Algebra model and experiment for semantic link network

Hai Zhuge*
Key Lab of Intelligent Information Processing,
Institute of Computing Technology,
Chinese Academy of Sciences, P.O. Box 2704, Beijing, China
Fax: +861062562703 E-mail: zhuge@ict.ac.cn
*Corresponding author

Yunchuan Sun
Institute of Computing Technology,
Graduate School of the Chinese Academy of Sciences,
P.O. Box 2704, Beijing, China
School of Economic and Business Administration,
Beijing Normal University, Beijing, China
E-mail: yunchsun@163.com

Ruixiang Jia and Jie Liu
Institute of Computing Technology,
Graduate School of the Chinese Academy of Sciences,
P.O. Box 2704, Beijing, China
E-mail: jiaruixiang@263.net E-mail: lj@kg.ict.ac.cn

Abstract: Semantic Link Network SLN is a semantic extension of hyperlink network. Based on the previous SLN model, this paper enriches the semantic links and reasoning rules, proposes an algebra model that supports semantic reasoning and the management of SLN by using a semantic matrix and presents the experiment for putting an SLN into practice. A solution to efficiently manage a huge SLN by dividing the whole SLN into partitioned semantic matrices is also put forth. The consistency issue about SLN is discussed, based on the matrix representation. Experiment of establishing and using the SLN shows the feasibility of the proposed model.

Keywords: knowledge grid; matrix; semantic grid; semantic link; semantic web.

Reference to this paper should be made as follows: Zhuge, H., Sun, Y., Jia, R. and Liu, J. (2005) 'Algebra model and experiment for semantic link network', *Int. J. High Performance Computing and Networking*, Vol. 3, No. 4, pp.227–238.

Biographical notes: Hai Zhuge is a Professor and the Director of the Key Lab of Intelligent Information Processing, at the Institute of Computing Technology in Chinese Academy of Sciences. He is the Founder of the China Knowledge Grid Research Group (http://kg.ict.ac.cn), which has 30 young researchers. Currently, he is the Chief Scientist of the China National Semantic Grid Project, a five-year project of the National Basic Research Program of China. He was the keynote speaker at a number of international conferences. He has organised several international journal special issues on *Semantic Grid and Knowledge Grid*. He is serving as an Area Editor, an Associate Editor and is on the editorial boards of many journals. His current research interest is the model and theory on future interconnection environment and applications in China. His monograph *The Knowledge Grid* is the first book in the area. He is the author of over 100 papers appeared mainly in leading international conferences and journals. He is among the top scholars in the systems and software engineering field during 1999–2003, according to the assessment report published in *Journal of Systems and Software* in 2004. Zhuge is a Senior Member of IEEE and a member of ACM.

Yunchuan Sun is an engineer at the School of Economic and Business Administration, Beijing Normal University, Beijing, China. He received his master's degree of computer science from the School of Information and Science, Beijing Normal University in 2003. He is also a PhD student of Institute of Computing Technology, Chinese Academy of Science.

Ruixiang Jia is a Research Assistant and graduate student of China Knowledge Grid Research Group. He has published two papers in international journals.

Jie Liu is a Research Fellow. Her research interest includes: P2P computing, semantic-based service integration and heterogeneous data integration. She has published six papers in international journals and received the Microsoft Fellow Award in 2003. She is a member of the IEEE Computer Society.

1 INTRODUCTION

The World Wide Web has become an important means to distribute and retrieve information for people around the world. However, the hyperlink-based network do not reflect machine-readable semantics, so it can hardly support accurate information retrieval and intelligent services because search engines and applications cannot understand the content of web pages and the semantic relationships among pages through simple hyperlinks on the current web (Allan, 1997; Henzinger, 2001; Tudhope and Taylor, 1997).

Tim Berners-Lee proposed the concept of semantic web to improve the current web by means of giving web pages well-defined meaning (Berners-Lee et al., 2001; Hendler, 2001, 2003). It is mainly based on ontology mechanisms and mark-up languages such as XML and RDF (Bray et al., 1997; Decker et al., 2000; Klein, 2001; Lassila and Swick, 1999b). RDF is an approach for processing metadata. It provides inter-operability between applications that exchange machine-understandable information on the web (Lassila and Swick, 1999a). An approach for knowledge representation by extending the RDF schema is presented in Broekstra et al. (2001). The semantic grid, a natural development of semantic web and grid, intends to incorporate the advantages of both (Foster et al., 2002; Hendler, 2001). OIL (Ontology Inference Layer), a major spin-off from the IST project On-To-Knowledge (Fensel et al., 2000), is a web-based representation and inference layer for ontology mechanisms, which unifies three important aspects provided by different communities:

- formal semantics and efficient reasoning support, as provided by Description Logics
- epistemological rich modelling primitives, as provided by the Frame community
- a standard proposal for syntactical exchange notations, as provided by the web community (Fensel et al., 2000).

DAML Query Language (DQL) and RDF Query Language (RQL) are two kinds of query interface for semantic web. DQL is a formal language and protocol for a querying agent (which is referred to a client) and an answering agent (which is referred to as a server) to use in conducting a query-answering dialogue using knowledge represented in DAML+OIL (Fikes et al., 2002). RQL is a typed functional language and relies on a formal model for directed labelled graphs, permitting the interpretation of superimposed resource descriptions by means of one or more RDF schemas. RQL adapts the functionality of semi-structured/XML query languages to the peculiarities of RDF, but it enables uniform query of both resource descriptions and schemas (Karvounarakis et al., 2002).

The semantic web has a great potential. However, there exist many difficulties in its full implementation. *Easy-to-use* and *easy-to-build* are two important criteria that determine the success or failure of a new technique. So we have proposed the Semantic Link Network (SLN) as a model to facilitate the semantic web (Zhuge, 2003). The semantic link is the natural extension of the current hyperlink (Zhuge, 2003). It provides the primitives to represent semantic relationships among resources (concepts, documents, images, etc).

In this paper, an algebra model for SLN is developed, based on our previous SLN model. The semantic links and reasoning rules are first enriched and then a computing model of SLN including semantic link reasoning and operations is proposed. Based on a semantic matrix representation of SLN, a matrix-based reasoning theory and management approach is developed.

2 SEMANTIC LINK REASONING

2.1 Enriched semantic links

Our previous SLN model has seven semantic link primitives: cause-effective link (*ce*), implication link (*imp*), subtype link (*st*), similar-to link (*sim*), instance link (*ins*), sequential link (*seq*) and reference link (*ref*) (Zhuge, 2003, 2004b). A semantic link with semantic factor (semantic relationship) α between two resources r_1 and r_2 is denoted as $r_1 \xrightarrow{\alpha} r_2$. The following semantic links and operations are added.

- *Equal-to link*, denoted as *e*, indicates that two resources are identical in semantics. Obviously, a resource is equal to itself. Equality relationship can be regarded as a special case of the similar relationship. So all rules of the *similar-to* link also holds for the *equal-to* link by replacing the *similar-to* link with the *equal-to* link. The *equal-to* link is useful in SLN reasoning processes.

- *Empty link*, denoted as ϕ, represents that two resources are absolutely irrelevant in semantics.
- *Null link or unknown link*, denoted as *Null* or *N*, indicates that the semantic relationship between two resources is uncertain or unknown. *Null* relationship means that the semantic relationship between two resources is not known, although there may exist some semantic relationship. *Null* relationship can be replaced with some other relationship, once it is changed by users or derived by reasoning mechanism.
- *Non-α relationship*, denoted as *Non*(α) *or* α^N for some semantic relationship α, which means that there does not exist the α relationship between two resources. Sometimes, it is useful in reasoning process if we know that there is no certain semantic relationship between two resources.
- *Opposite relationship*, denoted as $r_1 \xrightarrow{op} r_2$, which states that the successor *declares* the opposite idea of the predecessor. If $r_1 \xrightarrow{op} r_2$ and $r_2 \xrightarrow{op} r_3$ hold, we can get that $r_1 \xrightarrow{sim} r_3$. The opposite relationship is symmetrical, i.e., $r_1 \xrightarrow{op} r_2$ is equal to $r_2 \xrightarrow{op} r_1$.

Although the semantic relationship *e* and *Null* may not be marked on the SLN, they are surely useful and important to the semantic reasoning over an SLN.

2.2 Reasoning rules

Reasoning among semantic relationship is to get some unknown or unmarked semantic relationship between two resources by reasoning rules. For example, suppose that $d_1 \xrightarrow{ce} d_2$ and $d_2 \xrightarrow{ce} d_3$ are two semantic relationships between documents, we can get the third semantic relationship between documents d_1 and d_3: $d_1 \xrightarrow{ce} d_3$ according to the transitive characteristic of the cause-effective link.

Twenty-two reasoning rules about seven types of semantic links have been given in Zhuge (2003). More domain-dependent reasoning rules can be developed according to application requirement. Based on new semantic links and relationships discussed above, new domain-independent reasoning rules can be obtained as shown in Table 1.

Table 1 *Reasoning rules*

No.	Rules
1	$r \xrightarrow{e} r' \Rightarrow r' \xrightarrow{e} r$
2	$r \xrightarrow{e} r', r' \xrightarrow{\alpha} r'' \Rightarrow r \xrightarrow{\alpha} r''$
	$r \xrightarrow{\alpha} r', r' \xrightarrow{e} r'' \Rightarrow r \xrightarrow{\alpha} r''$
3	$r \xrightarrow{\alpha} r', r' \xrightarrow{N} r'' \Rightarrow r \xrightarrow{N} r''$
	$r \xrightarrow{N} r', r' \xrightarrow{\alpha} r'' \Rightarrow r \xrightarrow{N} r''$
4	$r \xrightarrow{\alpha} r', r' \xrightarrow{\phi} r'' \Rightarrow r \xrightarrow{N} r''$
	$r \xrightarrow{\phi} r', r' \xrightarrow{\alpha} r'' \Rightarrow r \xrightarrow{N} r''$
5	$r \xrightarrow{op} r', r' \xrightarrow{op} r'' \Rightarrow r \xrightarrow{sim} r''$
6	$r \xrightarrow{op} r' \Rightarrow r' \xrightarrow{op} r$

A semantic factor α is stronger than another α', denoted as $\alpha' \leq \alpha$ or $\alpha \geq \alpha'$, if α' is implied by α in semantics. That is to say, if there exists a semantic factor α between two resources, then there must exist a semantic factor α'. It is obvious that the implication relationship is reflexive, asymmetric and transitive. For two semantic factors α and β, if $\alpha \geq \beta$ and $\beta \geq \alpha$, then $\alpha = \beta$.

Specially, it is stipulated that each of the seven semantic factors (*ce, imp, sim, ins, ref, st* and *seq*) is weaker than the *equal-to* semantic factor, which means that a resource can be viewed as a cause, an implication, an instance, a subtype, a similarity, a reference or a sequence of itself. Therefore we have $ce \leq e$, $imp \leq e$, etc.

By using reasoning rules, the *Null* semantic links among resources over a semantic network can be updated to some certain semantic links that are derived by logical reasoning. In the following discussion, certain semantic links (which is marked clearly over the network) and *Null* semantic factors (which are unmarked over the network) are all called *semantic factors*.

Figure 1 is an example of semantic link network. For the semantic links $r_1 \xrightarrow{ce} r_3$ and $r_3 \xrightarrow{ins} r_4$ shown in Figure 1, the semantic link $r_1 \xrightarrow{ce} r_4$ can be deduced according to the reasoning rule *ce* • *ins* → *ce* (Zhuge, 2003).

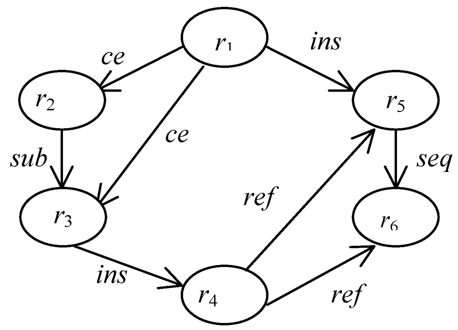


Figure 1 *Example of SLN*

3 OPERATIONS ON SEMANTIC FACTOR

We herein provide three operations: reversion, addition and multiplication for representing transformation and composition of semantic links. These operations take one or two semantic relationships as input and produce a new semantic relationship as output. Reversion is a unary operation for its unique input, while addition and multiplication are two binary operations, for they need two semantic relationships as their input.

Definition 1 (Reversion)**:** If there is a semantic relation α from r_1 to r_2, then there is a reverse semantic relationship from r_2 to r_1, we call it reversion relationship, denoted as *Reverse* (α) or α^R.

For example, a cause-effective link from r_1 to r_2 means that r_1 is the cause of r_2 and r_2 is the effect of r_1, i.e., a *Reverse*(*ce*) relation from r_2 to r_1. A semantic relation and its reverse declare the same thing, but the reverse relationship is useful in reasoning. Obviously we have the following operational laws:

- $e^R = e$
- $N^R = N$
- $\phi^R = \phi$
- $sim^R = sim$ (similarity degree is the same)
- $(\alpha^R)^R = \alpha$.

Definition 2 (Addition operation)**:** If there exist two semantic links with semantic factors α and β from r_1 to r_2 over an SLN, then the two semantic links can be merged into one with the semantic factor $\alpha + \beta$. The addition operation is depicted in Figure 2. Such semantic link merge operation is determined by the addition operation of α and β.

For example, if there exist two semantic links *ce* and *seq* from r_1 to r_2 over an SLN, then the meaning of the semantic link from r_1 to r_2 is *ce* + *seq*, which means that r_1 is not only the cause but also the predecessor of r_2. The addition operation can be extended from two semantic links to *n* semantic links α_1, α_2, …, α_n and the result can be denoted as $\alpha_1 + \alpha_2 + \ldots + \alpha_n$ or $\sum \alpha_t (1 \le i \le n)$.

Conflict occurs when there exist two semantic factors α and α^N between the same pair of resources. That means that a semantic conflict occurs in SLN when $\alpha + \alpha^N$ occurs during reasoning. For example, each of the following addition expressions leads to the semantic conflict: $\alpha + \phi$ ($\alpha \neq \phi$), $op + e$, $op + ce$, $op + st$, $op + sim$ and $op + imp$. Once a conflict occurs, the most important thing is to deal with the conflict by modifying the involved semantic links or resources. Only a consistent SLN can properly support problem-solving and question-answering. So in the following discussions, if not specified, operations and reasoning are carried out only in a consistent SLN.

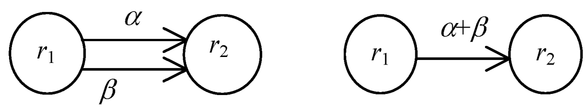


Figure 2 *Addition operation of semantic relations*

According to the definition of addition operation, we have the following operation laws and characteristics.

Laws for addition operation:

- $\alpha + \alpha = \alpha$ (Idempotency).
- $\alpha + \beta = \beta + \alpha$ (Commutativity)
- $(\alpha + \beta) + \gamma = \alpha + (\beta + \gamma)$ (Associative addition).
- $\alpha + Null = \alpha$, $Null + \alpha = \alpha$.
- If $\alpha' \le \alpha$, then $\alpha + \alpha' = \alpha$. Specially, $e + \alpha = e$, where α is a semantic factor that is compatible with *e*. For example, if α is any one of the prior seven semantic factors, $e + \alpha = e$ holds according to the stipulation in Section 2.2.
- $(\alpha + \beta)^R = \alpha^R + \beta^R$.

Characteristic 1: For any two semantic factors α and β involved in a consistent SLN, we have $\alpha \le \alpha + \beta$ and $\beta \le \alpha + \beta$.

Characteristic 2: For any three semantic factors α, β and γ involved in a consistent SLN, if $\alpha \ge \beta$ and $\alpha \ge \gamma$, then $\alpha \ge \beta + \gamma$ holds.

Definition 3 (Multiplication operation)**:** Assume there exist two semantic relations: α is from r_1 to r_2 and β is from r_2 to r_3 over a consistent SLN. If we can get the semantic factors γ_1, γ_2, …, and γ_k from r_1 to r_3 by reasoning based on prior assumption, then we call the reasoning process multiplication operation, denoted as $\alpha \times \beta = \gamma$ where $\gamma = \gamma_1 + \gamma_2 + \ldots + \gamma_k$.

Figure 3 shows an example of the multiplication process. Assume that there exist reasoning rules $\alpha \times \beta \to \gamma_1$, $\alpha \times \beta \to \gamma_2$, …, $\alpha \times \beta \to \gamma_k$, and $\gamma = \gamma_1 + \gamma_2 + \ldots + \gamma_k$, then the following reasoning rule $\alpha \times \beta \to \gamma$ holds. And if the result of $\alpha \times \beta$ is ambiguous, we denote the multiplication of α and β is *Null*, i.e., $\alpha \times \beta = Null$.

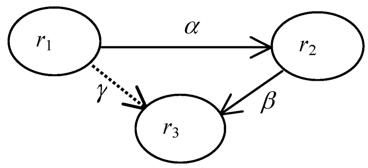


Figure 3 *Example of multiplication operation: $\alpha \times \beta = \gamma$*

In fact, the process of multiplication of *n* semantic relations, denoted as $\alpha_1 \times \alpha_2 \times \ldots \times \alpha_n$, is the process of a logical reasoning. Assume that there are *n* + 1 resources, denoted as r_0, r_1, …, r_n and *n* semantic relations $r_i \xrightarrow{\alpha_{i+1}} r_{i+1}$, $0 \le i \le n-1$, the objective of the multiplication of these *n* semantic relations is to get the semantic relation from r_0 to r_n. Based on the above two definitions, we can get the following multiplication operation laws.

Multiplication operation laws:

- $\alpha \times e = \alpha$, $e \times \alpha = \alpha$.
- $\alpha \times N = N$, $N \times \alpha = N$.
- $\alpha \times \phi = N$, $\phi \times \alpha = N$. Specially, $\phi \times \phi = N$, this means that the multiplication of $\phi \times \phi$ is not fixed on ϕ.
- $(\alpha + \beta) \times \gamma = \alpha \times \gamma + \beta \times \gamma$, $\alpha \times (\beta + \gamma) = \alpha \times \beta + \alpha \times \gamma$.
- $(\alpha \times \beta)^R = \beta^R \times \alpha^R$.

Lemma 1: *For any semantic relations $r_1 \xrightarrow{\alpha} r_2$, $r_1 \xrightarrow{\beta} r_2$, and $r_2 \xrightarrow{\gamma} r_3$ in a consistent SLN, if $\alpha \ge \beta$, the relation between two semantic meanings from r_1 to r_3 is $\alpha \times \gamma \ge \beta \times \gamma$.*

Proof: For any semantic relations α, β and γ, if $\alpha \ge \beta$ holds in a consistent SLN, we have $\alpha + \beta = \alpha$ according to addition laws and $\alpha \times \gamma = (\alpha + \beta) \times \gamma = \alpha \times \gamma + \beta \times \gamma \ge \beta \times \gamma$ according to the multiplication laws, so the lemma holds. □

In many cases, the reasoning rules are commutative, i.e., $\alpha \times \beta = \beta \times \alpha$. For example, $ce \times st = st \times ce$ and $imp \times st = st \times imp$ hold. However, we cannot assure the commutative characteristic holds for any two semantic relations.

Another issue is that whether the multiplication combination law $\alpha \times (\beta \times \gamma) = (\alpha \times \beta) \times \gamma$ holds or not. Take Figure 4(a) for example, there are two ways to compute the semantic relation from r_1 to r_4:

- compute the semantic relation from r_1 to r_3 firstly as showed in Figure 4(b) and the result is $(\alpha \times \beta) \times \gamma$
- compute the semantic relation from r_2 to r_4 firstly as showed in Figure 4(c) and the result is $\alpha \times (\beta \times \gamma)$.

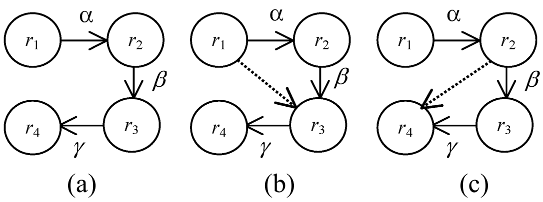


Figure 4 *Orders of the multiplication*

Obviously, the two results are both reasonable. However, we cannot assure whether these results are equivalent. A way to solve this problem is to take the summation of the two results as the final result. It is feasible to get the precise result considering all reasonable results. So the semantic relation from r_1 to r_4 of this example is $\alpha \times (\beta \times \gamma) + (\alpha \times \beta) \times \gamma$.

We can easily verify that the multiplication will be *Null* if one of α, β and γ is *Null*, which means $Null \times \beta \times \gamma = Null$, $\alpha \times Null \times \gamma = Null$ and $\alpha \times \beta \times Null = Null$. And the result can be extended to the multiplication for n semantic relationships and the following corollary describes this character.

Corollary 1: For n semantic relationships α_1, α_2, …, and α_n, the multiplication $\alpha_1 \times \alpha_2 \times \ldots \times \alpha_n$ will be *Null* if there is some α_i such that $\alpha_i = Null$.

The proof for the corollary is easy by using induction on the basic condition: $\alpha \times N = N$ and $N \times \alpha = N$.

4 MATRIX REPRESENTATION FOR SLN AND MATRIX-BASED REASONING

4.1 Concept definition

Similar to any other networks, an SLN can be represented as a directed graph with semantic relations. So an SLN can be expressed as $S(N, L)$, where N is a set of resource nodes and L is a set of directed semantic links.

As discussed above, a semantic link can be appended to an SLN once the exact meaning between two resources can be derived through logical reasoning. So we can get a closure for the SLN by adding semantic links over an SLN.

Definition 4 (Closure of SLN)**:** The closure of an SLN $S(N, L)$ is a new SLN (N, L'), denoted as *Closure*(S) or S^+, where L' is constructed as follows:

- all semantic links included in L are included in L'
- a semantic link from a resource to another is appended to L' if the semantic relation between the two resources is available via reasoning on L.

The closure is unique for a given SLN. We say two SLNs are equivalent if their closures are the same, i.e., two SLNs are equivalent if and only if their closures are identical. An SLN is equivalent to its closure. And the equivalence between SLNs is reflexive, symmetric and transitive. An SLN $S(N,L)$ is said to be included by another one $T(N',L')$ if $N \subseteq N'$ and $L \subseteq L'$, denoted as $S \subseteq T$. Obviously $S \subseteq S^+$.

Lemma 2: *For two SLNs S and T, S is equivalent to T if and only if* $T \subseteq S^+$ *and* $S \subseteq T^+$.

Proof: The proof consists of the following two steps:

(1) If S is equivalent to T, then $S^+ = T^+$, $S \subseteq S^+$, $T \subseteq T^+$.
(2) For any semantic link sl in T^+, we can get that sl can be retrieved from T, $T \subseteq S^+$, sl can be retrieved from S^+. That is to say, sl belongs to $Closure(S^+) = S^+$. It means that sl is in S^+, obviously $T^+ \subseteq S^+$. Also we have $S^+ \subseteq T^+$.

According to equations (1) and (2) we have $S^+ = T^+$. □

A semantic link can be removed if the attached semantic relations can be deduced by other semantic links. And these links are called redundant semantic links in view of reasoning. A minimal cover can be obtained by removing all redundant semantic links from the original SLN.

Definition 5 (Minimal Cover for SLN)**:** An SLN M is the minimal cover of another S, if M and S satisfy the following conditions.

- $M^+ = S^+$
- no semantic link sl exists in M such that $(M - sl)^+ = M^+$ holds.

A minimal cover of an SLN is a refined SLN, which involves the least number of semantic links and keeps equivalent to the original. We can use the approach for refining a rule base to get the minimal cover that is important to maintain an SLN (Zhuge et al., 2003).

4.2 Matrix representation of SLN

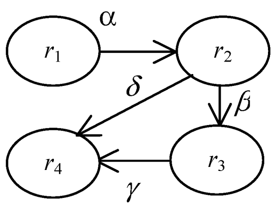
An SLN can be represented by an adjacent matrix. Given an SLN with n resources r_1, r_2, … , and r_n, it can be represented by matrix as follows, where l_{ij} represents the semantic factor from r_i to r_j. We call it semantic relationship matrix (*SRM*).

$$SRM = \begin{pmatrix} l_{11} & l_{12} & \dots & l_{1n} \\ l_{21} & l_{22} & \dots & l_{2n} \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix}. \quad (1)$$

As discussed in Section 2, the semantic relation from a resource to itself can be regarded as e, so for any i, $l_{ii} = e$. And for any i and j, $l_{ij} = l_{ji}^R$ in an SLN matrix. So the SRM defined in equation (1) takes the following form. Clearly, the SRM for an SLN is Reversion symmetrical, denoted R-Symmetrical.

$$SRM = \begin{pmatrix} e & l_{12} & \dots & l_{1n} \\ l_{12}^R & e & \dots & l_{2n} \\ \dots & \dots & \dots & \dots \\ l_{n1}^R & l_{n2}^R & \dots & e \end{pmatrix}. \quad (2)$$

If there does not exist marked semantic links between two resources r_i and r_j, then we have $l_{ij} = Null$ and $l_{ji} = Null$. For a given SLN, the corresponding matrix defined by equations (1) or (2) is unique and vice versa. For example, the right part of Figure 5 is the semantic matrix of the SLN shown on the left.



$$SRM = \begin{bmatrix} e & \alpha & N & N \\ \alpha^R & e & \beta & \delta \\ N & \beta^R & e & \gamma \\ N & \delta^R & \gamma^R & e \end{bmatrix}$$

Figure 5 *A simple SLN and its semantic relationship matrix*

4.3 Reasoning with SLN matrix

Reasoning in an SLN is to derive the semantic relation between two resources by logical reasoning via a series of semantic relations (links). Assume a consistent SLN consists of n resources: r_1, r_2, … , and r_n, and its semantic relationship matrix is M. Can we derive the reliable semantic relations of any two resources from a semantic relationship matrix? Obviously, we can get l_{ij} as the semantic relation between r_i and r_j if it is marked in the matrix. However, sometimes the matrix tells us *Null* as the semantic meaning between r_i and r_j although it may be any other semantic relations and can be retrieved by reasoning. So the reliable semantic relation, denoted as $l_{ij}^{\#}$, should be derived by reasoning.

Theorem 1: *In a consistent SLN, a reliable semantic relation can be computed by using the following formula:*

$$l_{ij}^{\#} = M_{i*} \times M^{n-2} M_{*j},$$

where M is the semantic relationship matrix, M_{i*} is the ith row vector and M_{*j} is the jth column vector of M, that is:

$$M_{i*} = \left[l_{i1}, l_{i2} \dots l_{in} \right] \text{ and } M_{*j} = \begin{bmatrix} l_{1j} \\ l_{2j} \\ \dots \\ l_{nj} \end{bmatrix}.$$

Proof: In fact, the solution for getting the reliable semantics from r_i to r_j is as follows:

1 find all possible paths from r_i to r_j in the SLN
2 reasoning along each possible path obtained in (1)
3 take the summation of all reasoning results as the final result.

Any two resources in the SLN are connected because all unknown semantic meanings between them are regarded as *Null* semantic meaning. Therefore, all paths from r_i to r_j can be classified as follows according to their lengths.

- length = 1: $r_i \rightarrow r_j$, the semantic meaning is l_{ij}
- length = 2: $r_i \rightarrow r_k \rightarrow r_j$, $(1 \le k \le n)$, the semantic meaning summation is $\sum l_{ik} l_{kj}$, $(1 \le k \le n)$
- length = 3: $r_i \rightarrow r_k \rightarrow r_m \rightarrow r_j$, $(1 \le k,m \le n)$, the semantic meaning summation is $\sum\sum(l_{ik} \times l_{km} \times l_{mj})$, $(1 \le k, m \le n)$

… … …

- length = $n - 1$: $r_i \rightarrow r_{k1} \rightarrow r_{k2} \rightarrow \dots \rightarrow r_{k(n-2)} \rightarrow r_j$, $(1 \le kp \le n$, here p is a positive integer and $1 \le p \le n-2)$, the semantic meaning summation is $\sum\sum \cdots \sum(l_{i,k1} \times l_{k1,k2} \times \dots \times l_{k(n-3),k(n-2)} \times l_{k(n-2),j})$, $(1 \le kp \le n, 1 \le p \le n-2)$

… … …

In the following, we prove that the summation of the semantic meaning when length = $n - 1$ implies all others. We need to check the following two cases:

Case 1: Length < $n - 1$. We take length = 2 as an example, assuming that $k_s = k$, while $k_1 = k_2 = \dots = k_{s-1} = i$ and $k_{s+1} = \dots = k_{n-2} = j$, the summation is like $\sum(e \times e \times \dots e \times l_{ik} \times l_{kj} \times e \times \dots \times e) = \sum(l_{ik} \times l_{kj})$. That means length = 2 can be viewed as a special case of length = $n - 1$. Similarly, it is true for all length < $n - 1$.

Case 2: Length > $n - 1$. There must be at least one ring in the path since there are only n resources. Each of these paths can be denoted as $r_i \rightarrow r_{k1} \rightarrow r_{k2} \rightarrow \cdots \rightarrow r_{ks} \rightarrow r_m \rightarrow r_{m1} \rightarrow \cdots \rightarrow r_{mt} \rightarrow r_m \rightarrow r_{l1} \rightarrow \cdots \rightarrow r_{lp} \rightarrow r_j$, the semantic meaning along such a path is $l_{i,k1} \times l_{k1,k2} \times \dots \times l_{ks,m} \times l_{m,m1} \times \dots l_{mt,m} \times l_{m,l1} \times \dots \times l_{lp,j}$, as shown in Figure 6. We can easily obtain that $l_{i,k1} \times l_{k1,k2} \times \dots \times l_{ks,m} \times l_{m,m1} \times \dots l_{mt,m} \times l_{m,l1} \times \dots \times l_{lp,j} \le l_{i,k1} \times l_{k1,k2} \times \dots \times l_{ks,m} \times l_{m,l1} \times \dots \times l_{lp,j}$ because the reasoning process from the ring $r_m \rightarrow r_{m1} \rightarrow \cdots \rightarrow r_{mt} \rightarrow r_m$ is actually to compute the semantic meaning from r_m to itself and the result is surely weaker than e. The right part of the inequality is exactly the semantic meaning by reasoning along the path $r_i \rightarrow r_{k1} \rightarrow r_{k2} \rightarrow \cdots \rightarrow r_{ks} \rightarrow r_m \rightarrow r_{l1} \rightarrow \cdots \rightarrow r_{lp} \rightarrow r_j$, if the length of this path is longer than n, there must be another ring in the path and we can deal with it as above till the length is smaller than n. That means the semantics of any semantic path with length $> n - 1$ is implied by that of a path with length $< n$.

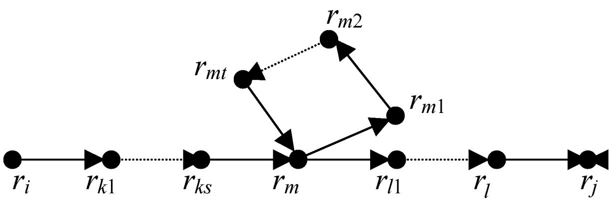


Figure 6 *Case length >n – 1*

According to Case 1 and Case 2, the summation of all reasoning results is the summation of the semantic meaning when length = $n - 1$, i.e.,

$$l_{ij}^{\#} = \Sigma\,\Sigma \ldots \Sigma\left(l_{i,k1} \times l_{k1,k2} \times \ldots \times l_{k(n-3),k(n-2)} \times l_{k(n-2),j}\right), \quad \left(1 \le k_p \le n, 1 \le p \le n-2\right).$$

And the result is exactly the result of $M_{i*} \times M^{n-2} \times M_{*j}$. □

Corollary 2: *Let* $l_{ij}^{\#}$ *and* l_{ij} *be the same meaning as defined above, we have* $l_{ij}^{\#} \ge l_{ij}$.

We can easily prove Corollary 2 according to the proving process of Theorem 1.

If we compute the semantic relationships of any two resources in a consistent SLN by the above formula, then we can get a new semantic relationship matrix, called a full semantic relationship matrix (FSRM) denoted as M_f where M is the original semantic matrix. In fact an FSRM is the SLN matrix of the closure of the original SLN. We get the reliable semantic relationship between any two resources in the SLN through the FSRM. Of course, some of these semantic relationships are marked clearly and attached with the semantic factors in the original SLN and the other can be derived by logical reasoning. The fact is that any logical reasoning using the semantic relationships can be realised by the multiplication of the SLN matrix by itself.

The FSRM is surely of concern for an SLN, the causes are as follows:

- it is an efficient tool to the SLN logical reasoning for we can get the semantic relationship between any two resources by searching it in the FSRM
- FSRM is useful to detect the inconsistency and maintain the consistency for an SLN.

The primary method is to find whether there exists a conflict semantic relationship in the FSRM or not. Further, the FSRM can be used to manage an SLN.

Corollary 3: *For a semantic relationship matrix M and its FSRM* M_f, *we have* $M_f = M^{n-1}$.

The proof for Corollary 3 is trivial and it provides a useful way to compute the FSRM. The full semantic relationship matrix for the SLN shown in Figure 5 can be computed by using Corollary 3 and the result is shown below. It is easy to testify that the semantic relationship of any two resources derived by the above matrix is consistent with the semantic relationship derived by logical reasoning using the original SLN shown in Figure 5. If n is huge, the complexity will be $O(n^5)$. We will investigate the solution to this issue in the next section.

$$\begin{pmatrix} e & \alpha & \alpha \times \beta + \alpha \times \delta \times \gamma^R & \alpha \times \delta + \alpha \times \beta \times \gamma \\ \alpha^R & e & \beta + \delta \times \gamma^R & \delta + \beta \times \gamma \\ \beta^R \times \alpha^R + \gamma \times \delta^R \times \alpha^R & \beta^R + \lambda \times \delta^R & e & \gamma + \beta \times \delta \\ \delta^R \times \alpha^R + \gamma^R \times \beta^R \times \alpha^R & \delta^R + \gamma^R \times \beta^R & \delta^R \times \beta + \gamma^R & e \end{pmatrix}.$$

Corollary 4: *For a semantic relationship matrix M and its FSRM* M_f, *we have* $M_f \times M = M_f$.

Proof: We only need to verify that each element of $M_f \times M$ is equal to the corresponding one of F, i.e.,

$$l_{ij}^{\#} = \left[l_{i1}^{\#} \quad l_{i2}^{\#} \quad \ldots \quad l_{in}^{\#}\right]\begin{pmatrix} l_{1j} \\ l_{2j} \\ \ldots \\ l_{nj} \end{pmatrix} = l_{i1}^{\#} \times l_{1j} + l_{i2}^{\#} \times l_{2j} + \cdots + l_{in}^{\#} \times l_{nj}.$$

The right of the equation can be denoted as $L_{ij}^{\#}$. Obviously, we have:

$$L_{ij}^{\#} = l_{i1}^{\#} \times l_{1j} + l_{i2}^{\#} \times l_{2j} + \cdots + l_{ij}^{\#} \times l_{jj} + \cdots + l_{in}^{\#} \times l_{nj} \ge l_{ij}^{\#} \times l_{jj} = l_{ij}^{\#} \times e = l_{ij}^{\#}, \text{ i.e., } L_{ij}^{\#} \ge l_{ij}^{\#}. \tag{3}$$

$$l_{im}^{\#} \times l_{mj} = \left(\Sigma\Sigma\ldots\Sigma\left(l_{i,k1} \times l_{k1,k2} \times \cdots \times l_{k(n-3)(n-2)} \times l_{k(n-2),m}\right)\right) \times l_{mj} \le l_{ij}^{\#}, \text{ for all } 1 \le m \le n.$$

According to Characteristic 2, we have:

$$l_{ij}^{\#} \ge l_{i1}^{\#} \times l_{1j} + l_{i2}^{\#} \times l_{2j} + \cdots + l_{in}^{\#} \times l_{nj} = L_{ij}^{\#}, \text{ i.e., } L_{ij}^{\#} \le l_{ij}^{\#}. \tag{4}$$

By equation (3) and (4), we have $L_{ij}^{\#} = l_{ij}^{\#}$, which means that $M_f \times M = M_f$. □

4.4 Partitioned semantic matrix and SLN management

The matrix-based reasoning and management operations will be complex and time-consuming when the semantic relationship matrix becomes huge. Fortunately, the SLN can be departed into many isolated parts and each of them belongs to different domains. These parts are almost irrelevant and the semantic relationship belonging to two different parts can be regarded as *NULL*. So the semantic relationship matrix for the whole SLN can be departed into many small matrices which can be dealt with easily.

Suppose that the whole SLN S can be departed into n isolated parts denoted as S_{P1}, S_{P2}, ..., S_{Pn} and the corresponding semantic relationship matrices are M and P_1, P_2, .., P_n, then the semantic relationship matrix S takes the following form.

$$M = \begin{pmatrix} P_1 & N & \dots & N \\ N & P_2 & \dots & N \\ \dots & \dots & \dots & \dots \\ N & N & \dots & P_n \end{pmatrix}.$$

Each N represents a matrix block in which element(s) are all *Null*. If not specified, N stands for the same meaning in the following discussion.

For the convenience of discussion, we take $n = 2$ as an example in the following, i.e., the whole SLN is departed into two parts, other cases can be discussed similarly.

Based on the above assumption and the characters we have discussed, the following results can be easily deduced.

Lemma 3: *For an SLN S, if S_P and S_Q are two isolated components, M, P and Q are the corresponding semantic relationship matrices, i.e.,*

$$M = \begin{bmatrix} P & N \\ N & Q \end{bmatrix}.$$

So,

$$M^k = \begin{bmatrix} P^k & N \\ N & Q^k \end{bmatrix}$$

holds, where k is a positive integer.

Lemma 3 can be easily proved according to the multiplication characters of the matrix and the laws of semantic addition and semantic multiplication.

Theorem 2: *Suppose an SLN S consists of S_P and S_Q, which are two isolated components, M, P and Q are the corresponding semantic relationship matrices for S, S_P and S_Q, we have*

$$M_f = \begin{bmatrix} P_f & N \\ N & Q_f \end{bmatrix}.$$

Proof: Suppose that m, m_1 and m_2 are the ranks for M, P and Q, $m = m_1 + m_2$. According to Corollary 3, Corollary 4 and Lemma 3, we have:

$$M_f = M^{m-1} = \begin{bmatrix} P^{m-1} & N \\ N & Q^{m-1} \end{bmatrix}$$
$$= \begin{bmatrix} P_f P^{(m-m_1)} & N \\ N & Q_f Q^{(m-m_2)} \end{bmatrix}$$
$$= \begin{bmatrix} P^{(m_1-1)+(m-m_1)} & N \\ N & Q^{(m_2-1)+(m-m_2)} \end{bmatrix}$$
$$= \begin{bmatrix} P_f & N \\ N & Q_f \end{bmatrix}.$$

Hence the theorem holds. □

According to Theorem 2, if an SLN consists of two isolated parts, the corresponding FSRM can be composed by two sub-SLN's FSRM. And Theorem 2 can be easily extended to the cases of n sub-parts, so we get the following corollary.

Corollary 5: *Suppose an SLN S consists of n isolated components, M, P_1, P_2, ..., and P_n are the corresponding semantic relationship matrices for S and S_{P1}, S_{P2}, ..., and S_{Pn}, then*

$$M_f = \begin{pmatrix} P_{1f} & N & \dots & N \\ N & P_{2f} & \dots & N \\ \dots & \dots & \dots & \dots \\ N & N & \dots & P_{nf} \end{pmatrix} \textit{ holds.}$$

The proof for Corollary 5 is similar to that of Theorem 2. As have discussed, it is a time-consuming process to compute the FSRM for an SLN. Theorem 2 and Corollary 5 provide a solution to this issue.

During computing an FSRM, it is obvious that the multiplication is the primary and time-consuming operation. For a given SLN, suppose that it involves m resources, the rank for its corresponding SRM is m. In order to compute the element in FSRM, we can compute M^{n-2} first. According to Theorem 1, m^3 multiplication operations are needed to compute an element in FSRM. So the complexity for computing its FSRM is $O(m^5)$.

According to Theorem 2 and Corollary 5, if an SLN is composed with two or more separate parts, its FSRM can be formed through merging FSRMs for all compositive parts. Suppose m_1, m_2, …, and m_n are the corresponding ranks for P_1, P_2, ..., and P_n, the complexity for computing FSRM for the whole SLN is $O(m_1^5 + m_2^5 + \cdots + m_n^5)$. On the assumption that all parts are equal, the complexity will be $O((m/n)^5 \times n))$, i.e., $O(m^5/n^4)$. That means the complexity decreases while n climbs to the limitation m.

Clearly, it is also important to separate the whole SLN into some isolated parts for checking the consistency and managing the SLN. Instead of checking the consistency of the whole SLN, we can check each part. If each parts is consistent individually, so is the whole SLN and vice versa.

Of course it is more efficient to examine the parts than to examine the whole. Meanwhile, the management operations for SLN are also easier and more efficient by using the partitioned semantic matrix. On one hand, almost all management operations need to verify the consistency. On the other hand, almost each operation on the whole SLN can be turned to one of the parts. And the operations on the part are certainly more efficient than those on the whole.

5 MATRIX-BASED SLN CONSISTENCY MAINTENANCE AND OPERATIONS

5.1 Semantic consistency maintenance

The information provided by the current web is isolated in semantics, so it does not have the semantic consistency issue. However, it is vital to ensure the consistency for the SLN because the inconsistency in SLN will damage logical reasoning (Baclawski et al., 2002). As discussed in Section 3, an inconsistent issue occurs while an impossible semantic relationship takes place between two resources. For example, the semantic relationship $\alpha + \alpha^N$ from a resource to another means a confliction. However, it is important to detect the inconsistent issue for an SLN. Fortunately, the semantic relationship matrix provides a useful tool for the detection. While scanning the FSRM, we can get the reliable semantic relationship between any two resources. So the next is to decide whether the semantic relationship is compatible or not. Conflict rules and a domain-dependent conflict list help the maintenance mechanism to decide the inconsistency in the SLN with the conflict rules and list. A conflict rule is used to decide whether a semantic relation means a confliction. Table 1 shows some conflict rules.

Table 1 *Examples for conflict rules (α is any possible semantic factor in the table)*

Conflict rules	*Instruction*
$\alpha + \alpha^N$ means a confliction	Obviously, α and α^N are two incompatible semantic factors
$\alpha + \phi$ means a confliction ($\alpha \neq \phi$)	α and ϕ can not occur in the same semantic link
op + *e* means a confliction	*op* and *e* are contrary in semantic meaning
op + *imp* means a confliction	*op* and *imp* are contrary in semantic meaning
op + *sub* means a confliction	*op* and *sub* are contrary in semantic meaning

A conflict list depends on application domain and varies with the practical semantic link network and it can be made and modified by domain experts. A conflict list lists all potential cases that can lead to a confliction in the practical domain.

With conflict rules and the conflict list, a program can detect any inconsistent semantic relationship in an FSRM. Once a confliction occurs, the most important thing is to eliminate it by modifying the related semantic links. Although an intelligent system can help to detect the inconsistency and find the outlaw semantic links, domain experts are responsible for the dominating task.

5.2 SLN manipulation

The SLN provides richer semantics than the hyperlink web. However, its maintenance cost is higher, because any operation on resources or semantic links may affect the semantics of the whole SLN. For example, while deleting a semantic link, what we should do is not only to delete the semantic link but also to consider updating some other involved semantic links via logical reasoning. In many cases, this is complex and time consuming. Fortunately, the semantic matrix helps us to manipulate an SLN. The SLN manipulation concerns two aspects: resource operation and semantic link operation, which consist of adding, deleting and updating. So there are six types of basic operations for an SLN.

Add a new semantic link. A new semantic link needs to be added to an SLN when one of the following cases occurs:

- a brand new semantic relationship is added between two resources
- adding a corresponding semantic link may improve the reasoning efficiency while a semantic relationship is frequently used in reasoning
- other cases needed to add a new semantic link, such as after adding some new resources.

When a new semantic link is added, the corresponding semantic relationship needs to be provided. This operation is easy if semantic relationships can be derived by reasoning. All we should do is to add the new link with the corresponding semantic relationship.

However, if the attached semantic relationship is a new one, we can deal with it by the following steps:

- decide whether the new semantic relationship conflicts with semantic links among the SLN or not
- if there exists conflict, the operation should be cancelled, otherwise the semantic link should be added to the SLN. The first step is the key and should be completed by the FSRM.

Delete a semantic link. Sometimes a semantic link should be removed from the SLN. If the corresponding semantic factor with the removed semantic link can be retrieved from the SLN, the semantic link can be deleted directly and the corresponding semantic relationship should be set as *Null* in the semantic matrix. If the semantic relationship cannot be retrieved from the SLN, there may exist some other semantic links affected by the deleted semantic links. The following three operations should be carried out:

- set the corresponding factor to ϕ
- compute the full semantic relationship matrix again and find all elements that become ϕ and for every one of them, e.g., $l_{ij} = \phi$, if there exists a semantic link from r_i to r_j, then delete it
- delete the semantic link.

Update a semantic link. This operation consists of the following two steps:

- delete the old semantic link
- add the new one. we can do them as discussed above.

Add a new resource. When adding a new resource, the semantic factors between the new resource and the other resource should be provided. We can add resources to the SLN first and then add the corresponding semantic links to the SLN according to the add operation as discussed above.

Delete a resource. While deleting a resource, we should remove all the involved semantic links and should update other semantic links after removing the resource. For the corresponding semantic matrix, the corresponding row and column should be deleted and the SLN should be refreshed.

Update a resource. Similar to the semantic link updating, this operation consists of the following two steps:

- delete the old resource
- add a new resource to the SLN. Both steps can be completed according to the above discussion.

Ensuring the consistency of an SLN is vital for resource and semantic link operations. The semantic relationship matrix and the FSRM provide a tool for the manipulation operations. All these management operations can be reflected in the matrix. Table 2 lists the operation mapping between SLN and the semantic relationship matrix.

Table 2 *The operation mapping between SLN and semantic relationship matrix*

Manipulation operations	*The operations among matrix*	*Instruction*
Add a new semantic link	Append a semantic factor to the corresponding link	Consistency checking needed
Delete a semantic link	Delete a semantic factor from the corresponding element	Affected semantic links needs to be updated
Update a semantic link	Change one semantic factor of the corresponding element to another	Consistency checking needed
Add a new resource node	Add a row and a column for the new resource	Some semantic links are added
Delete a resource node	Delete the corresponding row and column	Involved semantic links are deleted
Update a resource	Replace the corresponding row and column with a new one for the new resource	The corresponding semantic links are updated

6 PROTOTYPE FOR BUILDING AND USING SLN

6.1 SLN-builder

The SLN-Builder is a software tool that can add, delete, verify and store a semantic link. Its source input is a pure text document and final outputs are two XML document descriptions: document-content XML descriptions and semantic-link XML descriptions.

For a given document, the SLN-Builder provides users with an interface to add a semantic link. Figure 7 shows the interface for adding a semantic link. After adding a semantic link, SLN-Builder must execute verification function to ensure the semantic link to satisfy the data structure defined in the SLN-Builder. SLN-Builder stores the document in XML descriptions. SLN-Builder can also support the delete of a semantic link. Users can delete a semantic link from the main interface of the SLN-Builder.

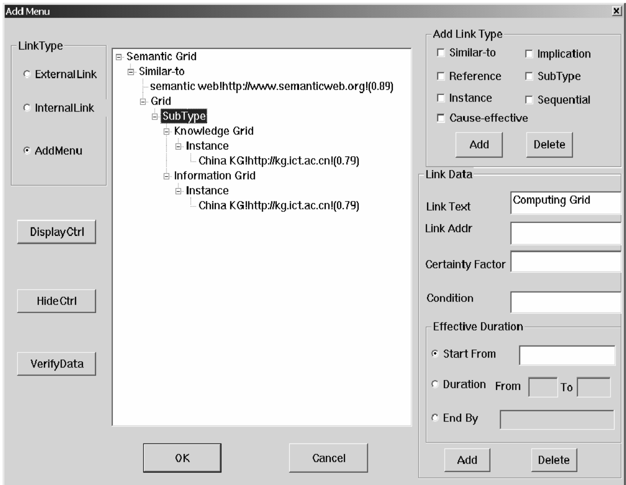


Figure 7 *Interface for adding a semantic link*

6.2 Intelligent browser

Intelligent browser consists of the following components: FSRM generator, query engine, HTML-Converter and browser interface. For every SLN, FSRM generator is executed automatically to create a full semantic relation matrix. The query engine can query relevant results with user input in the FSRM and the query results are described with XML forms. HTML-Converter can convert the above XML descriptions into HTML files. The browser interface can provide users with a full query result in an HTML view. Figure 8 shows a browser interface for the query result.

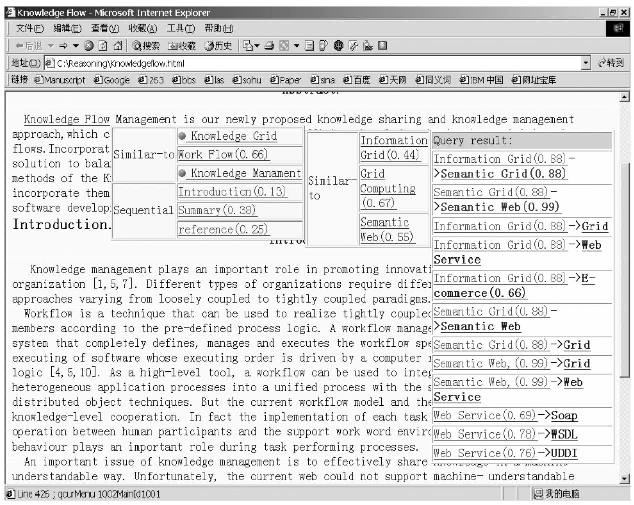


Figure 8 *Browser interface for the query result*

6.3 The role of FSRM

As discussed above, FSRM plays an important role in both reasoning process and consistency-checking process over an SLN. Through the FSRM of an SLN, we can easily obtain the accurate semantic meaning of any two given resources.

In order to answer question and provide proper solution for given problems, it is highly useful to retrieve the semantic meaning among resources over a proper SLN. By checking whether there exists any conflict element in semantic meaning in the corresponding FSRM, we can check the consistency of an SLN. Any confliction in FSRM means an inconsistency in the corresponding SLN.

The prototype discussed above includes an FSRM generator, which provides a way to compute a full semantic relation matrix automatically.

7 DISCUSSION AND CONCLUSION

Just as the construction of the World Wide Web, constructing a world wide SLN requires us to build local SLNs first and then merge them to the main. However, as discussed in Section 6.1, in order to ensure consistency, the local SLN cannot be combined to the main directly. Again we need to detect whether the local SLN agree with the main. The full semantic matrix needs to be re-computed after the merge operation.

In many cases, users just need a very small portion of the whole SLN. So a meaningful part should be generated from the main. The semantic matrix is a useful tool to divide an SLN into small parts and to ensure their consistency and integrity.

In order to maintain the semantic consistency, the future semantic web should establish an authority certification mechanism just as the UDDI for the current web services. All the newly added SLN should be carefully verified. Considering the semantic consistency issue and the cost of maintenance, the future semantic web should include the following three layers:

- the bottom is the hyperlink network or other organisation of resources
- the middle is the uncertified SLN that can provide some useful reference information but does not guarantee its correctness
- the top is the certified SLN, which can provide provable information and explanation.

So far, we have proposed an algebra theory for organising resources by semantic links and reasoning. The theory enables the Semantic Link Network SLN to be a promising model for the semantic web.

Experimental software tools have been developed to assist users to easily define the semantic link network over a plain text and to browse the SLN-based text. Experiments show that the approach is feasible. We have carried out applications in e-science and e-culture (Zhuge, 2002; Zhuge, 2004a).

ACKNOWLEDGEMENT

We thank all team members of China Knowledge Grid Research Group (http://kg.ict.ac.cn) and Knowledge Grid Center (http://www.knowledgegrid.net) for their diligent work on relevant research and software development. This research is supported by National Basic Research Program (No.2003CB317000) and National Science Foundation (No. 60503047) of China.

REFERENCES

Allan, J. (1997) 'Building hypertext using information retrieval', *Information Processing and Management*, Vol. 33, No. 2, pp.145–159.

Baclawski, K., Kokar, M.M., Waldinger, R. and Kogut, P.A. (2002) 'Consistency checking of semantic web ontologies', *Proceedings of International Semantic Web Conference ISWC*, pp.454–459.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001) 'The semantic web', *Scientific American*, Vol. 284, No. 5, pp.34–43.

Bray, T., Paoli, J. and Queen, C.S-M. (1997) 'Extensible markup language (XML)', *The World Wide Web Journal*, Vol. 2, No. 4, pp.29–66.

Broekstra, J. *et al*. (2001) 'Enabling knowledge representation on the web by extending RDF schema', *Proc. 10th Int'l WWW Conf.*, May, Hong Kong, http://www.cs.vu.nl/~frankh/abstracts/www01.html.

Decker, S. *et al.* (2000) 'The semantic web: the roles of XML and RDF', *IEEE Internet Computing*, September–October, pp.63–74.

Fensel, D., van Harmelen, F., Klein, M. *et al.* (2000) 'On-to-knowledge: ontology-based tools for knowledge management', *Proceedings of the eBusiness and eWork 2000 (EMMSEC 2000) Conference*, Madrid, Spain, 18–20 October.

Fikes, R., Hayes, P. and Horrocks, I. (2002) *DAML Query Language*, Available from http://www.daml.org/dql/.

Foster, I., Kesselman, C., Nick, J.M. and Tuecke, S. (2002) *Grid Services for Distributed System Integration, IEEE Computer*, June, pp.37–46.

Hendler, J. (2001) 'Agents and the semantic web', *IEEE Intelligent Systems*, March–April Vol. 16, No.2, pp.30–37.

Hendler, J. (2003) 'Science and the semantic web', *Science*, January, Vol. 299, pp.520, 521.

Henzinger, M.R. (2001) 'Hyperlink analysis for the web', *IEEE Internet Computing*, Vol. 5, No. 1, pp.45–50.

Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D. and Scholl, M. (2002) 'RQL: a declarative query language for RDF', *WWW*, pp.592–603.

Klein, M. (2001) 'XML, RDF, and relatives', *IEEE Internet Computing*, Vol. 5, No. 2, pp.26–28.

Lassila, O. and Swick, R.R. (1999a) 'Resource description framework (RDF): model and syntax specification', *Recommendation, World Wide Web Consortium*, February, See http://www.w3.org/TR/1999/REC-rdf- syntax -19990222/.

Lassila, O. and Swick, R.R. (1999b) 'Resource description framework (RDF) model and syntax specification', http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.

Tudhope, D. and Taylor, C. (1997) 'Navigation via similarity: automatic linking based on semantic closeness', *Inf. Process. Manage.*, Vol. 33, No. 2, pp.145–159.

Zhuge, H. (2002) 'Clustering soft-devices in semantic grid', *IEEE Computing in Science and Engineering*, Vol. 4, No. 6, pp.60–63.

Zhuge, H. (2003) 'Active document framework ADF: model and tool', *Information and Management*, Vol. 41, No. 1, pp.87–97.

Zhuge, H. (2004a) 'China e-science knowledge grid environment', *IEEE Intelligent Systems*, Vol. 19, No. 1, pp.13–17.

Zhuge, H. (2004b) *The Knowledge Grid*, World Scientific Publishing Co., Singapore.

Zhuge, H., Sun, Y. and Guo, W. (2003) 'Theory and algorithm for rule base refinement', *Proceedings of the 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-2003)*, Springer, LNAI 2718, pp.187–196.