

Semantic Web 与基于语义的网络信息检索

张晓林

(四川大学信息管理系, 成都 610064)

摘要 本文描述网络环境语义检索的需求, 分析 Semantic Web 的组成框架, 探讨概念集(ontologies)及其定义和标记语言, 并介绍基于概念集的信息资源语义标注和语义推理基本过程。

关键词 Semantic Web 语义检索 网络信息检索 概念集 概念集标记语言 语义标注 语义推理

Semantic Web and Semantic-based Networked Information Retrieval

Zhang Xiaolin

(Department of Information Management, Sichuan University, Chengdu 610064)

Abstract The paper describes the need for networked semantic search, analyses the framework of the Semantic Web, explores the structure ontologies and their definition and markup languages, explains the process of ontology-based annotation and inference.

Keywords Semantic Web, semantic search, networked information retrieval, ontology, ontology markup languages, ontological annotation, ontological inference.

1 基于逻辑语义关系的网络化检索

目前, 网络信息资源检索主要利用以下方法:

(1)以搜索引擎为代表的基于关键词的检索;

(2)以主题网关为代表的以元数据为基础的检索, 例如 AHDS、EELS、MathGuide 等^[1];

(3)基于数据库及相应数据模式和检索语言的深层网络资源(Deep-Web)检索;

(4)正在发展的利用 XPath 路径语言^[2]和 XQuery 检索语言^[3]的基于文献内容结构的检索。

这些方法或者要求了解检索对象数据结构或元数据格式、基于语法结构进行检索, 或者不能处理复杂语义关系、常检索出大量相关度很差的文献。实际上, 人们在检索时经常面对复杂的难以事先了解的资源类型及内容结构, 但又希望能针对复杂概念

进行准确的语义检索, 这就要求检索系统能解析各种内容结构的信息资源及其相互之间的复杂概念关系。例如, 我们欲检索“熟悉 XML 的专家”, 这个看似简单的检索语句实际上涉及复杂的逻辑概念、语义和语法关系, 如:

(1)XML 指 Extensible Markup Language(扩展标记语言)^[4], 属于数据标记语言, 可看成 SGML 的简化版本, 本身又涉及 XML Schema、XSL、XSLT、XPath、XLink、XQuery 等标准和技术;

(2)“专家”指具备特定专业知识的人, 例如写过相关文献、讲授过相关课程、或参与相关研究项目的人员, 甚至可是参加过有关课程或会议的其他人员(权威性较低而已);

(3)“文献”可能是标准、著作、文章、报告、网页、辅导材料等, “课程”可能包括正式课程、讲座、报告会、培训辅导等, “研究项目”可能是立项课题或者自

拟的课题、系统设计、实验、理论分析、文献综述活动等,“人员”则可能是具各种可能职衔甚至没有特定职衔的人。

(4) 这些信息可能存在于关于文献、文献集合、人员、机构、会议、活动、研究项目、系统、经费计划、专门公告、电子邮件等资源里;

(5) 这些资源可能具有不同内容结构、不同标记元素和元数据元素、不同自然语言,例如作者可能表示为 Author、Contributor、Creator、Editor、作者、编者、编撰者及其他名称。

根据上述语义关系分析下例网页,可知 section 是 chapter 的一部分,chapter 是图书一部分,图书是文献的一种形式,Contributor 代表作者,XSLT 是 XML 技术的一部分,因此 John Smith 可判为是“熟悉 XML 的专家”。

```
< chapter>
  < section>
    < title> Introduction to XSTL< /title>
    < subject> XSLT< /subject>
    < contributor> John Smith< /contributor>
    < date> 2000.05.05< /date>
    < xlink ref = " http: //example. com /books /
chapters /chapter 011. html" >
  < /section>
< /chapter>
```

同样,对下一网页,语义关系指出履历表包含所从事课题、课题涉及主题、课题可有合作者,如果主题包括 XML 或其部分,则履历表主及在该项目上的合作者可作为“熟悉 XML 的专家”。

```
< resumes>
  < people>
    < person>
      < name> Peter Johnson< /name>
      < xlink ref = " http: //example. edu /computers /faculty /JohsonP. html" >
      < projects>
        < project>
          < title> Metadata Representation with XML Schema< /title>
          < topics>
            < topic> Metadata< /topic>
            < topic> XML Schema< /topic>
          < /topics>
          < duration> 1999.10—2001.10< /duration>
          < coolaborated-with>
            < name> Mary Wedgeworth< /name>
            < xlink ref = " http: //companyA. com /labs /wedgeworth. html" >
          < /coolaborated-with>
        < /project>
        .....
      < /projects>
      .....
    < /person>
    .....
  < /people>
< /resumes>
```

基于语义关系,检索系统应能自动理解这些不同网页的内容结构及内容元素、能理解或推理这些

内容元素之间的逻辑语义关系,从而将 Smith、Johnson 和 Wedgeworth 都检索出来。

2 语义检索与 Semantic Web

要能做到理解网络信息资源中丰富的逻辑语义并进行推理检索,检索系统必须具备:

- (1)一定的知识体系来表达概念对象及其相互间的逻辑语义关系;
- (2)一定的词汇体系来描述这些对象类及其关系,建立对应的元数据元素;
- (3)一定的赋值机制来建立元数据元素与对应资源(或资源片段)的描述关系;
- (4)一定的标记语言和语法来对元数据元素及其赋值关系进行标记;
- (5)一定的检索推理机制为利用知识体系和标记语言进行搜索、验证和推理。

如果上述这些机制都能以计算机可理解可处理的方式建立起来并在网络资源系统中普及,就能实现网络环境下基于语义的检索和推理,这就是 Semantic Web 的目标。

实际上 Web 创始人 Tim Berners-Lee 在 1998 年就提出了 Semantic Web 的概念^[3],随着 W3C 在 2001 年 2 月正式推出 Semantic Web Activity^[6],网络环境下语义检索进入网络研究发展的主流。

根据 W3C, Semantic Web 研究活动的目标是开发一系列计算机可理解和处理的表达语义信息的语言和技术,以支持网络环境下广泛有效的自动推理。在目前阶段,它将主要致力于支持对网络信息资源及其内容的语义和语义关系表征,支持代理系统基于语义对数据的自动分析、理解和处理,支持代理系统相互间基于语义的知识交换,支持不同应用领域和系统间基于语义对数据进行自动的交换、转换和复用。

为了实现这个目标, Semantic Web 需要一个多层技术框架(图 1)^[7],其中:

(1)逻辑层(logical Layer),即前述逻辑知识体系,实际上,它们多局限于一定应用领域,称为概念集(Ontology)。Semantic Web 在该层次的任务是建立定义和标记概念集的标准方式。

(2)语义层(Semantic Layer),即前述词汇体系,用以表达元数据。实际应用中,不同应用领域会根据不同目的、针对不同概念对象建立多种描述性元数据模式^[8],对具体对象及其属性进行描述,例如 DC、MARC、EAD、GILS、MPEG7 等。Semantic Web 的任务是建立基于概念集来描述元数据元素、元数据

关系和约束元数据语义的机制。有时,概念集可直接定义元数据,或者将某些元数据模式引入到概念集中。

(3)赋值层(Assertion Layer),即前述赋值机制,以标准方式建立元数据与被描述资源的连接,从而保证计算机能明确地确认元数据、元数据含义及其与资源的关系。目前, RDF 正成为网络环境下的基本赋值机制。

(4)语法层(Syntax Layer),即前述标记语言标准,以计算机可理解和处理的方式对上述三个层次进行标记。目前, XML 作为网络资源的标准标记语言,也正成为语义和知识的基础标记语言。

在逻辑层之上,可能还需要利用概念集和其他描述性元数据进行语义分析的推理机制,基于数字签名方式对概念集、元数据、赋值关系、甚至推理机制等进行确认和验证的机制,从而建立可信赖的语义环境和推理环境。另外在语法层下,还需要数据编码和资源标识的标准,如 Unicode 和 URI。

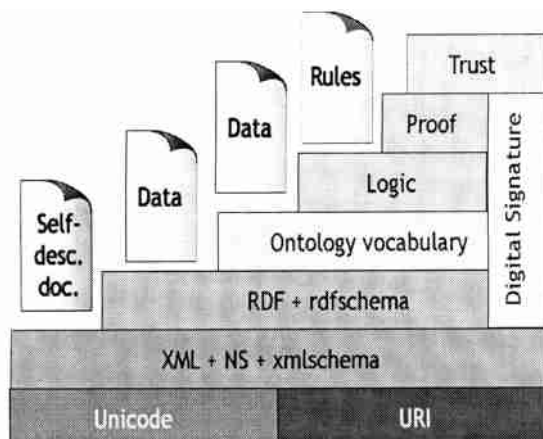


图 1

3 概念集

所谓概念集(Ontology, 或称应用知识体系),指特定应用领域公认的关于该领域的对象(包括实际对象和逻辑对象)及对象关系的概念化表述(请与表示“本体论”的同一词汇区别)。一般地,概念集包含以下组成部分^[9~10]:

(1)本领域对象类的层级体系,即不同对象类间存在 is-a-kind-of、part-of 等关系,通过这些层级关系从而构成整个领域的对象类体系。父子类之间往往存在继承关系。类别间还可能存在组合关系,例如某类是其他若干类的交集或并集,或是另一类的等同集、反集或补集。

(2)对象类的属性及属性取值限制, 这些属性由具体对象类决定, 可按照层级关系继承, 具有取值对象类和实际值范围、取值基数的限制, 并具有关于交换性、对称性、唯一性、可传递性、有序性等的规定。某些属性还可以是其他属性的子属性。

(3)语义关系体系, 即对象类之间的逻辑相互关系, 例如 caused-by、used-by、interact-with、collaborate-with、supervised-by、written-by 等。某些逻辑关系只能存在于特定对象类之间, 例如 written-by 关系就只能存在于文献与人(或代理系统)之间。

(4)关于对象类及语义关系的推理规则, 例如 IF-THEN 规则, 全称量化规则和存在量化规则, 属性选择性继承规则, 对象类组合规则, 逻辑关系推理规则等。这些规则往往用一阶或高阶谓词逻辑等形式表示。

概念集通过一个语言定义和标记后成为显性知识, 可用以在信息系统中表征和交换知识, 并可被知识代理(agents)自动解析和识别。但是, 在任何领域中都可能用到多个概念集, 每个概念集都可能涉及多个对象及相应的元数据模式。例如, 在高等教育领域, 需要关于学校结构、关于教学科研活动、关于教育资源等的多个概念集, 这些概念集都将涉及文献、人员、课程、项目、机构、设施等的描述性元数据模式。由于概念集之间在逻辑上和应用上的相互联系, 人们可通过概念继承关系、语义关系和引用关系来复用有关定义和标记, 从而提高概念集和元数据定义与标记的效率。这方面可以采取的措施包括:

(1)建立概念集层级体系, 下层概念集直接继承上层概念集的概念类。实际上, 下层概念集的概念类是上层概念集某个概念类的子类, 或者是存在一定属性限制的子类。有人提出^[9], 可建立如图 2 所示的概念集体系, 其中顶层概念集定义最基本的概念类、属性及语义关系, 例如时间、空间、物质、对象、事件、行为等; 领域概念集和任务概念集利用顶层概念集来细化定义具体应用领域(例如教育、医学等)或具体通用任务(如交易、传输等)的专用概念类、属性及语义关系; 而应用概念集则利用领域和任务概念集来进一步定义针对某个具体应用的概念集(例如图书交易概念集)。

(2)建立概念集引进机制, 某概念集可直接引进另一概念集, 将其嵌入到自己概念体系的合适位置。在引进中, 被引进概念集中某些概念类的定义或属性可能被限制、被重命名、或与当前概念集中某个类组合成新类。

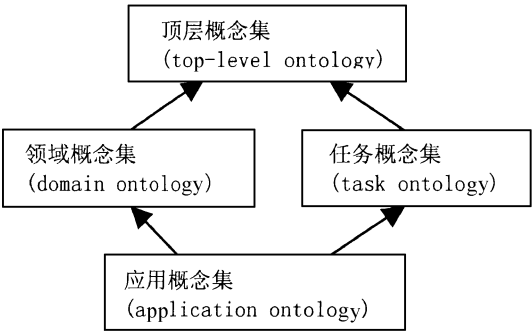


图 2

(3)建立元数据引入机制, 某概念集可直接引入某个描述性元数据模式及其定义作为自己的概念类, 或引入元数据模式作为特定概念类的属性集, 例如引入 Dublin Core 作为文献概念集, 或作为科学研究概念集中文献类的属性。

4 概念集定义与标记

在网络环境下利用概念集, 要用计算机可识别的语方来定义概念集。在人工智能和知识工程领域, 人们开发了 KIF (Knowledge Interchange Format)^[11]、CycL^[12]、Ontolingua^[13] 等表示语言。当前趋势是利用 XML/RDF/HTML 来标记, 有 DAML (DARPA Agent Markup Language)^[14]、OIL (Ontology Interchange Language)^[15]、SHOE (Simple HTML Ontology Extensions)^[16]、以及 XOL (XML-based Ontology Exchange Language)^[17]、OML/CKML (Ontology Markup Language/Conceptual Knowledge Markup Language)^[18], 现在被 Information Flow Framework Language^[19] 吸收)等。

概念集定义语言一般基于通用逻辑语言(例如一阶谓词逻辑), 至少提供以下语言工具:

- a. 概念集语言的版本、名称域、引入概念集、本身描述性元数据等的定义语句和语法形式;
- b. 概念类的定义语句和语法形式;
- c. 概念类属性的定义语句和语法形式;
- d. 概念集顶层概念类及其属性(下层概念集可用这些类来定义它们的概念类及属性);
- e. 概念集文件的封装方式。

利用这些语言工具, 可以用计算机可识别和理解的语言定义和标记概念集。

例如, OIL^[15] 将概念集定义为概念集封装层 (Ontology Container) 和概念集定义 (Ontology-definition) 两部分, 其中封装层直接引用 Dublin Core 的 15 个元素来描述概念集本身的作者、发表时间、覆盖领

域、版权声明等, 而概念集定义部分包含类定义、属性(OIL 称之为 slot)定义、定理(axioms)定义及实例定义(Instance-of), 如下表所示:

类定义 Class-def Subclass-of Slot-constraint Value type Has value Has filler Cardinality Max cardinality Min cardinality	属性定义 Slot-def Subslot-of Domain Range Inverse Properties Transitive Symmetric Functional	定理定义 Disjoint(c1 c2 ... cn) Covered c1 by c2 ... cn Disjoint-covered c1 by c2 ... cn Equivalent (c1 c2)
--	---	---

又如, DAML^[14] 定义每个类为:

class: := (sub Class Of *| disjoint With *| disjoint Union Of *| same Class As *| equivalent To| boolean Combination *| enumeration Element *)+

其中, *表示零次或多次、+表示一次或多次, 每个元素值都是用一定方式表示的类(参考类), disjoint With 表示被定义类与参考类互不交并, disjoint Union Of 表示被定义类是若干参考类的非交并集, boolean Combination 表示被定义类是若干参考类按指定布尔符号规定的集合(交集、并集等), enumeration Element 表示被定义类对应于指定的实例集合。

在上述逻辑化定义基础上, 人们需要用一定标准标记语言来标记有关定义, 并用这些定义中的概念或属性名作为标记符(Tags)标注具体资源文件, 从而保障计算机在任何情况下能明确地理解哪个概念集的什么概念按照哪些属性描述什么资源中哪部分。具体的标记方法主要有:

(1)扩展 HTML 语言来进行标记, 以 SHOE^[19] 为代表, 它在 HTML 上增加了一系列标记符, 利用它们 SHOE 可将概念集标记为 HTML 文件, 也可用概念集元素去标注资源, 所形成的元数据也为 HTML 文件:

<ONTOLOGY></ONTOLOGY>
<DEF-CATEGORY>
<DEF-RELATION></DEF-RELATION>
<DEF-INFERENCE></DEF-INFERENCE>
<INF-IF></INF-IF>
<INF-THEN></INF-THEN>
<CATEGORY>
<RELATION></RELATION>
<USE-ONTOLOGY>
<DEF-RENAME>
<INSTANCE></INSTANCE>

概念集文件定义符
概念类定义符
概念类关系定义符
推理规则定义符及其中的条件语句和结论语句
推理规则条件语句定义符
推理规则结论语句义符
概念类标记符
语义关系标记符
引入概念集标记符
引入概念类或属性元素重命名标记符
概念类实例标记符

(2)用 XML/RDF 语言标记, 如 DAML、OIL、XOL 等。RDF(资源描述框架, Resource Description Framework)^[20] 正成为用概念集和描述性元数据描述网络资源的基本赋值机制。RDF 将元数据作为关于特定资源的特定属性的取值声明, 是由资源、属性、属性值构成的三元关系模式, 例如“网页 ABC.com/XYZ/abc.html(资源)的制作者(属性)是 John Smith(属性

取值)”。不同的概念集或描述性元数据都只是这种三元关系模式的具体体现而已。鉴于此, RDF 定义了用 XML 语言描述这种三元关系的基准方式, 建立了元数据交换的基础。

例如, “Ora Lassia 是 <http://www.w3.org/Home/PubText/>的作者”可表示为:

< rdf: Description about = ” <http://www.w3.org/>

Home PubText />

<dc:Creator> Ora Lassila< /dc:Creator>

< /rdf:Description>

其中, rdf:Description about 通过 URI 指出所描述的资源, dc:Creator 标记资源的特定属性并可用名称域来链接该属性定义信息, 而属性标记符之间的值就是属性取值。对于更复杂情况, 例如属性值本身也是资源(例如该网站作者 Lassila 本身拥有邮件地址和主页)、属性本身有限定属性(例如来自什么主题词表或取什么重量单位)、元数据描述语句本身有自己的限定(例如谁做出的这个描述、该描述的可靠性)等, 都可用 RDF 三元语句来描述。由于 RDF 由 XML 标记, 因此用 RDF 标记的元数据可由任何基于 XML 的浏览器解析。

为描述元数据元素间的复杂语义关系, W3C 进一步定义了 RDFS (RDF Schema)^[21], 它可看成是一个概念集定义语言, 定义概念类体系以及类属性和类关系。RDFS 通过 sub Class Of 定义类层级关系、sub Property Of 定义属性层级关系、range 定义概念类允许的应用对象、domain 定义概念类允许的取值范围。但 RDFS 语言较简单, 尤其缺乏对复杂的类组合、类或属性限定、推理定理的定义工具, 难以直接作为可供复杂推理的概念集语言。但 RDFS 可作为顶层概念集, 提供某些顶层类和语言工具, 被引用到下层概念集定义语言中来定义它们的概念类(作为 RDFS 概念类的子类)和语言工具, 例如下面的 DAML 范例就用 RDF 和 RDFS 定义了“Thing”和“Nothing”两个类:

```
<Class rdf:ID="Thing">
  < rdfs: label> Thing< /rdfs: label>
  < rdfs: comment> The most general (abstract)
class in DAML.< /rdfs: comment>
  < union Of rdf: parse Type="daml: collection">
    < rdfs: Class rdf: about=" #Nothing" >
    < rdfs: Class>< complement Of rdf: resource
=" #Nothing" >< /rdfs: Class>
  < /union Of>
< /Class>
<Class rdf:ID="Nothing">
  < rdfs: label> Nothing< /rdfs: label>
  < rdfs: comment> the class with no things in it.
< /rdfs: comment>
  < complement Of rdf: resource=" #Thing" >
< /Class>
```

5 Semantic Web 应用环境与过程

在实际应用中, 我们需要一系列相互关联的工具、系统和资源来利用体现在概念集和其他元数据中的语义信息进行智能检索和推理, 它们所构成的体系^[22~23]可由图 3 表示:

在这个体系中, 概念集构建工具利用有关定义标记语言、上层概念集和其他概念集来定义和标记特定领域、任务或应用概念集; 网页标注工具利用资源搜索器或其他方式采集网页, 利用概念集对网页

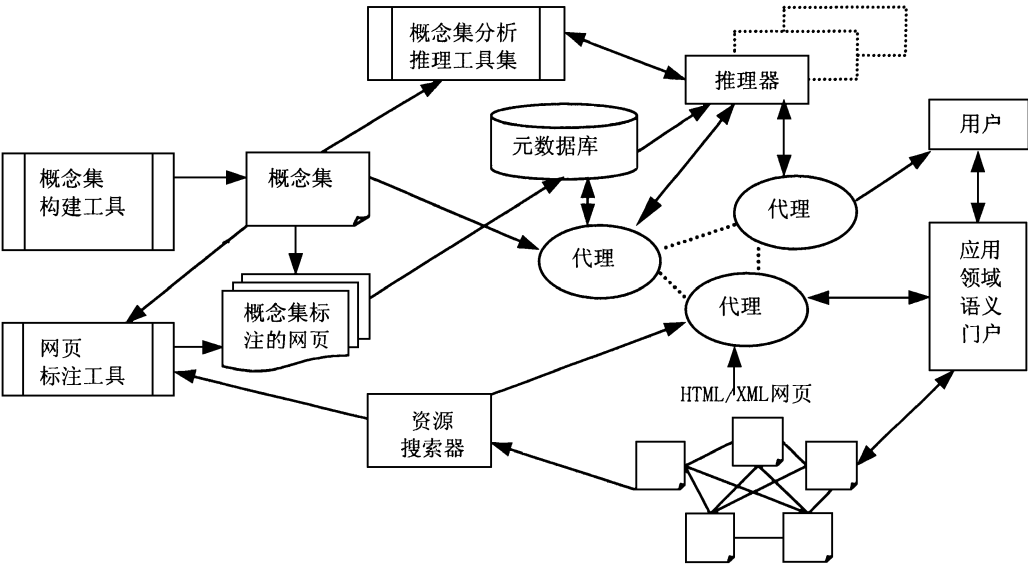


图 3

进行语义标注,标注所形成的元数据存入元数据库,某些元数据也可以嵌入网页中;代理系统利用概念集、元数据进行语义推理,实现语义检索和其他智能处理;概念集分析推理工具集主要在元数据涉及多个概念集时辅助推理器分析推断概念关系和处理规则,支持分布式概念集引用和复用;应用领域语义门户则利用对应的概念集支持用户界面、辅助检索、支持基于语义的资源再组织和动态组织。整个过程中比较复杂的是语义标注和语义推理,我们下面进一步说明。

语义标注是根据有关概念集为网页(包括网络上其他格式的资源)及其各个部分标注概念类、概念属性和其他元数据的过程,是语义推理的基础。语义标注的方法目前主要有三类^[22, 24~25]:

(1)人工标注,由专门人员确定网页的适用概念集、解析网页内容结构、选择元数据元素、建立用 RDF 或 HTML 语言标记的语义数据。这个过程往往在一定编辑器、概念集和标注过程知识库支持下进行,显然是一个智力密集和劳动密集过程,难以应付浩瀚和不断变化的网络资源。

(2)利用 DTD(文献类型定义)和 Schema(文献模式)进行概念集映射和标注。由于 SGML/XML 文献的 DTD 或 Schema 详细定义文献内容结构和各内容

元素,我们可以建立特定概念集和特定 DTD Schema 之间的映射关系,从而自动地将 SGML/XML 文献中的 DTD Schema 内容元素标记转换为对应的概念集元数据标记。但由于 DTD Schema 的适用领域及其内容结构和概念集的适用领域及其体系结构往往不协调不兼容,因此难以准确映射,还需要人工进行审查和修改。

(3)利用词汇语义分析进行标注。自动词汇抽取和分析的技术已较成熟,可在此基础上建立词汇集合与概念类别之间的映射关系,然后通过自动词汇分析找出文献或文献片段的类别、甚至与其他类别的语义关系,利用这些概念类别进行标注。实际分析中,从文献抽取出它的词汇矢量,分析该矢量与概念集中概念类别的匹配程度,选择匹配程度最高的一个或若干概念类别作为这个文献的概念类别。当然,这种标注可能要限制在一定应用领域和资源类型,而且也需要人工审查。

语义推理利用文献的语义标注和概念集语义关系及推理规则进行推理,从而实现智能检索和知识组织。例如,对于本文第一部分所举例子,在相应的文献概念集、科研概念集、XML 技术体系概念集支持下,利用下列列举定义和推理规则就可实现智能检索:

section SUBCLASS-OF chapter SUBCLASS-OF book SUBCLASS-OF publication SUBCLASS-OF document
document HAS-SUBJECT(x, y)

WRITTEN-BY(x, y)

author EQUIVALENT-TO contributor

project SUBCLASS-OF research SUBCLASS-OF work

work HAS-SUBJECT(x, y)

PARTICIPATED-BY(x, y)

Forall x, y, z

(written-by(y, x) and has-subject(y, z)) \Rightarrow has-knowledge(x, z)

Forall x, y, z

(participated-by(y, x) and has-subject(y, z)) \Rightarrow has-knowledge(x, z)

Forall x, y, z

(collaborated-with(x, y) and has-knowledge(y, z)) \Rightarrow has-knowledge(x, z)

在此基础上,检索系统还可自动抽取文件中的相关具体内容、或者根据特定内容将相关文件或其中内容组织到一定知识体系中、或者合并文件内容来建立新的集成文件。

参 考 文 献

- 1 Subject Based Information Gateways <http://www.lub.lu.se/desire/sbigs.html>

2 XML Path Language (XPath). Version 1. 0. W3C Recommendation 16 November 1999 <http://www.w3.org/TR/xpath>

3 XQuery: A Query Language for XML. W3C Working Draft 15 February 2001. <http://www.w3.org/TR/xquery/>

4 Extensible Markup Language <http://www.w3.org/XML/>

5 Tim Berners-Lee, Semantic Web Road map, 1998. 10 [http://www.w3.org/Design Issues/Semantic.html](http://www.w3.org/Design%20Issues/Semantic.html)

6 W3C Semantic Web Activity Statement <http://www.w3.org/2001/se/Activity>

7 The Semantic Toolbox; Building Semantics on top of XML-RDF [http://www.w3.org/Design Issues/Toolbox.html](http://www.w3.org/Design%20Issues/Toolbox.html)

8 张晓林. 元数据开发应用的标准化框架. 现代图书情报技术, 2001. 3

9 R. Studer, et al. Knowledge Engineering: Principles and Methods.

10 N. Guarino. Formal Ontology and Information Systems. In Formal Ontology in Information Systems, Proceedings of FOIS' 98, Jun, 1998

11 Knowledge Interchange Format. draft proposed American National Standard (dpANS). NCITS. T2/98-00. <http://logic.stanford.edu/kif/dpans.html>

12 Features of CycL <http://www.cyc.com/cycl.html>

13 T. R. Gruber. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, V. 5, p. 199~200, 1993

14 DARPA Agent Markup Language (DAML) <http://www.daml.org/>

15 Ontology Interchange Language (Ontology Inference Layer) <http://www.ontoknowledge.org/oil/>

16 S. Luke and J. Heflin. SHOE 1. 01. Proposed Specification, April 28, 2000 <http://www.cs.umd.edu/projects/plus/SHOE/spec.html>

17 P. D. Karp, et al. XOL: An XML-Based Ontology Exchange Language. July 3, 1999 <http://www.oasis-open.org/cover/xol-03.html>

18 Ontology Markup Language/Conceptual Knowledge Markup Language (OML/CKML) <http://www.oasisopen.org/cover/oml.html>

19 Information Flow Framework Language (IFF) <http://xml.cov-empags.org/iff.html>

20 Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999 <http://www.w3.org/TR/REC-rdf-syntax/>

21 Resource Description Framework (RDF) Schema Specification 1. 0. W3C Candidate Recommendation 27 March 2000 <http://www.w3.org/TR/rdf-schema/>

22 S. Decker, et al. An Information Food Chain for Advanced Applications on the WWW (2000). <http://citeseer.nj.nec.com/decker00information.html>

23 Onto Broker Home Page <http://ontobroker.aifb.uni-karlsruhe.de/>

24 M. Erdmann, et al. From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools (2000). <http://citeseer.nj.nec.com/erdmann00from.html>

25 M. Erdmann, and R. Studer. Ontologies as Conceptual Models for XML Documents (2000) <http://wem.ualgary.ca/KS1/KAW/KAW99/papers/Erdmann/erdmann.pdf>

(责任编辑 马 兰)