN is a triple $\hat{S}(\text{ResourceTypes}, \text{LinkTypes}, \text{Rules})$, where ResourceTypes is a set of resource types denoted as $\{rt_1, rt_2, \dots, rt_k\}$, LinkTypes is a set of semantic link types, each takes the form of $rt_1 \xrightarrow{\alpha} rt_2$,

where $rt_1, rt_2 \in \text{ResourceTypes}$, and α is a semantic factor that can be denoted by tags representing commonsense among users, and Rules is a set of reasoning rules on semantic links [25]. A resource type defines the basic semantics of a class of resources by a set of attributes. For example, *title*, *abstract*, *author*, *pubDate*, and *length* can be the attributes of the *paper* type.

The possible semantic relationship types between two resources are determined by the types of the start resource and the end resource. For example, the possible types of a semantic link between a researcher and a paper are *authorOf* and *readerOf*, but not *fatherOf*. So a semantic link with semantic factor α from a resource type rt_1 to another resource type rt_2 can be described as $rt_1 \xrightarrow{\alpha} rt_2$. For two resource types rt_1 and rt_2 , we use $[rt_1, rt_2]$ to denote the set of all semantic link types with the start resource type rt_1 and the end resource type rt_2 . For example, $[\text{researcher}, \text{paper}] = \{ \text{researcher} \xrightarrow{\text{author}} \text{paper}, \text{researcher} \xrightarrow{\text{reader}} \text{paper}, \text{researcher} \xrightarrow{\text{editor}} \text{paper} \}$. In abbreviation, $[\text{researcher}, \text{paper}] = \{ \text{author}, \text{reader}, \text{editor} \}$.

For a pair of resource types, relationships between semantic link types can be classified into three categories.

1. *Implication*, denoted as $\alpha \Rightarrow \beta$, i.e., a semantic link type α implies semantic link type β between the same pair of resources as shown in Fig.2(d).
2. *Compatible*. Two semantic link types α and β do not affect each other.
3. *Incompatible*. Two semantic link types α and β cannot co-occur between the same pair of resources.

For a semantic link $R_1 \xrightarrow{\omega} R_2$ between two resources R_1 and R_2 , the reversion is a semantic link in form of $R_2 \xrightarrow{\omega^{-1}} R_1$, which means that if there is a semantic relationship ω from R_1 to R_2 , then there is a semantic relationship ω^{-1} from R_2 to R_1 [27]. Obviously, $[rt_1, rt_2]$ and $[rt_2, rt_1]$ are equivalent.

A reasoning rule takes the following form as shown in Fig.2 (a): $R_1 \xrightarrow{\alpha} R_2, R_2 \xrightarrow{\beta} R_3 \Rightarrow R_1 \xrightarrow{\gamma} R_3$ denoted as $\alpha\beta \Rightarrow \gamma$ in abbreviation. Fig.2 (b) and (c) show the following two forms of reasoning rule:

$$R_1 \xrightarrow{\alpha} R_2, R_1 \xrightarrow{\beta} R_3 \Rightarrow R_2 \xrightarrow{\gamma} R_3 \Leftrightarrow \\ R_2 \xrightarrow{\alpha^{-1}} R_1, R_1 \xrightarrow{\beta} R_3 \Rightarrow R_2 \xrightarrow{\gamma} R_3$$

i.e., $\alpha^{-1} \cdot \beta \Rightarrow \gamma$.

$$R_2 \xrightarrow{\alpha} R_1, R_3 \xrightarrow{\beta} R_1 \Rightarrow R_2 \xrightarrow{\gamma} R_3 \Leftrightarrow \\ R_2 \xrightarrow{\alpha} R_1, R_1 \xrightarrow{\beta^{-1}} R_3 \Rightarrow R_2 \xrightarrow{\gamma} R_3$$

i.e., $\alpha\beta \Rightarrow \gamma$.

Lemma 1. The following two rules hold:

1. $\alpha \cdot \beta \Rightarrow \gamma \Leftrightarrow \beta^{-1} \cdot \alpha^{-1} \Rightarrow \gamma^{-1}$, and
2. $\alpha^{-1} \cdot \beta \Rightarrow \gamma \Leftrightarrow \beta^{-1} \cdot \alpha \Rightarrow \gamma^{-1}$.

Fig.2(d) shows the following reasoning rule form: $R_1 \xrightarrow{\alpha} R_2 \Rightarrow R_1 \xrightarrow{\beta} R_2$, in simple $\alpha \Rightarrow \beta$, which means that the semantic relationship α is stronger than the semantic relationship β between two resources. For example, $\text{Researcher} \xrightarrow{\text{editorOf}} \text{Paper} \Rightarrow \text{Researcher} \xrightarrow{\text{readerOf}} \text{Paper}$ means that there must be a semantic link *readerOf* between a researcher and a paper if there is an *editorOf* semantic link between researcher and paper.

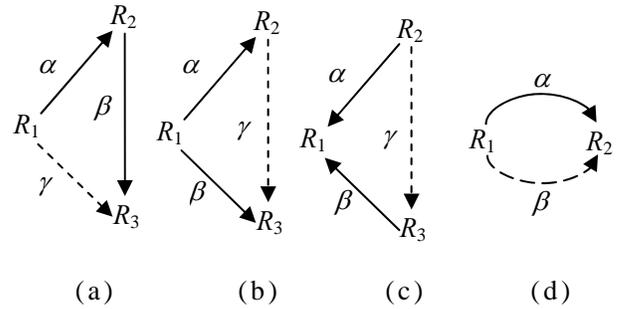


Fig.2 Basic forms of reasoning rule.

Proposition 1. For a rule $\alpha \Rightarrow \beta$, if $\alpha \in [rt_1, rt_2]$, then $\beta \in [rt_1, rt_2]$.

Proposition 2. For a rule $\alpha\beta \Rightarrow \gamma$, if $\alpha \in [rt_1, rt_2]$, $\beta \in [rt_2, rt_3]$, then $\gamma \in [rt_1, rt_3]$.

The consistency of the rule set is important for SLN reasoning as arbitrarily adding rules to the rule set of SLN may lead to inconsistency. The following are two cases of inconsistent rules.

1. The Rules includes two rules $\alpha \Rightarrow \beta_1$ and $\alpha \Rightarrow \beta_2$ but β_1 is incompatible to β_2 .
2. The Rules includes two rules $\alpha\beta \Rightarrow \gamma_1$ and $\alpha\beta \Rightarrow \gamma_2$, but γ_1 is incompatible to γ_2 .

It is easy to develop an algorithm to avoid inconsistency in a rule set. Moreover, redundant rules can be removed from the rule set by using the algorithms introduced in [32].

For two SLN schemas $\hat{S}(R, L, rs)$ and $\hat{S}'(R', L', rs')$, if $R' \subseteq R$, $L' \subseteq L$, and $rs' \subseteq rs$, we call \hat{S}' is a subschema of \hat{S} , denoted as $\hat{S}' \subseteq \hat{S}$. The intersection schema of $\hat{S}_1(R_1, L_1, rs_1)$ and $\hat{S}_2(R_2, L_2, rs_2)$ can be defined as $\hat{S}(R_1 \cap R_2, L_1 \cap L_2, rs_1 \cap rs_2)$, which clearly is a common subschema of \hat{S}_1 and \hat{S}_2 . The union schema of \hat{S}_1 and \hat{S}_2 can be defined as $\hat{S}(R_1 \cup R_2, L_1 \cup L_2, rs_1 \cup rs_2)$. However, execution of the union operation of two schemas requires that the two rule set are consistent with each other.

III. RULE-CONSTRAINT NORMAL FORMS FOR SLN SCHEMA

The following operation is to filter out the useful component from the given schema.

Definition 1. (Projection) Let $\hat{S}(R, L, rs)$ and $\hat{S}'(R', L', rs')$ be two SLN schemas, and S be an instance of \hat{S} . The projection of S with respect to \hat{S}' , denoted as $S' = \Pi_{\hat{S}'}(S)$,

is a new SLN derived from S by removing all resources whose types are not in R' and removing all semantic links whose types are not in L' , and the reasoning in \hat{S}' is executed according to the rule set rs' .

Especially for above definition, if $R=R'$, the projection, denoted as $\Pi_L(S)$, can be derived by keeping all resources and deleting all semantic links whose types are not in L' . Similarly, if $L=L'$, then the projection, denoted as $\Pi_{R'}(S)$, can be derived by removing all resources whose types are not in R' and removing all semantic links between the deleted resources.

A reasoning rule represents the semantic relevance among semantic link types. For example, the rule *editor* \Rightarrow *reader* shows that an *editor* of a paper should be a *reader* of the paper. Similarly, *fatherOf*:*brotherOf* \Rightarrow *fatherOf* shows that *brotherOf* is relevant to *fatherOf*.

Definition 2. (Semantic Relative) Let $\hat{S}(R, L, rs)$ be a schema of SLN, $\alpha_1, \alpha_2, \alpha_3 \in L$, α_1 is called direct relative to α_2 (denoted as $\alpha_1 \triangleleft \alpha_2$ or $\alpha_2 \triangleleft \alpha_1$) if there is a rule in rs with one of the following forms $\alpha_2 \Rightarrow \alpha_1$, $\alpha_2 \cdot \alpha_3 \Rightarrow \alpha_1$, or $\alpha_3 \cdot \alpha_2 \Rightarrow \alpha_1$.

If there is a sequence of semantic relative $\alpha_1 \triangleleft \alpha_2 \triangleleft \alpha_3 \triangleleft \dots \triangleleft \alpha_m$ (m is an integer), then α_1 is called relative to α_m , denoted as $\alpha_1 \triangleleft \alpha_m$. We can construct a semantic relative net as shown in Fig.2 for an SLN schema by drawing an arrow from α_i to α_j for each semantic relative $\alpha_j \triangleleft \alpha_i$ retrieved from the rule set.

Definition 3. (Closure under Semantic Relative) Let $\hat{S}(R, L, rs)$ be a schema of SLN, and $M \subseteq L$ be a set of semantic link types. The closure of M , denoted as $C(M)$, under semantic relative is a set of semantic link types derived from the following steps.

1. Let $C(M) = M$.
2. Check for each semantic dependence $\alpha_i \triangleleft \alpha_j$, if $\alpha_i \in C(M)$, let $C(M) = C(M) \cup \{\alpha_j\}$; and, if $\alpha_j \in C(M)$, let $C(M) = C(M) \cup \{\alpha_i\}$.
3. Repeat from step 2 until $C(M)$ keeps unchanging.

It is easy to verify that all closures of the single semantic link types construct a classification for the semantic link types of the SLN schema, and we can decompose an SLN schema based on such a classification. Actually, each disjoint part in the semantic link relative net forms a closure.

For an isolated semantic link type α , the closure includes only itself, i.e., $C(\alpha) = \{\alpha\}$. For a schema of semantic link network $\hat{S}(R, L, rs)$, there might be a cycle of semantic relatives $\alpha_1 \triangleleft \alpha_2 \triangleleft \alpha_3 \triangleleft \dots \triangleleft \alpha_m \triangleleft \alpha_1$, $\alpha_1, \alpha_2, \dots, \alpha_m \in L$, called *semantic relative cycle*. It is easily to verify the following lemma.

Lemma 2. Let $\alpha_1 \triangleleft \alpha_2 \triangleleft \alpha_3 \triangleleft \dots \triangleleft \alpha_m \triangleleft \alpha_1$ be a semantic relative cycle, where $\alpha_i \in L$ ($1 \leq i \leq m$). We have:

1. $C(\alpha_i)$ is identical to $C(\alpha_j)$, for all $1 \leq i, j \leq m$.
2. $\alpha_i \in C(\alpha_j)$, for any i, j , $1 \leq i, j \leq m$.
3. For any $\alpha \in L$, if $\alpha \triangleleft$ (resp. \triangleright) α_i for some i ($1 \leq i \leq m$), then $\alpha \triangleleft$ (resp. \triangleright) α_j for any j ($1 \leq j \leq m$).

All semantic link types at one semantic relative cycle construct an equivalent class according to lemma 2. A cycle $\alpha_1 \triangleleft \alpha_2 \triangleleft \alpha_3 \triangleleft \dots \triangleleft \alpha_m \triangleleft \alpha_1$ in the semantic link type relative net can be regarded as a unit, denoted as $U(\alpha_i)$, where α_i is any semantic link type in the cycle. Therefore, a semantic relative cycle is shrunk into a node (all out and in arrows in the cycle will be focused on the node). We can find all cycles in the net by using the classic algorithms which find the cycles in a directed map [14]. In the following discussion, we do not mention the cycles for they are regarded as single semantic link types. And, this operation should be executed before other operations. The following definition normalizes the schema of SLN.

Definition 4. A SLN schema $\hat{S}(R, L, rs)$ is in Rule-Constraint Normal Form 1 (RC-NF 1), if for any semantic link type $\alpha \in L$, $C(\alpha) = L$ holds.

Lemma 3. For SLN schema $\hat{S}(R, L, rs)$ with RC-NF1 and $\alpha_1, \alpha_2 \in L$, then $C(\alpha_1) = C(\alpha_2)$.

Any SLN schema $\hat{S}(R, L, rs)$ can be decomposed into several sub-schemas satisfying RC-NF1 by the following algorithm.

Algorithm 1. Let $\hat{S}(R, L, rs)$ be an SLN schema.

1. Compute the classification of the semantic link type schema according to the definition of the closure of semantic relative and the rule set rs denoted as $\{L_1, L_2, \dots, L_k\}$.
2. For each L_i , $1 \leq i \leq k$, we can get a resource type set R_i , where each is attached with the semantic link types in L_i , and a rule set rs_i where each rule is only involved in semantic link types from L_i . Clearly, $R_i \subseteq R$, $L_i \subseteq L$, and $rs_i \subseteq rs$. Thus, $\hat{S}_i(R_i, L_i, rs_i)$ is a sub-schema of $\hat{S}(R, L, rs)$.
3. $\hat{S}(R, L, rs)$ is decomposed into k sub-schemas $\hat{S}_i(R_i, L_i, rs_i)$, where $1 \leq i \leq k$.

The classification of the schema of semantic link type can be easily found from the semantic relative net actually.

Different unconnected parts determine different sub-schemas. For a RC-NF1 SLN schema, its semantic relative net is a connected map. That means the reasoning within such an SLN is closed. However, this does not mean that any two semantic link types are semantically related.

Let $\hat{S}(R, L, rs)$ be an SLN schema, $\alpha \in L$ is called a top semantic link type if there is no semantic link type α' ($\neq \alpha$) such that $\alpha \triangleleft \alpha'$. α is called a bottom semantic link type if there is no semantic link type α' ($\neq \alpha$) such that $\alpha' \triangleleft \alpha$. Two top semantic link types or two bottom ones are not semantic relative. α is called an isolated semantic link type if there is no semantic link type α' ($\neq \alpha$) such that $\alpha \triangleleft \alpha'$ or $\alpha' \triangleleft \alpha$. An isolated semantic link type is both top and bottom. We overlook isolated semantic link types in most reasoning cases and exclude them from the top and bottom semantic link types because they do not participate in reasoning with other semantic link types. The following algorithm finds all bottom semantic link types for an SLN schema.

Algorithm 2. For an SLN schema $\hat{S}(R, L, rs)$, find the set B of all bottom semantic link types in \hat{S} .

1. Let $B = \{\alpha \mid \alpha_i \cdot \alpha_j \Rightarrow \alpha \in rs\}$, where $\alpha_i, \alpha_j \in L$;
2. Loop for each $\alpha_j \in B$. Check each rule r in rs , if α occurs in the precondition of r and does not occur in the post-condition of r , remove α from B , continue to check the next link type; else check next rule in rs .

The bottom semantic link types cannot affect other link types in reasoning. We can compute all semantic link types that affect a certain semantic link type.

Definition 5. Let $\hat{S}(R, L, rs)$ be an SLN schema, and $\alpha \in L$. The up-closure of α with respect to the rule set rs , denoted as $C_{up}(\alpha)$, is a set of semantic link types derived from the following steps.

1. Let $C_{up}(\alpha) = \{\alpha\}$;
2. Check L , for each semantic relative $\alpha_i \triangleleft \alpha_j$, if $\alpha_i \in C_{up}(\alpha)$, let $C_{up}(\alpha) = C_{up}(\alpha) \cup \{\alpha_j\}$;
3. Repeat from step 2 until $C_{up}(\alpha)$ does not change.

Lemma 4. Let $\alpha_1 \triangleleft \alpha_2 \triangleleft \alpha_3 \triangleleft \dots \triangleleft \alpha_m \triangleleft \alpha_1$ be a semantic relative circle, then

1. $C_{up}(\alpha_i)$ is identical to $C_{up}(\alpha_j)$ for all $1 \leq i, j \leq m$.
2. $\alpha_j \in C_{up}(\alpha_i)$, for any i and j , $1 \leq i, j \leq m$.

Definition 6. An SLN schema $\hat{S}(R, L, rs)$ is in Rule-Constraint Normal Form 2 (RC-NF2), if it satisfies the following conditions.

1. $\hat{S}(R, L, rs)$ is RC-NF1; and,
2. Let α be a bottom semantic link type in L , for any other link type β , we have $\alpha \triangleleft \beta$.

Obviously, for a RC-NF2 schema, the up closure of the bottom semantic link type α according to the rule set rs is just L , i.e., $L = C_{up}(\alpha)$.

Lemma 5. For a RC-NF2 SLN schema $\hat{S}(R, L, rs)$, it has a *unique* bottom semantic link type.

Proof. Assume that there are at least two bottom semantic link types $\alpha_{b1} \neq \alpha_{b2}$. From definition 3, $C_{up}(\alpha_{b1}) = C_{up}(\alpha_{b2})$. So $\alpha_{b1} \in C_{up}(\alpha_{b2})$, $\alpha_{b2} \triangleleft \alpha_{b1}$. Similarly, $\alpha_{b1} \triangleleft \alpha_{b2}$. It leads to a contradiction, so the assumption is false.

We can decompose a RC-NF1 SLN schema into several RC-NF2 sub-schemas according to the following algorithm.

Algorithm 3. Let $\hat{S}(R, L, rs)$ be a RC-NF1 SLN schema.

1. Find all bottom semantic link types of \hat{S} , denoted as $\alpha_{b1}, \alpha_{b2}, \dots$, and α_{bm} .
2. For each α_{bi} ($1 \leq i \leq m$), compute its up-closure according to reasoning rule set rs , denoted as $C_{up}(\alpha_{bi})$. Then, we can get a resource type set R_{bi} in which each element is involved with at least one link type in L_{bi} , and a rule set rs_i in which each rule is involved only with link types from L_{bi} . Clearly, $R_{bi} \subseteq R, L_{bi} \subseteq L$, and $rs_{bi} \subseteq rs$. Thus, $\hat{S}_{bi}(R_{bi}, L_{bi}, rs_{bi})$ is a sub-schema of $\hat{S}(R, L, rs)$.
3. $\hat{S}(R, L, rs)$ can be decomposed into k sub-schemas $\hat{S}_{bi}(R_{bi}, L_{bi}, rs_{bi})$, where $1 \leq i \leq m$.

Lemma 6. For an SLN schema $\hat{S}(R, L, rs)$, let $\hat{S}_1(R_1, L_1, rs_1)$ and $\hat{S}_2(R_2, L_2, rs_2)$ be two RC-NF2 sub-schemas. If $\alpha \in \hat{S}_1 \cap \hat{S}_2$, $C_{up}^{(1)}(\alpha)$ in \hat{S}_1 is identical to $C_{up}^{(2)}(\alpha)$ in \hat{S}_2 .

Proof. For a semantic link type $\alpha_0 \in C_{up}^{(1)}(\alpha)$, $\alpha_0 \triangleleft \alpha$. Let α_{b1} be the bottom semantic link types for \hat{S}_1 and α_{b2} for \hat{S}_2 according to Lemma 4. Obviously, $\alpha_{b1} \triangleleft \alpha$ and $\alpha_{b2} \triangleleft \alpha$. Thus, we get that $\alpha_{b2} \triangleleft \alpha_0$, which means that $\alpha_0 \in C_{up}^{(2)}(\alpha)$. \square

Lemma 7. $\hat{S}_1(R_1, L_1, rs_1)$ and $\hat{S}_2(R_2, L_2, rs_2)$ are two RC-NF2 sub-schemas of $\hat{S}(R, L, rs)$. Let $E = L_1 \cap L_2$, if $E \neq \emptyset$, then there is no any semantic relative $\alpha \triangleleft \beta$ satisfying that $\alpha \in E$ and $\beta \notin E$.

Proof. For $\alpha \in E$, and a semantic relative $\alpha \triangleleft \beta$. $E = L_1 \cap L_2$, then $\alpha \in L_1$. For L_1 is the closure of a bottom link type according to the semantic relatives, then $\beta \in L_1$. Similarly, we can get $\beta \in L_2$. So $\beta \in E$. \square

Reasoning is closed in a RC-NF2 SLN schema. SLNs based on different RC-NF2 sub-schemas of the same original schema are reasoning closed and independent. And, reasoning service is more efficient and easier to execute in a sub-schema than in the original one.

For an application, the whole schema might include several SLN sub-schemas with RC-NF2.

Definition 7. Let $\hat{S}(R, L, rs)$ be a SLN schema, and $\alpha \in L$ be a semantic link type. The down closure of α with respect to the rule set rs , denoted as $C_{down}(\alpha)$, is a set of semantic link types

derived from the following steps.

1. Let $C_{down}(\alpha) = \{\alpha\}$;
2. Check L , for each semantic dependence $\alpha_j \triangleleft \alpha_i$, if $\alpha_i \in C_{down}(\alpha)$, let $C_{down}(\alpha) = C_{down}(\alpha) \cup \{\alpha_j\}$;
3. Repeat from step 2 until $C_{down}(\alpha)$ does not change.

Intuitively, the down closure for a link type α is the set of all link types which can be affected by α . And, we can easily get the following characters.

Lemma 8. For a bottom link type α_b in the SLN schema $\hat{S}(R, L, rs)$, the down closure $C_{down}(\alpha_b)$ construct a tree with root α_b in the semantic link type relative net.

Lemma 9. For two bottom link types α_1 and α_2 in the SLN schema $\hat{S}(R, L, rs)$, if $C_{down}(\alpha_1) \cap C_{down}(\alpha_2) \neq \emptyset$, then α_1 and α_2 are in the same RC-NF2 sub-schemas.

IV. SCHEMA MAINTENANCE AND REASONING

A. SCHEMA MAINTENANCE

The updating for an SLN schema involves in resource types, semantic link types, and reasoning rules. The essence of semantic rule-constraint normal form is to classify the semantic link type set into different parts according to the rule set. So only the updating of the rule set can lead to different sub-schemas. The resource type set and semantic link type set for the sub-schemas will change respectively. Therefore, we can study the updating according to the variation of the rule set.

The following algorithm is to get a new version of decomposition after extension with a new rule.

Algorithm 4. (Schema Revision after Extension) Let $\{\hat{S}_1(R_1, L_1, rs_1), \hat{S}_2(R_2, L_2, rs_2), \dots, \hat{S}_n(R_n, L_n, rs_n)\}$ be the RC-NF2 schemas decomposed from SLN schema $\hat{S}(R, L, rs)$. A new semantic link type set $\{\alpha_1', \alpha_2', \dots, \alpha_i'\}$ and a new rule set $\{r_1', r_2', \dots, r_k'\}$ are appended to the schema \hat{S} . The extension of \hat{S} is denoted as $\hat{S}_E(R_E, L_E, rs_E)$.

1. Take all new semantic link types as isolated semantic link types firstly, and then construct new sub-schemas as $\hat{S}_1'(R_1', L_1', \phi)$, $\hat{S}_2'(R_2', L_2', \phi)$, \dots , $\hat{S}_i'(R_i', L_i', \phi)$. Let $\hat{S}_E = \{\hat{S}_1, \hat{S}_2, \dots, \hat{S}_n, \hat{S}_1', \hat{S}_2', \dots, \hat{S}_i'\}$.
2. For a new rule r_i' with the form of $\alpha_{i1} \cdot \alpha_{i2} \Rightarrow \alpha_j$ in $rs_E - rs$, we can get a direct semantic relative $\alpha_j \triangleleft \alpha_{i1}$ and $\alpha_j \triangleleft \alpha_{i2}$. And for a new rule r_i' with the form of $\alpha_i \Rightarrow \alpha_j$ in $rs_E - rs$, we can get a direct semantic relative $\alpha_j \triangleleft \alpha_i$.
3. For each semantic relative $\alpha_j \triangleleft \alpha_i$ retrieved in step 2, the list of sub-schemas in which α_i involved are $\hat{S}_{i1}, \hat{S}_{i2}, \dots, \hat{S}_{is}$, and the list of subschemas in which α_j involved in are $\hat{S}_{j1}, \hat{S}_{j2}, \dots$, and \hat{S}_{jt} .

- (a) Compute the up-closure $C_{up}(\alpha_i)$ for α_i in \hat{S}_{i1} , let $L_{jp} = L_{jp} \cup C_{up}(\alpha_i)$, for all $1 \leq p \leq t$;

- (b) Union all resource types involved in $C_{up}(\alpha_i)$ into R_{jp} , and union all rules involved in $C_{up}(\alpha_i)$ into rs_{jp} ; and,
- (c) Check if $\hat{S}_{jp1} \subseteq \hat{S}_{jp2}$, for each pair (p_1, p_2) , $1 \leq p_1 \leq s$, $1 \leq p_2 \leq t$. If yes, remove \hat{S}_{jp1} from the list of \hat{S}_E .

The following algorithm is for schema revision after deletion.

Algorithm 5. Let $\{\hat{S}_1(R_1, L_1, rs_1), \hat{S}_2(R_2, L_2, rs_2), \dots, \hat{S}_n(R_n, L_n, rs_n)\}$ be the RC-NF2 schemas decomposed from SLN schema $\hat{S}(R, L, rs)$. Denote the deleted resource types set as R_D , the semantic link types set as L_D , and the deleted rules set rs_D .

1. For each semantic sub-schema \hat{S}_i , let $R_i = R_i - R_D$, $L_i = L_i - L_D$, and $rs_i = rs_i - D$. Applying algorithm 1 or 3 on \hat{S}_i to get a decomposition for \hat{S}_i with corresponding semantic constraint normal form. Assume that the decomposition is $\hat{S}_{i1}(R_{i1}, L_{i1}, rs_{i1}), \hat{S}_{i2}(R_{i2}, L_{i2}, rs_{i2}), \dots, \hat{S}_{ik}(R_{ik}, L_{ik}, rs_{ik})$.
2. Union all sub-schemas retrieved in step 1, denote the new set of sub-schemas as $\{\hat{S}_1'(R_1', L_1', rs_1'), \hat{S}_2'(R_2', L_2', rs_2'), \dots, \hat{S}_m'(R_m', L_m', rs_m')\}$.
3. Remove the redundant sub-schemas. Check if $\hat{S}_j' \subseteq \hat{S}_i'$ for each pair (i, j) , $1 \leq i, j \leq m$. If yes, remove \hat{S}_j' .

B. REASONING ALGORITHMS

There are several kinds of reasoning over an SLN. The basic reasoning is to obtain the potential semantic relationships between resources. The following algorithm is for deriving potential semantic links.

Algorithm 6. Let R_1, R_2 and R_3 be resources of an SLN instance S of schema $\hat{S}(R, L, rs)$, the set of semantic links from R_1 to R_2 and from R_2 to R_3 be $A = \{\alpha_1, \alpha_2, \dots, \alpha_s\}$ and $B = \{\beta_1, \beta_2, \dots, \beta_t\}$ respectively, the set of known semantic links from R_1 to R_3 be C_0 , and R_i 's type be rt_i ($1 \leq i \leq 3$).

1. Let $C = \emptyset$.
2. For each pair (α_i, β_j) in $A \times B$, check rs to determine if there is a rule in form of $\alpha_i \cdot \beta_j \Rightarrow \gamma_k$. If yes, let $C = C \cup \gamma_k$; else skip to the next pair.
3. Let $C = C \cap [rt_1, rt_2]$.
4. Then, $C = C \cup C_0$ is the solution.

Because $A \subseteq [rt_1, rt_2]$, $B \subseteq [rt_2, rt_3]$ and the measure of $A \times B$ is a constant $s \times t$, and the number of rules in rs is a constant, the complexity of the above algorithm is $O(1)$. For a connected path $R_0 \xrightarrow{\alpha_1} R_1 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_n} R_n$ with length n , the algorithm to compute the semantic relationships between R_0 and R_n is to recursively use algorithm 6. And, the complexity is $O(n)$.

Moreover, we can compute the semantic relationships between any two resources R_1 and R_2 . The immediate idea is to find all connected paths from R_1 to R_2 , compute the semantic relationships of each connected path, and then combine all semantic relationships between R_1 and R_2 . For an SLN with m resources and n semantic links, under the

assumption of equal distribution, there are $d = \frac{n}{m(m-1)}$ links

between two resources on average. Therefore, from R_0 to R_n , the number of paths with length k is $P_{m-2}^{k-1} d^k$. Thus, the complexity for computing the semantic relationships between any two resources

is $O(\sum_{k=1}^{m-1} k P_{m-2}^{k-1} d^k) = O(\sum_{k=1}^{m-1} k \frac{(m-2)!}{(k-1)!} \frac{n^k}{m^k (m-1)^k})$. We can see that

the complexity depends on the two variables m and n , which could be large. However it can be reduced by reducing the scale of the SLN. The idea of decomposing an SLN into RC-NF2 forms can reduce the two variables prominently.

Therefore, we should find some better qualified method. Indeed, the up-closure might work to develop an efficient algorithm to determine semantic links between two resources. The following algorithm is for determining the existence of semantic relationship $R_1 \xrightarrow{\alpha} R_2$.

Algorithm 7. Let rs be the reasoning rule set of SLN S , R_1 's type be rt_1 , and R_2 's type be rt_2 .

1. Determine if $\alpha \in [rt_1, rt_2]$. If no, then return false; otherwise, continue the next step.
2. Compute the up closure of α w.r.t. rs , denoted as $C_{up}(\alpha)$.
3. Compute the projection of S w.r.t. $C_{up}(\alpha)$, denoted as S_0 , which might be constituted by several segments or some isolate resources, each segment is interconnected.
4. If R_1 and R_2 are in two different segments, then return false.
5. If R_1 and R_2 are in the same segment S_1 , then return determination if $R_1 \xrightarrow{\alpha} R_2$ is true in S_1 by using algorithm 6.

This algorithm is efficient because the up-closure for a semantic link type might be much smaller than the original link type set, and the projection might be sparse in light of semantic links. The following algorithm computes semantic link types set $\Gamma = \{\alpha | R_1 \xrightarrow{\alpha} R_2\}$ between resources R_1 and R_2 .

Algorithm 8. Let S be an SLN, R_1 's type be rt_1 , and R_2 's type be rt_2 , then any potential semantic link type between R_1 and R_2 is in $[rt_1, rt_2]$.

1. Let $\Gamma = \phi$.
2. For each semantic link type $\alpha \in [rt_1, rt_2]$, execute algorithm 7 to determine if $R_1 \xrightarrow{\alpha} R_2$. If yes, let $\Gamma = \Gamma \cup \{\alpha\}$; else skip to the next semantic link type.
3. Return Γ .

The following algorithm computes the resource set $A = \{R | R_1 \xrightarrow{\alpha} R\}$ for a given resources R_1 .

Algorithm 9. Let S be an SLN and R_1 's type be rt_1 .

1. Compute the up-closure of α w.r.t. rs , denoted as $C_{up}(\alpha)$;
2. Compute the projection of S w.r.t. $C_{up}(\alpha)$, denoted as S_0 ;
3. Denote $R' = \{rt | \alpha \in [rt_1, rt]\}$, $A = \{R | R \in S_0, \text{ and } R' \text{'s type is from } R'\}$;
4. For each resource $R \in A$, applying algorithm 7 to determine if $R_1 \xrightarrow{\alpha} R_2$. If no, let $A = A - \{R\}$; else skip to the next resource in A .
5. Return A .

The algorithm for computing the set of resource pair $\{(R_1, R_2) | R_1 \xrightarrow{\alpha} R_2\}$ for a given semantic link type α is similar to algorithm 7. However, the complexity would be much higher because both the start and end resources are uncertain.

V. CASE STUDY AND ANALYSIS

We firstly construct a schema of science network consisting of three components: resource types in Table I, semantic link types between these resource types in Table II, and reasoning rules for these semantic link types in Table III. Then, we construct the semantic link type relative net, which consists of six parts as shown in Fig.3. Therefore, we get the following six RC-NF1 sub-schemas.

Sub-schema \hat{S}_1 : $ResourceType = \{Researcher, Paper, Journal, Conference, Project, Institute, Publisher, Field\}$, $LinkType = \{colleague, Supervisor, Reader, Author, editor, employeeOf, Preside, takePartIn, Chair, 6\ subfield, involveIn, Sponsor, engageIn, sameField, publishedIn\}$, and $Rules$ includes the rules shown in Table III except rule 20.

Sub-schema \hat{S}_2 : $ResourceType = \{Journal, Conference, Publisher\}$, $LinkType = \{publishedBy, samePublisher\}$, and $Rules = \{rule\ 20\}$.

Sub-schema \hat{S}_3 : $ResourceType = \{Researcher\}$, $LinkType = \{friend\}$, and $Rules = \phi$.

Sub-schema \hat{S}_4 : $ResourceType = \{Researcher, Institute\}$, $LinkType = \{visit\}$, and $Rules = \phi$.

Sub-schema \hat{S}_5 : $ResourceType = \{paper\}$, $LinkType = \{sameAuthor\}$, and $Rules = \phi$.

Sub-schema \hat{S}_6 : $ResourceType = \{paper\}$, $LinkType = \{reference\}$, and $Rules = \phi$.

Sub-schemas 2-6 satisfy RC-NF2 because they are only with at most one rule. Sub-schema 1 is not RC-NF2, but it can be divided into RC-NF2 by the following approach.

We first find the following three bottom semantic link types in Fig.3: *engageIn*, *reader* and *employeeOf*, and then compute the following three up-closures of them.

$L_{1-1} = C_{up}(engageIn) = \{engageIn, subField, sponsor, sameField, publishedIn, involveIn, editor, chair, author, preside, takePartIn, supervisor\}$,

$L_{1-2} = C_{up}(reader) = \{reader, editor, publishedIn\}$, and

$L_{1-3} = C_{up}(employeeOf) = \{employeeOf, supervisor\}$,

colleague}.

The set of involved resource types and involved rules for L_{1-1} , L_{1-2} and L_{1-3} are listed as follows.

$R_{1-1} = \{Researcher, Paper, Journal, Conference, Project, Field\}$,

$R_{1-2} = \{Researcher, Paper, Journal, Conference\}$,

$R_{1-3} = \{Researcher, Institute\}$.

$rs_{1-1} = \{rule2, rule5, rule7, rule8, rule9, rule10, rule11, rule12, rule13, rule14, rule15, rule16, rule18, rule19\}$,

$rs_{1-2} = \{rule1, rule8\}$, and

$rs_{1-3} = \{rule3, rule4, rule6, rule17\}$.

TABLE I. RESOURCE TYPES FOR SCIENCE RESEARCH SLN SCHEMA

Resource Type	Attributes List
Researcher	name, sex, birthday, address, age
Paper	title, pubDate, length, author
Journal	name, vol, pubDate
Conference	name, vol, pubDate, location
Project	name, startDate, endDate, fund
Institute	name, location
Publisher	name, location, telephone
Field	name

TABLE II. SEMANTIC LINK TYPES FOR SCIENCE RESEARCH SLN SCHEMA

Resource Type Pairs	Semantic Link Type Set
[Researcher, Researcher]	{colleague, friend, supervisor}
[Researcher, Paper]	{reader, author, editor}
[Researcher, Journal]	{reader, author, editor}
[Researcher, Institute]	{employeeOf, visit}
[Researcher, Project]	{preside, takePartIn}
[Researcher, Conference]	{chair, author, reader, editor}
[Paper, Paper]	{reference, sameField, sameAuthor}
[Researcher, Field]	{engageIn}
[Conference, Conference]	{sameField, samePublisher}
[Conference, Field]	{involveIn}
[Journal, Journal]	{sameField, samePublisher}
[Journal, Conference]	{sameField, samePublisher}
[Journal, Field]	{involveIn}
[Field, Field]	{subfieldOf}
[Project, Conference]	{sponsor}
[Project, Paper]	{sponsor}
[Project, Field]	{involveIn}
[Journal, Publisher]	{publishedBy}
[Conference, Publisher]	{publishedBy}

Therefore, we get three RC-NF2 sub-schemas $\hat{S}_{1-1}(R_{1-1}, L_{1-1}, rs_{1-1})$, $\hat{S}_{1-2}(R_{1-2}, L_{1-2}, rs_{1-2})$, and $\hat{S}_{1-3}(R_{1-3}, L_{1-3}, rs_{1-3})$ from sub-schema 1. The corresponding semantic link type relative nets are shown in Fig. 4.

The number of semantic links in an SLN influences the reasoning complexity. Dividing a schema into several RC-NF2 sub-schemas can significantly reduce the scale so that the query and reasoning can be executed within a small scale. The soundness and the integrity of the query and reasoning on sub-schemas and on up closure are guaranteed by previous sections.

TABLE III. RULE SET FOR SCIENCE RESEARCH SLN SCHEMA

No.	Rules
rule 1	editor \Rightarrow reader
rule 2	preside \Rightarrow takePartIn
rule 3	colleague \bullet employeeOf \Rightarrow employeeOf
rule 4	supervisor \bullet employeeOf \Rightarrow employeeOf
rule 5	supervisor \bullet engageIn \Rightarrow engageIn
rule 6	employeeOf \bullet supervisor \Rightarrow employeeOf
rule 7	publishedIn \bullet involveIn \Rightarrow involveIn
rule 8	editor \bullet publishedIn \Rightarrow editor
rule 9	takePartIn \bullet involveIn \Rightarrow engageIn
rule 10	author \bullet involveIn \Rightarrow engageIn
rule 11	chair \bullet involveIn \Rightarrow engageIn
rule 12	editor \bullet involveIn \Rightarrow engageIn
rule 13	sponsor \bullet involveIn \Rightarrow involveIn
rule 14	involveIn \bullet subField \Rightarrow involveIn
rule 15	engageIn \bullet subField \Rightarrow engageIn
rule 16	sameField \bullet involveIn \Rightarrow involveIn
rule 17	colleague \bullet colleague \Rightarrow colleague
rule 18	sameField \bullet sameField \Rightarrow sameField
rule 19	subField \bullet subField \Rightarrow subField
rule 20	samePublisher \bullet publishedBy \Rightarrow publishedBy

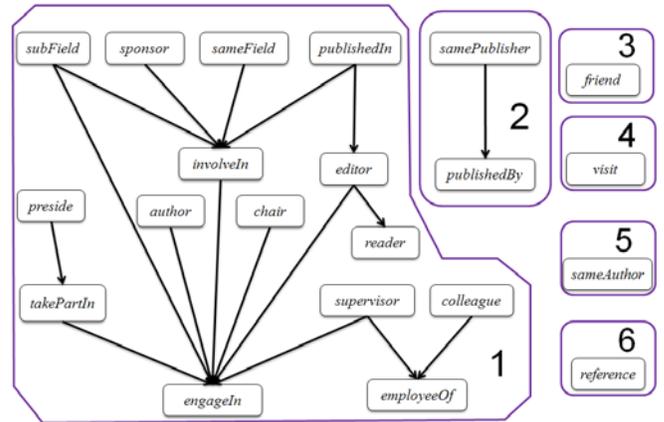


Fig.3. Semantic link types relative net for RC-NF1 schemas of a research SLN.

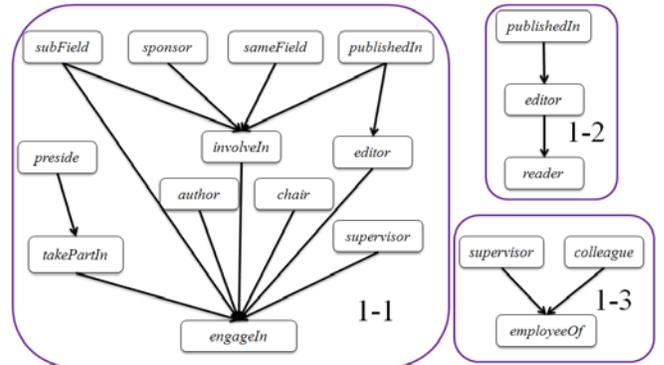


Fig.4. Semantic link types relative nets for RC-NF2 schemas of subschema \hat{S}_{1-1} , \hat{S}_{1-2} , and \hat{S}_{1-3} .

VI. COMPARISON

As a form of knowledge representation, traditional semantic network is a directed graph of concepts and semantic relations between concepts [23, 24]. SLN is different in the following aspects.

1. Nodes in semantic network are concepts, while nodes in SLN can be anything such as web pages, documents, software, images, concepts and even an SLN.
2. Edges in semantic network represent conceptual subsumption relationships, while links in SLN represent any semantic relations and even implicit relations.
3. Semantic network represents knowledge by organizing concepts, while the SLN is a semantic data model for managing resources on the Web.
4. Semantic network cannot support rule reasoning, while the SLN can support rule reasoning.

The schema of relational database is for defining the structure and metadata of relational tables. But it is hard to define rich semantics, and cannot support relational reasoning. Both XML schema and RDF schema mainly provide syntax for XML and RDF respectively, they lack reasoning ability. By comparison, SLN schema has stronger ability in representing various semantic relations between resources and supporting relational reasoning.

VII. CONCLUSIONS

SLN schema regulates the semantics of a SLN. The proposed approach provides a way to define SLN instances according to schema. The proposed rule-constraint normal forms can help manage and maintain SLN schema efficiently. Two algorithms for SLN extension and SLN reduction are introduced. Reasoning algorithms for retrieving new semantic relationships between resources or finding resources with certain semantic relationships have been proposed. The proposed SLN schema and relevant theory are important parts of the SLN as a Web semantic data model.

ACKNOWLEDGEMENTS

We thank all team members of the China Knowledge Grid Group for their help and cooperation (www.knowledgegrid.net). This work was supported by National Basic Research Program of China (2003CB317001) and International Cooperation Project of Ministry of Science and Technology of China (2006DFA11970).

REFERENCES

- [1] <http://www.ruleml.org/>
- [2] <http://www.w3.org/XML/Schema.html>
- [3] <http://www.w3.org/2004/OWL/>
- [4] *Extensible Markup Language (XML)*, available at <http://www.w3.org/XML>
- [5] *RDF Semantics*, available at <http://www.w3.org/TR/rdf-nt/#prelim>
- [6] F. Baader, D. Calvanese, et al, *The Description Logic Handbook*. Cambridge University Press, 2003.
- [7] F. Baader, I. Horrocks, U. Sattler, *Description Logics as Ontology Languages for the Semantic Web*. Mechanizing Mathematical Reasoning, 2005: 228-248

- [8] S. Bechhofer, F.V. Harmelen, et al, *OWL web ontology language reference*, available at <http://www.w3.org/TR/owl-ref/>, 2004.
- [9] T. Berners-Lee, J. Hendler, O. Lassila, *The Semantic Web*, Scientific American, vol.284, no.5, 2001, pp.34-43.
- [10] D. Brickley, R.V. Guha. *RDF vocabulary description language 1.0: RDF Schema*, <http://www.w3.org/tr/2002/wd-rdf-schema-20020430/>, 2002.
- [11] H. Boley, S. Tabet, G. Wagner, *Design Rationale of RuleML: A Markup Language for Semantic Web Rules*. International Semantic Web Working Symposium, 2001, Stanford, USA.
- [12] E.F. Codd, *Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks*, IBM Research Report, 1969
- [13] E.F. Codd, *A Relational Model of Data for Large Shared Data Banks*. (1970). Communications of the ACM 13 (6): 377-387.
- [14] A. Drozdek, *Structures and Algorithms in C++*, Second Edition, Course Technology, 2000.
- [15] J. Hefflin, J. Hendler. *A Portrait of the Semantic Web in Action*, vol.16, no.2, 2001, 54-59.
- [16] J. Hendler, *Agents and the Semantic Web*, IEEE Intelligent Systems, vol.16, no.2, 2001. 30-37.
- [17] I. Horrocks, P. F. Patel-Schneider, et al. *OWL Rules: A Proposal and Prototype Implementation*. Journal of Web Semantics, 2005,3(1):23-40.
- [18] I. Horrocks, P.F. Patel-Schneider. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, available at <http://www.w3.org/Submission/SWRL/>.
- [19] M. Klein. *XML, RDF, and Relatives*, IEEE Internet Computing, (2001)Mar./Apr, 26-28.
- [20] O. Lassila, R.R. Swick. *Resource description framework (RDF) model and syntax specification*. recommendation, W3C, 1999. <http://www.w3.org/tr/1999/rec-rdf-syntax-19990222>, 1999.
- [21] A. Maedche and S. Staab, *An Ontology Learning for the Semantic Web*, IEEE Intelligent Systems, (2001). Mar./Apr, 72-79.
- [22] D.L. McGuinness, F.V. Harmelen. *OWL Web Ontology Language Overview*, available at <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
- [23] M. R. Quillian. *Semantic Memory*. Semantic Information Processing, M.Minsky (Ed.), MIT press, Cambridge, Mass, 1968.
- [24] J.F. Sowa. *Semantic Networks*. <http://www.jfsowa.com/pubs/semnet.htm>, August, 2002.
- [25] H. Zhuge, *Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning*, IEEE Transactions on Knowledge and Data Engineering, <http://doi.ieeecomputersociety.org/10.1109/TKDE.2008.141>.
- [26] H. Zhuge, *Active Document Framework ADF: Model and Tool*, Information and Management, vol.41, no.1, pp.87-97, 2003.
- [27] H. Zhuge, Y. Sun, et al, *Algebra Model and Experiment for Semantic Link Network*, International Journal of High Performance Computing and Networking, vol.3, no.4, pp.227-238, 2005.
- [28] H. Zhuge, *The Knowledge Grid*, World Scientific, 2004.
- [29] H. Zhuge, *Autonomous semantic link networking model for the Knowledge Grid*, Concurrency and Computation: Practice and Experience, vol.7, no.19, (2007), pp.1065-1085.
- [30] H. Zhuge, et al, *Modeling Language and Tools for the Semantic Link Network*, Concurrency and Computation: Practice and Experience, vol.20, no.7, (2008), pp.885-902.
- [31] H. Zhuge and X. Li, *Peer-to-Peer in Metric Space and Semantic Space*, IEEE Transactions on Knowledge and Data Engineering, vol.6, no.19, (2007), pp.759-771.
- [32] H. Zhuge, Y. Sun and W. Guo, *Theory and Algorithm for Rule Base Refinement*. 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2003, Loughborough, UK, June 23-26, 2003, LNCS 2718 Springer 2003, pp. 187-196.

*Technical Report of Knowledge Grid Research Center, KGRC-2008-01, 2008. www.knowledgegrid.net/TR.