Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Information & Management 41 (2003) 87–97

INFORMATION & MANAGEMENT

www.elsevier.com/locate/dsw

# Active e-document framework ADF: model and tool

Hai Zhuge*

*Knowledge Grid Research Group, Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, P.O. Box 2704, Beijing 100080, China*

**Abstract**

An active document framework is a self-representable, self-explainable, and self-executable document mechanism. A document's content is reflected in four aspects: granularity hierarchy, template hierarchy, background knowledge, and semantic links between fragments. An active document has a set of build-in engines for browsing, retrieving, and reasoning, which can work in a way best suited to the document's content. Besides browsing and retrieval services, the active document supports intelligent information services such as complex question answering, online teaching, and assistant problem solving. The client side service provider is only responsible for the retrieval of the required active document. The detailed information services are provided by the document mechanism. This improves the current Web information retrieval approach by raising the efficiency of information retrieval, enhancing the preciseness and mobility of information services, and enabling intelligent information services. A tool for making semantic links in a document and an intelligent browser have been developed to support the proposed approach, which provides a new type of web information service.

*Keywords:* e-Document; Information service; Rules; Semantic link; Web

## 1. Introduction

Current Web information services are evolving from low to high level and from simple to complex. One basic service is *information retrieval* to retrieve the required information from large-scale web information resources. In order to provide web users succinct and useful content, much research has been performed on methods to provide information extraction and filtration. The basic premise underlying *traditional information retrieval* is that related documents use the same words. If two documents share enough terms, then they are regarded as similar. But these approaches encounter synonym and polysemy problems. Efforts to deal with these issues have met with limited success [7].

One higher-level service in document processing is *question answering*, i.e. replying to questions about the given document. Current approaches only focus on answering simple questions whose solutions (usually sentences) can be found in the document *exactly as stated*. Efforts to realize complex question answering have been made but they have achieved very limited success. Previous work focuses mainly on the representation, understanding, and processing of very fine-granularity documents like words and sentences [2].

Current Internet users require more intelligent services, such as:

* Tel.: +86-10-62565533x5690; fax: +86-10-62567724.
*E-mail address:* zhuge@ict.ac.cn (H. Zhuge).
*URL:* http://kg.ict.ac.cn.

0378-7206/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0378-7206(03)00029-6

(1) *Precise information retrieval*, i.e. the retrieval result must satisfy the user's requirement and not include irrelevant information;
(2) *Complex question answering*, providing a solution that cannot be found at the word-level content or sentence-level content of the document; and,
(3) *Assistance to solve problems*, to find the document or the fragment that includes the solution.

The main obstacle to provide intelligent services is that the common search engine cannot utilize the content or semantics of the HTML-based Web documents, and current web documents do not reflect their content. The efficacy of using the approaches of analysis and generation of the hypertext link is very limited [1,11,18,21,22].

The Semantic Web aims at providing services based on the machine-understandable Web resources. Research concerns intelligent indexing and semantic retrieval [20], knowledge management [5,19], the ontology and service markup languages [6,9,10,16].

Markup languages developed by industry can be used to represent a document's content to some extent [4,12], e.g. XML (http://www.w3.org/TR/REC-xml) can be used to reflect the structural information of the document, and this is helpful in raising the preciseness of the information retrieval. The XML-based Resource Description Framework (RDF), (see http://www.w3c.org/rdf) defines some machine-understandable semantics of web resources using the object-attribute-value model. The RDF schema (RDFS) enhances the representation ability of the RDF by providing the means for defining the vocabulary, the class-based structure, and the constraints for expressing the metadata about Web resources. An approach for representing knowledge by extending the RDF schema has been proposed [3].

Ontologies are regarded as a key to support information exchange across various networks, and they can be used to enhance the document content [14,15]. The ontology of a particular domain establishes a common understanding between people. It usually contains a hierarchy of concepts in a domain and describes each concept's crucial properties by using an attribute-value. Tools for assisting in the creation and management of ontologies have been reported. Approaches for representing knowledge within documents have been proposed; these include the Web Knowledge Base (WebKB) [17], the frame-based Simple HTML Ontology Extensions (SHOE) [8], and the Ontology Inference Layer (OIL ). A service markup language DAPA Agent Markup Language (DAML ) is under development for mobile Web services like service discovery, execution, composition, and inter-operation. The markup language DAML + OIL (see http://www.daml.org/2000/10/daml-oil) currently under development intends to combine the best features of the DAML, the OIL, the SHOE, and the RDF, so as to enable the markup and manipulation of complex taxonomic and logical relationships between Web resources.

A document is a kind of communication media between people. During the content transformation process from the writer to the reader, the content will be distorted due to differences in writing style and understanding of mutual meaning of words and sentences. The content distortion will become more serious when the writer and the reader do not share knowledge of the same field or at the same level. Current Web-document processing is faced with the same issue. Traditional content representation approaches (like the SVM) are based on very fine document fragments, such as words and phrases. But small granularity semantics cannot directly arrive at large-granularity semantics. Software tools without any ontology cannot achieve a satisfactory efficacy of understanding. A document incorporating the relevant ontology can be processed or understood more easily and accurately by tools. Unfortunately, the capability of the current ontology is not strong enough to support effective understanding.

In our approach, a document content transformation is carried out not only for its ontology but also for its background knowledge, structural knowledge, and semantic knowledge; in these, the information content will be kept at the highest level during the writing and understanding processes. The content of a document is described top-down in four ways: its *granularity hierarchy*, *template hierarchy*, *typed semantic links*, and *background knowledge*.

## 2. Document content

### 2.1. Granularity hierarchy

A document has two types of granularities: *document* and *content*. *Document granularity* depends on its size. A high-granularity document usually has several low-granularity documents. The structural hierarchy of a document is a natural partition of granularity document fragments. The document granularity of an ancestor in the hierarchy is larger than that of its successor.

*Content granularity* is defined by the abstraction relationship between document fragments. In a content granularity hierarchy, a predecessor is more abstract than its successor. Generally, a high document granularity fragment may not lead to a high content granularity fragment. But people usually divide a document into fragments according to their understanding of the content in order to help other people understand the document, e.g. a paper's content is usually organized as several sections. In the following, the term *document granularity* implies *content granularity* when the term *granularity* is used.

Granularity hierarchy provides a problem-solving approach based on algebra theory and an analogy approach [24]: if a problem does not have a solution at a high-abstraction level then it will not have a solution at a low-abstraction level; if a problem has a solution at a low-abstraction level then it must have a solution at a high-abstraction level. Similarly, we have the following principle with relation to the granularity hierarchy.

### 2.2. Principle

If the solution to a problem can be found at a small-granularity level, it can also be found at a large-granularity level. Also, if the solution to a problem cannot be found at a large-granularity level, it cannot be found at a small-granularity level.

This allows us to provide a reasoning mechanism for problem-solving or question answering across different granularity levels if the solution is difficult to find at the currently examined granularity level.

### 2.3. Template hierarchy and background knowledge

Documents belonging to the same category can be described by a single template. For example, a research paper's template could be a frame that consists of a title, author(s), abstract, introduction, main text, conclusion, and references. The fragments of a document can also have their templates, e.g. the introduction template of a paper can also be a frame that includes the description of the significance, the related work, and the research approach. A *template hierarchy* can thus be formed according to the relationship between the document template and its fragments' templates. The template hierarchy is helpful for understanding a document, and it can also be used to assist the automatic composition of a new document.

*Background knowledge* of a document is crucial in understanding its content [28]. It relates to a set of theories and a set of related application fields. A *theory* usually consists of a set of conceptual ontologies, a set of axioms, a set of reasoning rules, a set of methods (problem-solution pairs or processes of solving problems), and a set of constraints. A theory can have several sub-theories, each of which can further have several sub-theories. A leaf-node of the theory hierarchy can be represented as a frame: $\mathrm{Bk}_i = \{\mathit{Theory}[\mathit{ConceptOntology}, \mathit{Axioms}, \mathit{Rules}, \mathit{Methods}, \mathit{Constraints}], \mathrm{FieldID}\}$. With background knowledge, the synonym and polysemy (or homonym) issues can be reduced. Accordingly, documents can be understood more accurately than with approaches that only consider conceptual ontology.

### 2.4. Typed semantic link network

A *semantic link* is an ordered relationship between two documents. It can be represented as a pointer with a type directed from one document or document fragment (predecessor) to another (successor). A semantic link can be one of the following types:

(1) *Cause-effect*, denoted as $d\text{-ce} \rightarrow d'$, which means that the predecessor is the cause of its successor, and the successor is the effect of its predecessor. The cause-effective link is transitive, i.e. $d\text{-ce} \rightarrow d'$, $d'\text{-ce} \rightarrow d'' \Rightarrow d\text{-ce} \rightarrow d''$ holds. Cause-effective reasoning can be formed by chaining cause-effect links.

(2) *Implication*, denoted as $d\text{-imp} \rightarrow d'$, which states that the semantics of the predecessor implies to that of its successor. The implication link is transitive, i.e. $d\text{-imp} \rightarrow d'$, $d'\text{-imp} \rightarrow d'' \Rightarrow d\text{-imp} \rightarrow d''$ holds. It can help the reasoning mechanism

find semantic implication relationship between documents.

(3) *Subtype*, denoted as $d$-st $\rightarrow d'$, where the successor is a part of its predecessor. The subtype link is also transitive, i.e. $d$-st $\rightarrow d'$, $d'$-st $\rightarrow d'' \Rightarrow d$-st $\rightarrow d''$ holds.

(4) *Similar-to*, which defines that the semantics of the successor are similar to those of the predecessor, denoted as $d$-(sim, sd) $\rightarrow d'$, where sd is degree of similarity between $d$ and $d'$. Similar to the partial–inheritance relationship [25], the similar-to link is not transitive.

(5) *Instance*, denoted as $d$-ins $\rightarrow d'$, which states that the successor is an instance of the predecessor.

(6) *Sequential*, denoted as $d$-seq $\rightarrow d'$, which defines that $d$ should be browsed before $d'$, i.e. the content of $d'$ is the successor to the content of $d$. The sequential link is transitive, i.e. $d$-seq $\rightarrow d'$, $d'$-seq $\rightarrow d'' \Rightarrow d$-seq $\rightarrow d''$ holds. The transitive relationship allows the relevant sequential links to be connected to form a sequential chain.

(7) *Reference*, denoted as $d$-ref $\rightarrow d'$, which means that $d'$ is the further explanation of $d$. The reference link has a transitive characteristic, i.e. $d$-ref $\rightarrow d'$, $d'$-ref $\rightarrow d'' \Rightarrow d$-ref $\rightarrow d''$ holds.

More types of semantic links can be defined according to the application domain. A *semantic link network* (SLN) is a directed network, where the nodes are document fragments and the edges are the typed semantic links. The *main chain* of the SLN is a sequential chain that connects the main fragments of the document from the beginning to the end node. The content of a document can be wholly browsed if the browser follows the main chain. An SLN of a document is said to be *connective* if all the fragments are linked onto its main chain. A well-defined SLN of a large-scale document should be connected at all granularity levels.

The SLN can also be used for describing the semantic relationship between a set of related documents. For example, research papers about the same topic can be sequentially connected through the sequential links according to their publication date, and, in each paper, the sections can be sequentially connected according to the content dependence relationship between them. The main chain of the SLN of this paper is shown in Fig. 1.

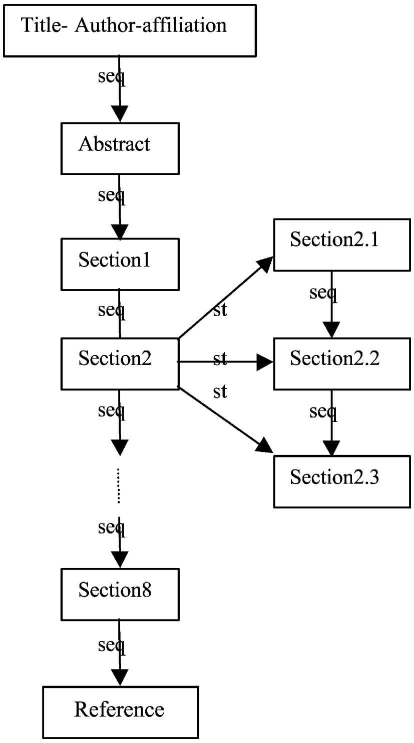


Fig. 1. The main chain of the SLN of this paper.

## 3. Document reasoning rules

*Document reasoning rules* are used for chaining the relevant semantic links and obtaining the reasoning result from the chaining; for example, if we have two links: $d$-ce $\rightarrow d'$ and $d'$-ce $\rightarrow d''$, we can obtain the result: $d$-ce $\rightarrow d''$ due to the transitive nature of the cause-effective link. The reasoning process can be represented as a rule: $d$-ce $\rightarrow d'$, $d'$-ce $\rightarrow d'' \Rightarrow d$-ce $\rightarrow d''$. It can also be represented as $\alpha \cdot \beta \Rightarrow \gamma$, where $\alpha, \beta, \gamma \in$ {ce, imp, ins, st, sim, ref, seq}, e.g. the above rule can be represented as ce · ce ⇒ ce.

A simple case of the reasoning is that all the semantic links have the same type (called single-type reasoning). According to the transitive characteristic of the semantic links, we have the following reasoning rule: $d_1$-$\alpha \rightarrow d_2, d_2$-$\alpha \rightarrow d_3, \ldots, d_{n-1}$-$\alpha \rightarrow d_n \Rightarrow d_1$-$\alpha \rightarrow d_n$, where $\alpha \in$ {ce, imp, st, ref}.

The heuristic rules for connecting different types of links are presented in Table 1. Rules 1–4 are for the connection between the cause-effective link and the

Table 1
Reasoning rules

| No. | Rules | Summarization |
|---|---|---|
| Rule 1 | $d\text{-ce} \rightarrow d', d'\text{-imp} \rightarrow d'' \Rightarrow d\text{-ce} \rightarrow d''$ | $\text{ce}\cdot\beta \Rightarrow \text{ce}$ |
| Rule 2 | $d\text{-ce} \rightarrow d', d'\text{-st} \rightarrow d'' \Rightarrow d\text{-ce} \rightarrow d''$ | $\text{ce}\cdot\beta \Rightarrow \text{ce}$ |
| Rule 3 | $d\text{-ce} \rightarrow d', d'\text{-sim} \rightarrow d'' \Rightarrow d\text{-ce} \rightarrow d''$ | $\text{ce}\cdot\beta \Rightarrow \text{ce}$ |
| Rule 4 | $d\text{-ce} \rightarrow d', d\text{-ins} \rightarrow d'' \Rightarrow d''\text{-ce} \rightarrow d'$ | $\text{ce}\cdot\beta \Rightarrow \text{ce}$ |
| Rule 5 | $d\text{-imp} \rightarrow d', d'\text{-st} \rightarrow d'' \Rightarrow d\text{-imp} \rightarrow d''$ | $\text{imp}\cdot\text{st} \Rightarrow \text{imp}$ |
| Rule 6 | $d\text{-imp} \rightarrow d', d'\text{-ins} \rightarrow d'' \Rightarrow d\text{-ins} \rightarrow d''$ | $\text{imp}\cdot\text{ins} \Rightarrow \text{ins}$ |
| Rule 7 | $d\text{-imp} \rightarrow d', d'\text{-ce} \rightarrow d'' \Rightarrow d\text{-ce} \rightarrow d''$ | $\text{imp}\cdot\text{ce} \Rightarrow \text{ce}$ |
| Rule 8 | $d\text{-imp} \rightarrow d', d'\text{-ref} \rightarrow d'' \Rightarrow d\text{-ref} \rightarrow d''$ | $\text{imp}\cdot\text{ref} \Rightarrow \text{ref}$ |
| Rule 9 | $d\text{-st} \rightarrow d', d'\text{-ce} \rightarrow d'' \Rightarrow d\text{-ce} \rightarrow d''$ | $\text{st}\cdot\text{ce} \Rightarrow \text{ce}$ |
| Rule 10 | $d\text{-st} \rightarrow d', d'\text{-imp} \rightarrow d'' \Rightarrow d\text{-imp} \rightarrow d''$ | $\text{st}\cdot\text{imp} \Rightarrow \text{imp}$ |
| Rule 11 | $d\text{-st} \rightarrow d', d'\text{-ref} \rightarrow d'' \Rightarrow d\text{-ref} \rightarrow d''$ | $\text{st}\cdot\text{ref} \Rightarrow \text{ref}$ |
| Rule 12 | $d\text{-st} \rightarrow d', d'\text{-ins} \rightarrow d'' \Rightarrow d\text{-ins} \rightarrow d''$ | $\text{st}\cdot\text{ins} \Rightarrow \text{ins}$ |
| Rule 13 | $d\text{-ins} \rightarrow d', d'\text{-ce} \rightarrow d'' \Rightarrow d\text{-ce} \rightarrow d''$ | $\text{ins}\cdot\text{ce} \Rightarrow \text{ce}$ |
| Rule 14 | $d\text{-ins} \rightarrow d', d'\text{-imp} \rightarrow d'' \Rightarrow d\text{-imp} \rightarrow d''$ | $\text{ins}\cdot\text{imp} \Rightarrow \text{imp}$ |
| Rule 15 | $d\text{-ins} \rightarrow d', d'\text{-ref} \rightarrow d'' \Rightarrow d\text{-ins} \rightarrow d''$ | $\text{ins}\cdot\text{ref} \Rightarrow \text{ref}$ |
| Rule 16 | $d\text{-ins} \rightarrow d', d_1\text{-ins} \rightarrow d_1' \Rightarrow (d\text{-seq} \rightarrow d_1)\text{-ins} \rightarrow (d'\text{-seq} \rightarrow d_1')$ | $d\text{-}\beta \rightarrow d', d_1\text{-}\beta \rightarrow d_1' \Rightarrow (d\text{-seq} \rightarrow d_1)\text{-}\beta \rightarrow (d'\text{-seq} \rightarrow d_1')$ |
| Rule 17 | $d\text{-ref} \rightarrow d', d_1\text{-ref} \rightarrow d_1' \Rightarrow (d\text{-seq} \rightarrow d_1)\text{-ref} \rightarrow (d'\text{-seq} \rightarrow d_1')$ | Same to the above |
| Rule 18 | $d\text{-seq} \rightarrow d', d_1\text{-seq} \rightarrow d_1' \Rightarrow (d\text{-seq} \rightarrow d_1)\text{-seq} \rightarrow (d'\text{-seq} \rightarrow d_1')$ | Same to the above |
| Rule 19 | $d\text{-ce} \rightarrow d', d_1\text{-ce} \rightarrow d_1' \Rightarrow (d\text{-seq} \rightarrow d_1)\text{-ce} \rightarrow (d'\text{-seq} \rightarrow d_1')$ | Same to the above |
| Rule 20 | $d\text{-imp} \rightarrow d', d_1\text{-imp} \rightarrow d_1' \Rightarrow (d\text{-seq} \rightarrow d_1)\text{-imp} \rightarrow (d'\text{-seq} \rightarrow d_1')$ | Same to the above |
| Rule 21 | $d\text{-st} \rightarrow d', d_1\text{-st} \rightarrow d_1' \Rightarrow (d\text{-seq} \rightarrow d_1)\text{-st} \rightarrow (d'\text{-seq} \rightarrow d_1')$ | Same to the above |
| Rule 22 | $d\text{-sim} \rightarrow d', d_1\text{-sim}{>}d_1' \Rightarrow (d\text{-seq} \rightarrow d_1)\text{-sim} \rightarrow (d'\text{-seq} \rightarrow d_1')$ | Same to the above |

others. Rules 5–8 are for the connection between the implication and the other links. Rules 9–12 are for the connection between the sub-type and other links. Rules 13–15 are for the connection between the instance and the other links. Rules 16–22 show that the sequential connection satisfies *additivity*, i.e. any two links with the same type can be added by sequentially connecting their predecessors and successors. These rules can be formally proved after formally defining the semantic links. To avoid complex formal statements, the reasoning rules in Table 1 are introduced as heuristic rules for supporting reasoning.

An order relationship exists between these semantic links: ref $\leq$ ins $\leq$ st $\leq$ imp $\leq$ ce, where the rightmost reflects a stronger relationship between two documents than the one on its left. In order to obtain a good reasoning result, the reasoning mechanism should find the strongest link between the candidate links. Summarizing the rules, we have the following characteristic.

**Characteristic 1**. For a connection: $\alpha{\cdot}\beta$, if $\beta \leq \alpha$, then $\alpha \cdot \beta \Rightarrow \alpha$ will hold.

Semantic links can also be inexact. An inexact semantic link reflects the possibility of its existence. We use a *degree of certainty* cd to reflect such a possibility. Therefore, an inexact semantic link can be represented as: $d\text{-}(\alpha, \text{cd}) \rightarrow d'$, where $\alpha \in \{\text{ce, imp, st, sim, ins, ref}\}$. Inexact single-type reasoning takes the following forms: $d_1\text{-}(\alpha, \text{sd}_1) \rightarrow d_2$, $d_2\text{-}(\alpha, \text{sd}_2) \rightarrow d_3, \ldots, d_n\text{-}(\alpha, \text{sd}_n) \rightarrow d_{n+1} \Rightarrow d_1\text{-}(\alpha, \text{sd}) \rightarrow d_n$, where $\text{sd} = \eta(\text{sd}_1, \ldots, \text{sd}_n)$, $\eta$ maps $\{\text{sd}_1, \ldots, \text{sd}_n\}$ into [0, 1].

Different types of inexact semantic links can be chained according to the rules. For example, Rule 1 can be extended as the following inexact rule: $d\text{-(ce, cd}_1) \rightarrow d'$, $d'\text{-(imp, cd}_2) \rightarrow d'' \Rightarrow d\text{-(ce}, \mathit{Min}(\text{cd}_1, \text{cd}_2)) \rightarrow d''$. Other inexact rules can be similarly formed. Another type of inexactness is caused by the similar-to link, e.g. connecting the cause-effective link with the similar-to link can produce the following inexact reasoning rules: $d\text{-ce} \rightarrow d'$, $d'\text{-(sim, sd)} \rightarrow d'' \Rightarrow d\text{-(ce, cd)} \rightarrow d''$, where cd depends on sd (cd = sd is one possible choice).

## 4. Active document framework

Traditional e-documents are passive, as are hardcopy documents. People need a search engine that can retrieve documents and browse their content manually. A passive document has two shortcomings:

- The user will feel it is inefficient in getting the required information when browsing a large-scale document;
- The search engine does not have any background knowledge of the document's content, so it cannot provide the ideal information service.

An active document encapsulates the textual document, its content, and the possible operations that may be performed on it. Such an active document can work like a teacher with a textbook. The reader/learner does not necessarily know much background knowledge of the document and may not have good reading skills, but he or she can learn from the document. This feature can enhance the quality of information services for a large-scale document or a large collection of inter-related documents.

An active document (AD) is a function of the input requirement ($I$). The output ($O$), corresponding to the input, depends on the content of the document ($C$) and a set of engines ($E$) that transform from $I$ to $C$. So an active document can be described as a function: $O = \mathrm{AD}(I, C, E)$.

The document content consists of the structural knowledge (SK), the background knowledge, and the semantic link network (SLN), represented as: $C = \langle \mathrm{SK}, \mathrm{BK}, \mathrm{SLN} \rangle$.

$\mathrm{SLN} = \langle \mathrm{FS}, \mathrm{LINK} \rangle$, where FS is a set of different granularity document fragments and LINK is a set of semantic links between the fragments. The reader working with the network can be regarded as a kind of workflow [13,23,26,29], where the reader can be one or more people and can be at a single location or

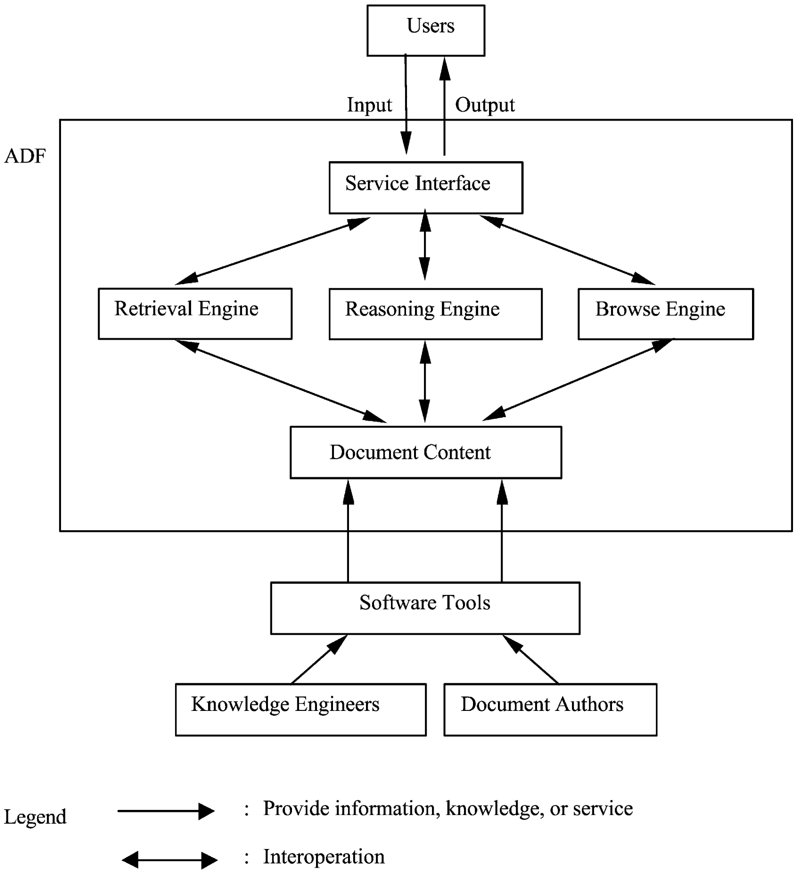


Fig. 2. Architecture of the ADF.

geographically distributed. The difference is that the flows are multiple types of the semantic links while the flows of the former are control flows, so we term this network *text-flow* to differentiate it from normal workflow. SLN can provide different views for simplifying this operation. A view of SLN only consists of one type of link between fragments.

An active document engine consists of three components:

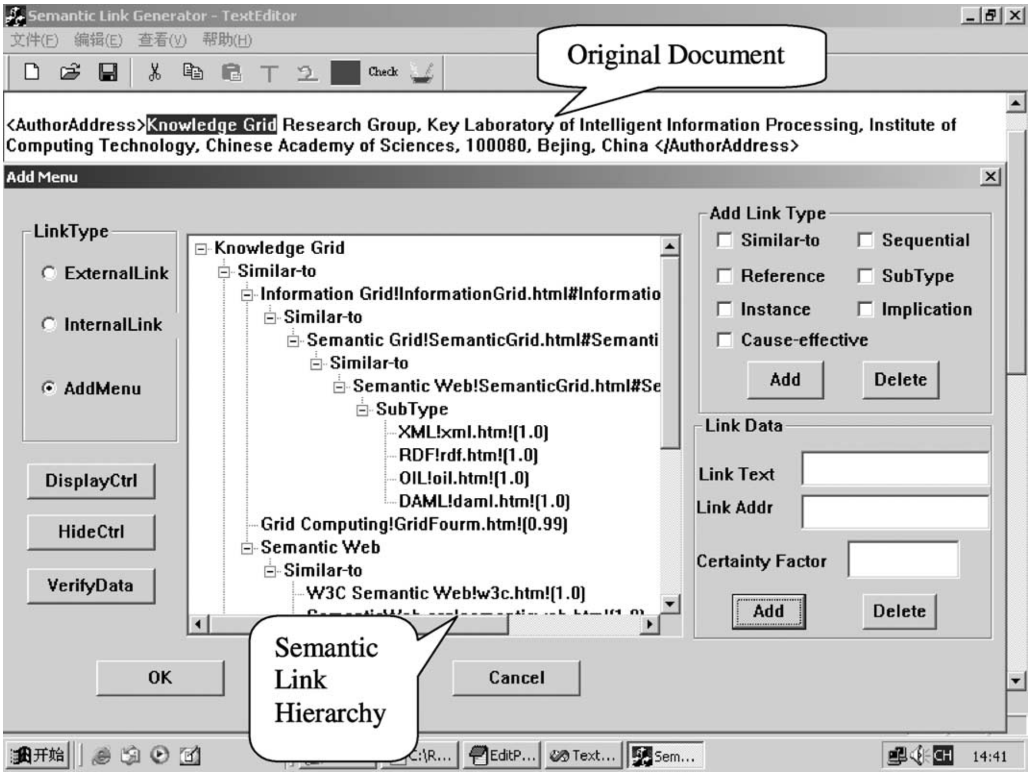
1. An *execution engine*, which is responsible for the execution of the text-flow according to the order view of its SLN, like the workflow engine [26];
2. A *search engine*, which is responsible for searching the fragment that matches the input requirement according to the SLN, the reasoning rules enable the flexible search result; and
3. A *reasoning engine*, which is responsible for reasoning according to the SLN and the rules. Any web user can retrieve an active document using a search engine and can activate it to ask for a service whose operations are done by its internal engine.

The architecture of an ADF is shown in Fig. 2.

An active document category is a set of documents that share the same background knowledge. To avoid redundancy, an active document can only contain the structural knowledge and the semantic link network. The background knowledge is shared by all the active documents in the category. The category can be represented as: $\mathrm{ADC} = \langle\{\mathrm{AD}_1, \ldots, \mathrm{AD}_n\}, BK\rangle$, the output of $\mathrm{AD}_i$ is represented as $O_i = \mathrm{AD}_i(I_i, C_i, E_i)$ and $C_i = \langle \mathrm{SK}_i, \mathrm{SLN}_i\rangle$.

## 5. Tool for making semantic link and intelligent browser

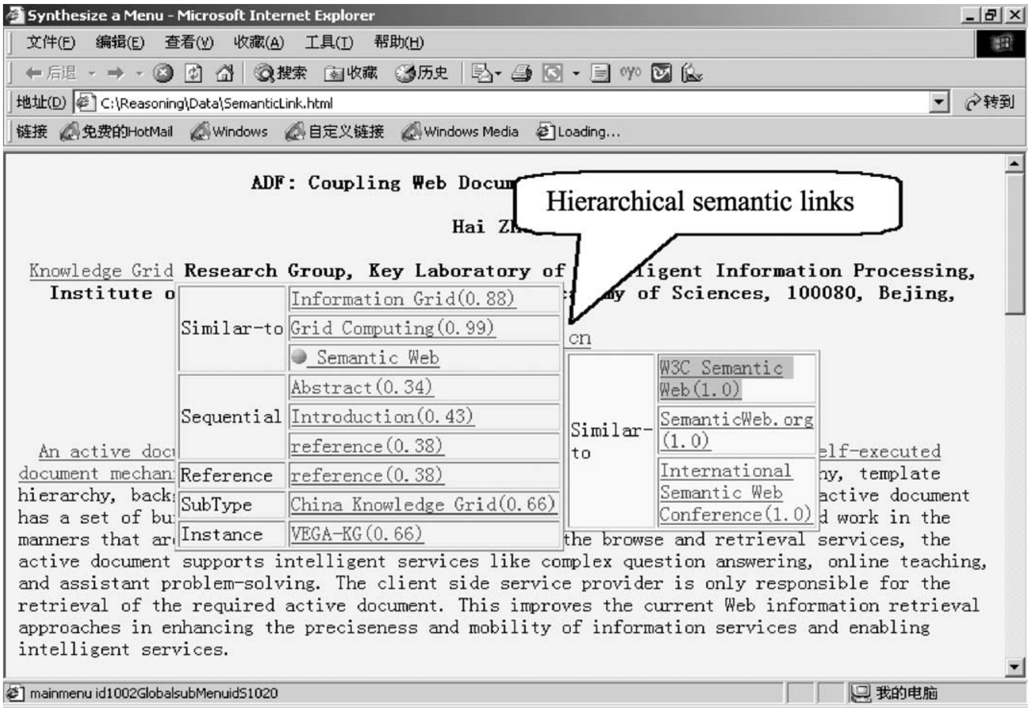
A software tool for making the semantic links in plain texts has been developed. Fig. 3 shows the interface for markup of the semantic link, where the



<AuthorAddress>Knowledge Grid Research Group, Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, 100080, Bejing, China </AuthorAddress>

```
⊟ Knowledge Grid
  ⊟ Similar-to
    ⊟ Information Grid!InformationGrid.html#Informatio
      ⊟ Similar-to
        ⊟ Semantic Grid!SemanticGrid.html#Semanti
          ⊟ Similar-to
            ⊟ Semantic Web!SemanticGrid.html#Se
              ⊟ SubType
                  XML!xml.htm!(1.0)
                  RDF!rdf.htm!(1.0)
                  OIL!oil.htm!(1.0)
                  DAML!daml.htm!(1.0)
    Grid Computing!GridFourm.htm!(0.99)
    ⊟ Semantic Web
      ⊟ Similar-to
          W3C Semantic Web!w3c.htm!(1.0)
```

Fig. 3. The Interface for the tool for making semantic links.


Synthesize a Menu - Microsoft Internet Explorer
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退 搜索 收藏 历史
地址(D) C:\Reasoning\Data\SemanticLink.html
转到
链接 免费的HotMail Windows 自定义链接 Windows Media Loading...
Hierarchical semantic links
mainmenu id1002GlobalsubMenuidS1020
我的电脑

ADF: Coupling Web Docu

Hai Z

Knowledge Grid Research Group, Key Laboratory of ligent Information Processing, Institute o y of Sciences, 100080, Bejing,

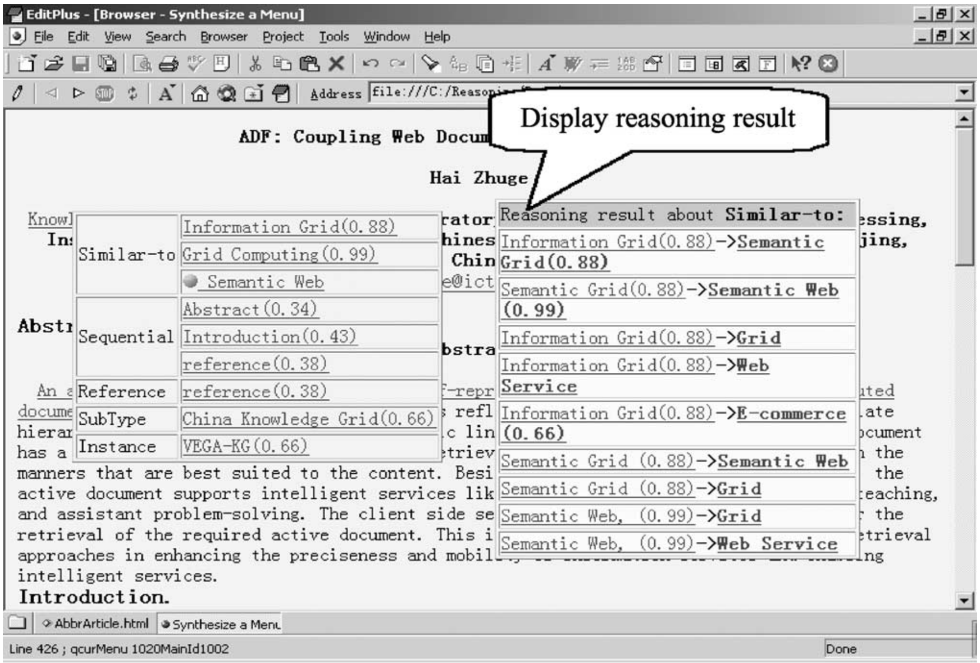
cn

| | |
|---|---|
| Similar-to | Information Grid(0.88)<br>Grid Computing(0.99)<br>● Semantic Web |
| Sequential | Abstract(0.34)<br>Introduction(0.43)<br>reference(0.38) |
| Reference | reference(0.38) |
| SubType | China Knowledge Grid(0.66) |
| Instance | VEGA-KG(0.66) |

| | |
|---|---|
| Similar-to | W3C Semantic Web(1.0)<br>SemanticWeb.org (1.0)<br>International Semantic Web Conference(1.0) |

An active doc elf-executed document mechan ny, template hierarchy, back active document has a set of bu d work in the manners that ar the browse and retrieval services, the active document supports intelligent services like complex question answering, online teaching, and assistant problem-solving. The client side service provider is only responsible for the retrieval of the required active document. This improves the current Web information retrieval approaches in enhancing the preciseness and mobility of information services and enabling intelligent services.

Fig. 4. Browsing semantic-linked text.


EditPlus - [Browser - Synthesize a Menu]
File Edit View Search Browser Project Tools Window Help
Address file:///C:/Reaso
Display reasoning result
AbbrArticle.html
Synthesize a Menu
Line 426 ; qcurMenu 1020MainId1002
Done

ADF: Coupling Web Docu

Hai Zhuge

Knowl rator ssing, In hines jing, Chin e@ict

Abstr bstra

| | |
|---|---|
| Similar-to | Information Grid(0.88)<br>Grid Computing(0.99)<br>● Semantic Web |
| Sequential | Abstract(0.34)<br>Introduction(0.43)<br>reference(0.38) |
| Reference | reference(0.38) |
| SubType | China Knowledge Grid(0.66) |
| Instance | VEGA-KG(0.66) |

Reasoning result about **Similar-to:**

- Information Grid(0.88)->**Semantic Grid(0.88)**
- Semantic Grid(0.88)->**Semantic Web (0.99)**
- Information Grid(0.88)->**Grid**
- Information Grid(0.88)->**Web Service**
- Information Grid(0.88)->**E-commerce (0.66)**
- Semantic Grid (0.88)->**Semantic Web**
- Semantic Grid (0.88)->**Grid**
- Semantic Web, (0.99)->**Grid**
- Semantic Web, (0.99)->**Web Service**

An a f-repr ted docume s refl ate hierar c lin ocument has a triev the manners that are best suited to the content. Besi the active document supports intelligent services lik eaching, and assistant problem-solving. The client side se the retrieval of the required active document. This i trieval approaches in enhancing the preciseness and mobil ng intelligent services.

**Introduction.**

Fig. 5. Displaying reasoning result during browsing.

background is the text and the front window displays the main markup functions. The user can click the button "T" to display the front window that contains the markup functions and the semantic link hierarchy. A certainty factor can be attached to each semantic link to reflect the user's certainty degree for each semantic link.

An intelligent browser was developed to browse the document with the help of the hierarchical semantic links and to enable semantic link reasoning. Fig. 4 shows the browser interface. The relevant semantic links will be displayed when the user points to the hyperlink mark. The user can further point to the semantic links to display the next level semantic links. The semantic link reasoning is carried out according to the linking rules to enable the user to foresee the terminal of the semantic links so that the user can select the proper path to carry out the next-step of browsing. Fig. 5 shows an example of displaying the reasoning resulting from browsing.

## 6. Application prospects and evaluation

The ADF is useful in forming active services for large-scale web documents (e.g. encyclopedias, textbooks, and software documents) or a large collection of inter-related web documents. The larger the document's scale, the better the application effectiveness of the ADF. Considering the cost of constructing an ADF, it is not effective to construct it for an isolated small Web document that does not have any positive impact on improving the efficacy of Web information services.

The ADF can be used to provide a complex question answering mechanism. If we regard a question as a document, then the question answering process is a reasoning process for finding a matching document, i.e. a process of chaining the related semantic links according to the reasoning rules. The proposed reasoning rules can provide more candidates for the matchmaker of the reasoning mechanism, e.g. the answer document can be extended to that document that provides the answer.

Many approaches for realizing the matching between two documents have been proposed. These are based on the fact that two documents about the same realm of knowledge will tend to use similar words. The ADF can be used to assist users in solving problems whose solutions exist in or are implied by a document fragment.

The ADF can also be used to provide online learning services. Compared to the existing HTML-based online teaching approaches, the advantages of an ADF-based teaching environment can actively guide students to read necessary materials and can answer students' questions according to teaching principles and the document content; it can provide the background knowledge of the document to the student during the learning process.

Current ontology markup languages provide the implementation basis for the ADF. The SHOE supports the expression of the Horn clause axioms, and the OIL supports the description logic. Comparing the ADF with the DAML + OIL, both of them agree that a fundamental component of the Semantic Web will be the markup of Web services to make them machine interpretable and use-apparent. The difference is that DAML + OIL intends to make the component into agent-ready resources so as to enable agents to find and use them easily, while the ADF will encapsulate the active behavior of the agent mechanism into the document component to provide active and intelligent services for web users. The rationale is that only a domain-specific agent can truly understand and manipulate domain resources.

## 7. Conclusion

The active document framework has three major advantages. *Fir*st, the representation of the document content incorporates not only the ontology and structural knowledge but also the background knowledge and semantic links, so it can reflect the document's content more accurately and completely than the traditional document representation approaches. This provides the basis for enhancing the preciseness of information retrieval services and enabling more intelligent services. *Second*, the operations are encapsulated in the document, and this provides an advantage similar to that of the encapsulation feature of object-oriented technology. Document authors or knowledge engineers incorporate the operations according to the document content; obviously, the document author best knows its content and meaning. Ordinary users do not need to know much more than the material they want. *Third*, the client side service provider does not need to search for the information from large-scale web information resources based on a word-level match in the way required by the current information retrieval approaches; the provider just needs to subscribe to a relevant indexing mechanism (e.g. a central library), which can suggest the active document that provides the needed information services like a soft-device [27]. Users need only provide their requirements (problems or questions). This can raise efficiency and mobility of Web information services. Tools for making the semantic links and the intelligent browser have been developed. Applications in cooperative research and online cooperative learning have shown that the proposed approach is applicable.

## Acknowledgements

The author thanks the editor-in-chief for his valuable edition of this paper. Thanks also go to all members of China Knowledge Grid research group for their diligent work for system implementation, especially Ruixiang Jia, Weiyu Guo, Lianhong Ding, Jia Bi, Peng Shi, Liping Zheng, Yunpeng Xing, Xue Chen and Xiang Li. This work was supported by the National Science Foundation of China.

## References

[1] J. Allan, Building hypertext using information retrieval, Information Processing and Management 33 (2), 1997, pp. 145–159.

[2] R. Beckwith, C. Fellbaum, D. Gross, G. Miller, in: U. Zernik (Ed.), WordNet: A Lexical Database Organized on Psycholinguistic Principles, Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon, Lawrence Erlbaum, London, 1991, pp. 211–231.

[3] J. Broekstra, et al., Enabling Knowledge Representation on the Web by Extending RDF Schema, in: Proceedings of the 10th International WWW Conference, May 2001, Hong Kong (http://www.cs.vu.nl/~frankh/abstracts/www01.html).

[4] S. Decker, et al., The Semantic Web: The Roles of XML and RDF, IEEE Internet Computing, September–October 2000, pp. 63–74.

[5] R. Dieng, Knowledge Management and the Internet, IEEE Intelligent Systems, May–June 2000, pp. 14–17.

[6] D. Fensel, et al., OIL: An Ontology Infrastructure for the Semantic Web, IEEE Intelligent Systems, vol. 16, issue 2, March–April 2001, pp. 38–45.

[7] S.J. Green, Building hypertext links by computing semantic similarity, IEEE Trans. Knowledge Data Eng. 11 (5), 1999, pp. 713–730.

[8] J. Heflin, J. Hendler, A Portrait of the Semantic Web in Action, IEEE Intelligent Systems, vol. 16, issue 2, March–April 2001, pp. 54–59.

[9] J. Hendler, D. McGuinness, The DARPA Agent Markup Language, IEEE Intelligent Systems, vol. 15, issue 6, November–December 2000, pp. 72–73.

[10] J. Hendler, Agents and the Semantic Web, IEEE Intelligent Systems, vol. 16, issue 2, March–April 2001, pp. 30–37.

[11] M.R. Henzinger, Hyperlink Analysis for the Web, IEEE Internet Computing, January–Feburary 2001, pp. 45–50.

[12] M. Klein, XML, RDF, and Relatives, IEEE Internet Computing, March–April 2001, pp. 26–28.

[13] F. Leymann, D. Roller, Workflow-based applications, IBM Syst. J. 36 (1), 1997, pp. 102–122.

[14] A. Maedche, S. Staab, A Ontology Learning for the Semantic Web, IEEE Intelligent Systems, March–April 2001, pp. 72–79.

[15] G. Marchionini, S. Dwiggins, A. Katz, X. Lin, Information roles of domain and search expertise, Libr. Inf. Sci. Res. 15 (1), 1990, pp. 391–407.

[16] S.A. McHraith, T.C. Son, H. Zeng, Semantic Web Services, IEEE Intelligent Systems, March–April 2001, pp. 46–53.

[17] P. Martin, P.W. Eklund, Knowledge Retrieval and the World Wide Web, IEEE Intelligent Systems, May–June 2000, pp. 18–25.

[18] R. Rada, D. Diaper, in: H. Brown (Ed.), Converting Text to Hypertext and Vice Versa, Hypermedia/Hypertext and Object-Oriented Databases, Chapman & Hall, London, 1991, pp. 167–200 (Chapter 9).

[19] F.E. Ritter, G.D. Baxter, G. Jones, R.M. Young, Supporting cognitive models as users, ACM Trans. Comp. Hum. Interact. 7 (2), 2000, pp. 141–173.

[20] R.K. Srihari, Z. Zhang, A. Rao, Intelligent indexing and semantic retrieval of multimodel documents, Inf. Retrieval 2, 2000, pp. 245–275.

[21] P. Thistlewaite, Automatic construction and management of large open webs, Inf. Process. Manage. 33 (2), 1997, pp. 145–159.

[22] D. Tudhope, C. Taylor, Navigation via similarity: automatic linking based on semantic closeness, Inf. Process. Manage. 33 (2), 1997, pp. 233–242.

[23] WfMC, The Workflow Reference Model (http://www.wfmc.org/).

[24] H. Zhuge, X. Shi, J. Ma, Abstraction and analogy in cognitive space: a software process model, Inf. Software Technol. 39, 1997, pp. 463–468.

[25] H. Zhuge, Inheritance rules for flexible model retrieval, Decis. Support Syst. 22 (4), 1998, pp. 383–394.

[26] H. Zhuge, T.Y. Cheung, H.K. Pung, A timed workflow process model, J. Syst. Software 55, 2000, pp. 231–243.

[27] H. Zhuge, Clustering Soft-device in the Semantic Grid, IEEE Computing in Science and Engineering, November–December 2002, pp. 60–62.

[28] H. Zhuge, Active Document Framework ADF: Concept and Method, in: Proceedings of the 5th Asia Pacific Web Conference, Xian, China, April 23–25 2003.

[29] H. Zhuge, Workflow-based cognitive flow management for distributed team cooperation, Inf. Manage. 40 (5), 2003, pp. 419–429.

**Hai Zhuge** is a professor at the Institute of Computing Technology, Chinese Academy of Sciences. He is serving on the editorial boards of a number of international journals such as *Information and Management*, *Future Generation Computer Systems*, and *Journal of Systems and Software*. He is the leader of the China Knowledge Grid project (http://kg.ict.ac.cn). His current research interests include: semantic web, knowledge grid, knowledge management, problem-oriented model base systems, component reuse, cognitive-based software process model, interoperation model for group decision, and web-based workflow model. He is the author of one book and over 50 papers appeared mainly in leading international conferences and the following international journals: *Communications of the ACM*, *IEEE Intelligent Systems*, *IEEE Computing in Science and Engineering*, *IEEE Transactions on Systems*, *Man*, *and Cybernetics*; *Information and Management*; *Decision Support Systems*; *Journal of Systems and Software*; *International Journal of Cooperative Information Systems*; *Expert Systems with Applications*, *Knowledge-based Systems*; *Information and Software Technology*; *and*, *Lecture Notes in Computer Science*.