

基于语义 Web 的知识处理流程及其技术框架研究

朝乐门

摘 要 在对基于语义 Web 进行知识处理的必要性进行分析的基础上,研究基于语义 Web 的知识处理流程和基于语义 Web 的知识处理技术框架。首先,从语义 Web 与传统 Web 的区别入手,分别探讨语义 Web 中知识表示、联网、发现、获取、集成的基本特征和常用技术,对人脑和电脑进行合理分工和有效定位,充分体现人机在知识处理流程中的互补作用;其次,结合基于语义 Web 的知识处理流程,从多层次上集成知识处理新技术,为人机结合处理知识提供技术方案。图 2。表 1。参考文献 22。

关键词 语义Web 知识处理 技术框架

分类号 TP274

ABSTRACT Semantic Web not only makes it easier for Artificial Intelligence to process knowledge, but also provides a human-machine cooperative solution to make humans and machines complement each other perfectly when processing knowledge. This paper describes the workflow of knowledge processing on the semantic web, and demonstrates the basic features and common technologies in representing, interconnecting, discovering, capturing and integrating semantic web knowledge. In addition, it proposes a framework for knowledge processing on the semantic web, which integrates a variety of knowledge processing technologies. 2 figs. 1 tab. 22 refs.

KEY WORDS Semantic web. Knowledge processing. Technology framework.

CLASS NUMBER TP274

知识处理(Knowledge Processing)和知识管理(Knowledge Management)是两个不同层次的概念。知识处理是改变知识生命期的系列活动,而知识管理则是如何改善知识处理活动来实现管理目标的管理学科^[1]。语义 Web(Semantic Web)是传统 Web 的一种延伸,其所有数据均被赋予规范的含义,以便于人机合作完成数据处理任务^[2]。因此,语义 Web 的出现对知识处理活动产生了直接影响。本文主题是基于语义 Web 的知识处理的流程及其技术框架研究。

1 基于语义 Web 进行知识处理的必要性

1.1 人工智能技术的发展不能满足计算机知识处理的需要

人工智能是计算机知识处理的核心技术。虽

然人工智能领域的学者在知识的获取、表示、更新和推理,大型知识库的组织与维护,新一代逻辑处理机制等方面进行了大量的研究,但是该领域的现有研究成果仍不能满足计算机知识处理的需要。人工智能技术在知识处理中的瓶颈,缘于传统 Web 中知识表示具有复杂性。1999 年, Tim Berners-Lee 等人提出了语义 Web 的理念,旨在通过改变传统 Web 的知识表示方法来降低计算机知识处理的难度,进而解决二者之间的矛盾^[3]。

1.2 解决人脑与电脑的关系是知识处理的重要课题

随着信息技术的广泛应用,计算机已成为知识产生、处理、传递和存储的重要工具之一。信息爆炸式增长更加凸现了人脑和电脑在知识处理过程中的不同特点。电脑擅长于处理“物理”与“事理”,不善于处理“人理”;而人脑擅长于处理“人

理”,也能处理“物理”与“事理”,但效率不及电脑^[4]。因此,如何最大程度地发挥人脑和电脑的优势,并且使二者处于最优协同状态是知识处理领域亟待解决的问题之一。语义 Web 的出现能够对人脑和电脑在知识处理流程中的角色进行合理定位,使人机协同知识处理成为现实。

2 基于语义 Web 的知识处理流程

从流程视角看,语义 Web 与传统 Web 的区

别决定了基于语义 Web 的知识处理流程的特殊性。基于语义 Web 的知识处理流程可以分为五个步骤:知识表示、知识联网、知识发现、知识获取和知识集成,如图 1 所示。基于语义 Web 的知识处理流程是人机协同完成的过程。根据人和计算机在知识处理过程中各自的优势,在语义 Web 中知识处理流程的前端,如知识表示和联网阶段,人起到主导作用,在知识的获取和集成阶段的活动则主要由计算机完成。

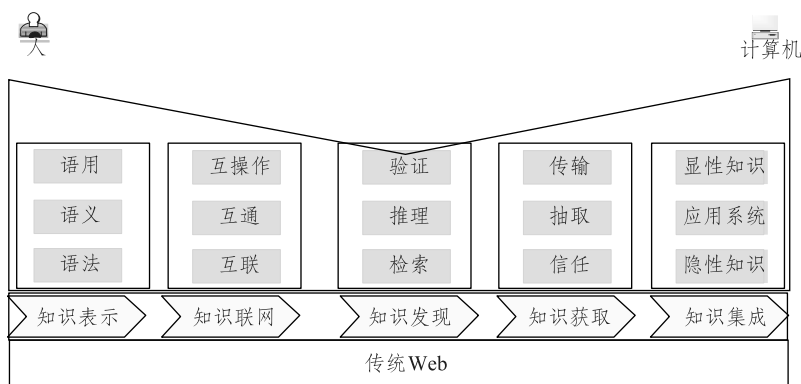


图 1 基于语义 Web 的知识处理流程模型

2.1 知识表示

语义 Web 与传统 Web 的第一个不同在于语义 Web 中知识表示的目的发生了重大变化。传统 Web 内容的表示方法是面向人类使用,重视人的可理解性而忽略了机器的可理解性。与传统 Web 不同的是,语义 Web 中知识表示以“机器可理解”为主要目的,同时兼顾知识的“人可理解性”^[5]。语义 Web 中的知识表示分为语法、语义和语用三个层次。

2.1.1 语法层次的知识表示

追根溯源,语义 Web 中知识表示的语法基础是 XML 语言,与 HTML 语言不同,XML 语言实现了知识内容及其显示方式的分离。XML 文档负责描述知识的内容和结构,而具体显示任务由 CSS(Cascading Style Sheets)或 XSL(Extensible Style Language)来实现。因此,XML 语言可以采用多种方式显示同一个内容,实现文档内

容和显示方式相互独立。

XML 语言实现了知识的“机器可读性(Machine Readable)”。XML 语言不仅具有很好的跨平台性,而且 XML Schema(或 DTD 技术)支持用户自定义 XML 文档的模式信息,如某一个元素出现的次数、内容类型、可用属性等,计算机根据这些模式信息,可自动判断 XML 文档的有效性。但由于缺少必要的语义信息,XML 语言本身并没有实现知识的“机器可理解性”,计算机只能判断 XML 文档在语法上的有效性,并不能理解其语义信息,更不能进行语义推理等智能操作。

2.1.2 语义层次的知识表示

XML 文档中缺少语义信息,主要原因在于 XML 语言本身并没有提供任何领域的知识建模方法,而 RDF(S)模型和 OWL 语言为此问题的解决提供了新的思路。

(1) RDF(S)模型。RDF(Resource Descrip-

tion Framework) 模型和 RDFS (RDF Schema) 模型提供了规范的描述知识的语义信息方法。在 RDF 模型中, 由“对象—属性—值”表示知识原子 (Knowledge Atom), 用 RDF Schema 描述知识原子的语义信息。RDF Schema 则支持类 (或属性) 间的继承机制和限定属性值的方法。

但 RDF 本身仅仅是一种语义模型, 不具备“计算机可读性”。因此, 又出现了基于 XML 的 RDF (S) 模型, 实现了 RDF (S) 知识描述的“计算机可读性”。基于 XML 的 RDF 文档采用计算机可读的 XML 元素和属性代替 RDF (S) 模型中的各种概念及其关系, 如采用 `<rdf:Description>` 元素标记一个知识原子; 通过修改 `<rdf:type>` 元素的 `rdf:resource` 属性为知识原子指定所对应 RDF Schema; 运用 `rdfs:subClassOf` 和 `rdfs:subPropertyOf` 属性表示继承机制; 通过 `rdfs:domain` 和 `rdfs:range` 属性可以对资源元素进行有效性限制。

但是, 基于 XML 语言的 RDF (S) 的语义表达能力和推理能力非常有限。RDF 局限于二元常谓词, 而 RDFS 则仅限于表示类 (或属性) 之间的继承关系和属性的定义域以及值域的限制, 不能表达更复杂的语义信息, 如类之间的交叉关系、逻辑运算等。

(2) OWL 语言。OWL 语言 (Ontology Web Language) 是改进基于 XML 的 RDF (S) 模型的语义表达能力和推理能力的方案之一。OWL 不但继续沿用了基于 XML 的 RDF (S) 语法, 并且以继承的方式重新定义了 RDFS 的部分元素, 同时还新增了诸多语义元素。如 OWL 语言的 `<owl:Class>` 类继承了 RDFS 的 `<rdfs:Class>`; 继承后的 `<owl:Class>` 类中可通过元素 `<owl:disjointWith>` 表示类之间的不可交叉性; 元素 `<owl:equivalentClass>` 表示类之间的等同关系; 元素 `<owl:Restriction>` 可以限定一个类; 元素 `<owl:complementOf>`、`<owl:unionOf>`、`<owl:intersectionOf>` 表示类之间的逻辑关系。

从语义表示能力来看, OWL 语言可以进一步分为 OWL Full、OWL DL 和 OWL Lite 三种子语言, 不同层次的 OWL 子语言对语义表示进行不同程度的限制, 以便于解决语义层次上知识

表示的丰富性和计算机处理复杂性之间的矛盾。为了方便进行知识表示过程, 软件开发者提供了支持 OWL (或 RDF) 的知识表示工具, 比较常用的有 Protégé、SemanticWorks、SWOOP、OntoStudio 等。这些语义知识表示工具的出现很大程度上减少了人们在知识表示过程中编写代码的工作量, 提高知识表示的效率。

2.1.3 语用层次的知识表示

在语义 Web 中, 多数语用层次的知识均以应用 (Application) 或服务 (Services) 的形式提供。因此面向服务的体系结构 (Service Oriented Architecture, SOA), 尤其是 XML Web Services 技术是语义 Web 语用层次知识表示的重要技术方案。与语义 Web 相似, Web Services 技术也采用 XML 语言作为语法基础。Web Services 技术采用 XML 语言统一描述、发现和集成 Web 服务^[6]。

虽然 Web Services 技术开发的最初目的并不是为了提供语义服务, 但通过 Web Services 技术可以将语用层次的知识封装成 Web 服务进行对外服务。因此, Web Services 技术为语义知识的语用层次表示提供了重要思路。近年来, 语义 Web 和 Web Services 技术的交叉应用——语义 Web 服务 (Semantic Web Services, SWS) 的研究越来越多。一些研究^[7-8] 较全面地总结了语义 Web 服务的发展现状及其趋势; 有的研究^[9] 则提出了 SESA (Semantically Enabled Service-oriented Architecture) 语义 Web 服务的框架结构在知识的语用层次表示方面具有较高的借鉴价值。

上述对语义 Web 中知识表示的探讨和图 1 所示, 语义 Web 知识表示具有如下特点: ①知识表示是语义 Web 中知识处理的基础, 语义 Web 知识表示的方式和质量直接影响后续知识处理工作的效率和效果; ②语义 Web 中知识表示的目的与传统 Web 知识表示的目的不同, 以“计算机可理解”为主要目的; ③语义 Web 中知识表示的过程是一种人机合作过程, 因此, 这种知识表示过程往往是一种隐性知识显性化的过程。

2.2 知识联网

语义 Web 与传统 Web 的第二个不同在于

前者是在数据层次上建立的“机器可理解的网
络”。传统 Web 是一种硬件设备或软件应用之
间的网络,数据被应用程序所控制,数据之间缺
少计算机可理解的联网关系。除了使知识具备
“计算机可理解性”,语义 Web 中知识表示的另
一个目的是在知识之间建立联网关系。语义
Web 环境下的知识联网可以分为互连、互通和
互操作三个层次。

2.2.1 互连

在语义 Web 中,互连可以分为两种,即物理
互连和逻辑互连。物理互连是指在知识供需双
方之间至少有一条在物理上连接的线路,为在
两者之间进行数据交换提供了物质基础;逻辑
互连是指在两个知识之间至少存在一种逻辑关
联。显然,物理互连与逻辑互连相互独立。

(1)命名空间(Namespace)与 URI(Universal
Resource Identifier)。命名空间和 URI 机制是语
义互联的重要手段。语义 Web 中声明命名空
间时,可以运用 URI 实现唯一标识知识资源的目

的。但是 URI 中并不一定要记录所指知识资源
的实际地址。因此,目前的命名空间和 URI 技
术水平有时只能实现知识之间的互联关系,不
能保证知识的互通性。

(2)逻辑互连。计算机可理解的数据联网
主要以逻辑层次上的互联关系为基础。语义
Web 中的逻辑互连可以进一步分为显性互联和
隐性互联。显性互联是以超链接等显性方式表
示的,识别并不需要语义逻辑推理;隐性互联一
般通过知识的结构和特定标记的方式间接体
现,识别和理解需要一定的逻辑推理。因此,逻
辑互连尤其是隐性逻辑互连是语义 Web 建设
的重要任务之一。当然,在逻辑互连过程中采
用了命名空间和 URI 技术。

以 OWL 语言为例,除了通过知识元素间的
嵌套关系来表达知识的语义关联之外,还可以
通过指定的元素、属性来建立互联关系,如表 1
所示。

表 1 OWL 中表示知识互联关系的方法

编号	互联关系	对应标记或属性
1	类之间的继承、等同、交叉关系	< rdfs:subClassOf > 、< owl:equivalentClass > 、< owl:disjointWith >
2	属性之间的继承、等同、互逆关系	< rdfs:subPropertyOf > 、< owl:equivalentProperty > 、< owl:inverseOf >
3	属性与类之间关系	< rdfs:domain > 、< rdfs:range >
4	类与实例之间关系	< rdf:Description > 、< rdf:type >
5	知识间的传递、对称、函数和反函 数关系	owl:TransitiveProperty、owl:SymmetricProperty、owl:FunctionalProp- erty 和 owl:InverseFunctionalProperty
6	知识间的集合运算关系	< owl:unionOf > 、< owl:intersectionOf > 、< owl:complementOf >
7	同一个知识的不同版本关系	< owl:priorVersion >

2.2.2 互通

互通是指在知识供需双方间可以交换知
识。互联是语义 Web 互通的必要条件,但不是
充分条件。除了互联之外,网络协议、领域本体
是语义 Web 环境知识联网的重要影响因素。

(1)网络协议。除了 TCP/IP 协议簇和 ISO/
OSI 体系结构之外,SOAP(Simple Object Oriented
Protocol)是语义 Web 互通层次最重要的协议。

SOAP 是一种基于 XML 的消息协议,它指定从
一个过程发送基于 XML 的消息到另一个过程的
方式,通常是从一台机器到另一台机器^[5]。
SOAP 协议具有较好的封装机制,可以发送任何
XML 消息。因此,SOAP 协议更符合语义层次上
的知识联网的需求。

(2)本体。构建领域本体是语义 Web 互通
的另一个重要前提条件。本体的目标是获取、

描述和表示相关领域的知识,提供对该领域知识的共同理解,确定共同认可的词汇,从不同层次的形式化模式上给出这些词汇和词汇间的相互关系^[10]。因此,本体确保知识供需双方能够对同一个知识达成共识,是语义 Web 中知识供需双方互通的重要基础。一些文献^[11-12]较详细探讨了如何在语义 Web 中进行基于本体的知识处理问题。

2.2.3 互操作

互操作是指网络中知识供需双方之间透明地访问对方资源。语义知识本身并没有互操作能力,语义互操作性应由高层和应用实现。从目前的发展趋势看,Agent 技术尤其是智能 Agent 是语义互操作的主要技术。FIPA (Foundation for Intelligent Physical Agents) 规范和 JAS (Java Agent Services) 接口为语义 Web 中的 Agent 技术进一步发展提供了重要基础^[13]。文献^[14]总结了用于语义 Web 的 Agent 技术的分类、特征及应用,并认为多主体系统 (Multi-Agent System, MAS) 是语义 Web 中实现互操作的重要技术手段。

根据上述知识联网的探讨和图 1 所示,语义 Web 中的知识互联过程具有三个重要特征:①语义 Web 中知识表示的目的之一是在数据之间建立计算机可理解的联网关系;②语义知识联网可以分为互联、互通和互操作三个层次,其中语义互联是语义联网的基础,语义互通是手段,语义互操作是语义 Web 的最终目的之一;③知识联网过程也是一种人机合作过程,如智能 Agent 可以用计算机完成,领域本体的建设应由人负责完成。

2.3 知识发现

语义 Web 与传统 Web 的第三个不同在于语义 Web 支持以知识原子 (Knowledge Atom) 为单位进行表示、检索、推理和抽取知识。知识原子被认为是知识的最小构成单位,是不可再分的知识元素^[15]。语义 Web 采用“对象—属性名称—属性值 (Object-Attribute-Value)”组成的三元组 (Tripe) 构成的语句 (Statement) 来描述一个知识原子。基于语义 Web 的知识发现既可以发

现知识原子为目的,也可以发现由知识原子组成的知识集为目的。从流程上看,基于语义 Web 的知识发现可以分为检索、推理和验证三个主要步骤。

2.3.1 检索

语义 Web 的出现改变了传统信息检索中查准率和查全率的概念。与传统 Web 不同的是,在语义 Web 中进行检索时,不仅可以发现知识文档,而且可以进一步查找文档内部的知识片段,甚至准确定位到知识原子层。在发现知识所在的语义文档之后,可以采用路径表达式 (Path Expression) 对知识原子进一步检索,以便抽取保存在文档中的目标知识片断或知识原子。常见的语义文档检索语言有以下几种:

(1) XML 文档查询语言。Xpath、Xpointer、XML-QL、XQuery 是常用于 XML 文档的查询语言。Xpath 是一种查询部分 XML 文档的通用语言标准;Xpointer 是一种指向 XML 文档的特定区域或特定部分的机制;XML-QL 则提供了类似于访问层次型数据库和关系型数据库的查询技术。

(2) RDF 文档查询语言。SPARQL 是一种典型的 RDF 文档查询语言,检索 RDF 文档的能力较强。在 SPARQL 语言中,可以采用类似于 SQL 语言的 select-from-where 结构进行检索,方便编写查询代码。

(3) OWL 文档查询语言。OWL-QL (OWL Query Language) 和 OWL SAIQL 是典型的 OWL 查询语言。OWL-QL 作为一种演绎式的问答语言,较准确地描述了问题、答案和知识库之间的语义联系。与数据库语言和其他 Web 查询语言不同的是,OWL-QL 支持问答式会话,答案代理具有自动推理所需答案和进行跨库检索等语义操作能力^[16]。OWL SAIQL 是一种基于 OWL DL 的查询语言,可用于 OWL DL 文档的查询操作。

2.3.2 推理

语义 Web 与传统 Web 的第四个区别在于语义 Web 中增加了“计算机可理解的规则系统”,可分为单调规则 (Monotonic Rules) 和非单调规则 (Nonmonotonic Rules) 两种,分别用于不完全信息情况和完全信息情况下的语义推理。

语义 Web 的规则推理机制为基于语义 Web 的知识挖掘、推理和验证操作奠定基础,进一步推动知识“机器可理解”化的进程。从目前的研究和应用进展看,常用于描述语义 Web 规则系统的语言有三种,即 DLP、RuleML 和 SWRL。

(1) DLP。DLP (Description Logic Programs, 描述逻辑程序) 是谓词逻辑和 OWL 的结合,主要由 OWL 语言和谓词逻辑的交叉部分组成^[4]。因此 DLP 只能表示部分谓词逻辑或部分 OWL 语言的概念,规则表达和推理能力有限。

(2) RuleML。RuleML (Rule Markup Language, 规则标记语言) 是一种在万维网上共享和发布规则的标记语言。RuleML 语言内核是基于 Horn 逻辑的 Datalog 语言, Datalog 是 SQL 语言和 Prolog 语言的交叉语言。在 RuleML 语言中,运用 XML Schema 描述语法,并通过 Herbrand 模型来定义语义^[17]。2008 年度国际 RuleML 语言论坛于 2008 年 10 月 31 日在美国佛罗里达州奥兰多举行,会议主要讨论分布式规则技术和基于规则的应用系统。

(3) SWRL。SWRL (Semantic Web Rule Language, 语义 Web 规则语言) 是结合 RuleML 和 OWL 语言的语义 Web 规则语言^[18]。SWRL 支持运用 OWL DL 和 OWL Lite 描述 Horn 规则的高层次抽象语法。因此,相对于 RuleML 语言,SWRL 可以更好地描述语义 Web 中的规则机制。

2.3.3 验证

语义 Web 与传统 Web 的第五个不同在于语义 Web 中的知识更便于验证,尤其使运用知识的语义信息进行校验成为可能。语义 Web 环境下,知识仍然面临着过时、伪造、篡改、拒绝服务、行为否认、非授权访问等安全威胁。因此,如何验证知识的有效性、保密性、完整性、可用性、可控性、不可否认性也是促进语义 Web 健康发展的重要课题。从目前国内外研究来看,可用于语义 Web 知识验证的技术有以下几种:

(1) 传统数据验证技术。采用类似于奇偶校验、CRC 校验、Hash 码的方法判断知识是否被伪造或非法篡改。此类技术的实现方法简单,成本较低,但是其适用范围较有限,效果有时很

难保障。

(2) 数字签名技术。在这种技术体系中,知识的提供方使用自己的私钥对数据校验和与其他数据内容有关的变量进行加密处理,完成对数据的合法签名;知识需求方则利用对方的公钥解读收到的数字签名,并将解读结果用于对数据完整性的检验,以确认签名的合法性。

(3) 语义层次的验证技术。语义 Web 的出现为传统信息安全领域提供了新思路,即基于语义信息判断信息的保密性、完整性、可用性、可控性、不可否认性。近年来,有些学者开始意识到语义层次的验证技术的必要性和可行性,但尚未提出可行的验证技术。

除了上述技术之外,知识提供方的“信用信息”将会是验证语义知识的重要依据。语义 Web 中的信用信息主要来自其他用户的评价、知识被调用的次数、黑名单机制等。另外,知识验证时需要考虑复杂性和安全性之间的平衡问题。在特定问题的处理上,计算复杂性和安全性是一对矛盾。语义 Web 的出现一定程度上缓解了这一矛盾。

根据以上探讨和图 1 所示,在知识发现过程中语义 Web 具有如下特性:①语义 Web 中知识的具体存储位置对知识用户(智能代理或人类)来说是透明的,知识用户对语义 Web 进行挖掘才能发现所需知识;②由于在语义 Web 中可以知识原子(或知识片断)为单位进行标记,并提供语义关联信息,所以基于语义 Web 的知识检索查准率和查全率远远高于传统网络;③语义 Web 的规则系统可以满足用户的逻辑推理需要,有利于挖掘更多的知识或对知识有效性进行验证;④语义 Web 的规则系统建设必须由人参与完成,即建设语义 Web 的规则系统的过程也是人机结合的过程。语义 Web 的出现改变了传统 Web 信息检索模式,知识检索的效果和效率得到了较大提高。

2.4 知识获取

语义 Web 与传统 Web 的第六个不同是在语义 Web 中可以更容易地抽取知识片断,提高获取知识的速度和准确度。需求方对目标知识

进行验证后,得到对方的“信任”才能获得抽取所需知识的权限。这种“信任”关系可能是默许的,也可能是强制性的。知识抽取可以以知识原子或知识片段为单位抽取,改变传统 Web 检索结果的返回机制。用户成功抽取知识后可对其进行评价,为其他用户的选择提供参考。

2.4.1 信任

在语义 Web 中,知识的需求方发现所需知识后,有时需经知识提供方的授权“信任”后才能进行知识抽取。语义 Web 可以继续采用传统 Web 的信任机制,也可以重新设计基于语义 Web 的新型信任机制。传统 Web 信任机制有很多种,如黑名单机制、CA 认证、加密技术、数字签名等。文献[13]探讨了基于语义 Web 的信任机制,如 XML 数字签名(XML DigSig)和加密(XML Enc)、语义授权模式以及相关等问题。

2.4.2 抽取

知识抽取是语义 Web 的重要特征之一,知识需求方获得授权后可以对所需求知识进行抽取。在语义 Web 中,知识的单位是知识原子。语义 Web 中独特的知识表示方法较好地支持了知识抽取,可以选择任何知识表示单位,如知识原子、知识片段、知识文档等。语义 Web 的精准定位知识和任意抽取知识的特征,可以提高知识检索和挖掘的效率。RSS 和 Wrapper 是目前常用的两种知识抽取技术。除此之外,语义 Web 中的查询语言也较好地支持了知识抽取活动。

2.4.3 传递

知识抽取结果的反馈是语义 Web 知识处理的重要环节之一。语义 Web 中知识的传输过程可以采用 P2P(Peer to Peer)技术、BT 技术(Bit Torrent)等。文献[19]认为语义 Web 和知识管理的结合在信息共享和知识管理中具有巨大的可挖掘的潜力。在需求方成功接收所需知识后,可以对所获知识或知识的供应方进行评价。用户反馈评价是语义 Web 中信用信息的重要来源。

根据对语义 Web 知识获取过程的探讨以及图 1 所示,语义 Web 中的知识获取具有以下特征:①知识用户在语义 Web 中发现所需知识后,

往往在得到知识提供方的“信任”后才能获取知识,信任机制是语义 Web 技术健康发展和推广的重要前提;②与传统 Web 不同的是,知识用户可以在语义 Web 中轻松实现知识的抽取操作。语义 Web 既可以提高知识检索的准确率,也可以降低反馈结果的传递成本;③知识用户可以对抽取知识进行评价,为其他用户的选择提供参考。可见,语义 Web 中的知识获取过程主要由计算机来完成,但不排斥人的参与。

2.5 知识集成

语义 Web 与传统 Web 的第七个不同是语义 Web 中的知识更易满足知识集成的需求。语义 Web 中的知识集成可以分为以下三种:

2.5.1 与显性知识集成

从语义 Web 中成功获取所需知识后,可以将其与显性知识进行集成,有以下三种技术方法:

(1)虚拟视图。在虚拟视图方法中,中间集成系统不需要存储实际数据,可驻留在各自原有的系统中。虚拟视图法的数据接口和功能接口紧密联系在一起,数据接口实现原有系统的数据提取并转换成集成系统可接受的数据形式,功能接口则依据集成需求约束实现与集成系统的衔接^[20]。

(2)公共数据仓库。与虚拟视图方法不同的是,公共数据仓库方法中的数据接口和功能接口是完全分离的,连接二者的中间部分就是公共数据仓库。采用公共数据仓库方法就是在用户提出查询之前,把需要共享的信息从各个数据源中提取出来预先转换成统一模式,异构环境在公共数据库上得到统一,再由集成系统提供对公共数据库的数据访问机制^[20]。

(3)知识网格。除了上述两种基本技术之外,知识网格是语义 Web 中进行知识集成的重要手段。知识网格是一种建立在万维网上的知识共享环境。它以网格/语义 Web 为基础架构,实现异构语义资源的整合与获取,并提供基于语义资源的知识服务,其本质是万维网知识资源和知识服务的协同与共享^[21]。

2.5.2 与应用程序集成

从语义 Web 中成功获取所需知识后,有时需要与应用程序进行集成。语义 Web 中的知识与应用程序集成的关键在于其间的接口。从目前的理论与技术条件看,应用程序可以通过以下几种接口方式访问语义知识:

(1)XML 编程接口,如 DOM、SAX、JAXP 等。DOM(Document Object Model)是一种与平台和语言无关的应用程序接口,采用对象模型描述文档的结构。通过 DOM 接口,应用程序可以对 XML 文档进行加载、遍历、操作控制节点以及进行出错处理等操作。与 DOM 接口相比,SAX(Simple API for XML)接口规范是由 XML 分析器和 XML 处理器提供的,属于更底层的接口,更具有灵活性。此外,SAX 还支持基于事件的处理模式。JAXP(Java API for XML Programming)支持用户随意更换解析器,不需要修改应用程序就可以访问 XML 语义 Web 中的知识。

(2)RDF 编程接口。从目前的应用看,Jena 是最常用的 RDF 编程接口之一。Jena 提供了 ModelFactory.createDefaultModel()、Model.listStatements()、Model.write()/Model.read()方法、jena.rdfquery 工具和 com.hp.hpl.jena.rdql 包,分别用于创建 RDF 模型、RDF 查询、RDF 读写、从命令行执行 RDQL 查询和 Java 代码中使用 RDQL 查询等功能。此外,Jena 还提供了 OWL 和 SPARQL 文档的编程 API 接口。

(3)OWL 编程接口。除 Jena 也支持 OWL 编程外,Protégé-owl API 是专门为 OWL 语言和 RDF 语言开发的一种 Java 开源接口。Protégé-owl API 分别提供了 OWLNamedClass/OWLAnonymousClass、OWLDatatypeProperty/OWLObjectProperty、OWLAllValuesFrom/OWLSomeValuesFrom/OWLHasValue、OWLUnionClass/OWLComplementClass/OWLEnumeratedClass 等类可用于 OWL 中的类、属性、属性值和逻辑运算,以便用户进一步开发自己的语义 Web 软件或语义 Web 中间件,实现语义 Web 中的知识与应用系统的集成。

2.5.3 与隐性知识集成

语义 Web 中的知识较好地显示了知识的语义内容及知识间的相互关系,因此,语义 Web 中的知识更容易与人的隐性知识集成。从集成知

识的需求分析,集成语义 Web 中的知识与隐性知识的工具有:

(1)语义桌面。语义桌面是本地计算机中的知识与人的隐性知识集成的重要手段。传统资源管理器主要以知识资源的语法特征为依据进行资源管理,缺少必要的语义理解和关联,管理和维护不便。语义桌面的出现改善了传统资源管理模式,能够促进语义 Web 中的知识与人的隐性知识的集成。语义桌面架构主要分为数据层和应用层,数据层存储信息和本体,应用层通过调用数据层 API 为用户提供信息查询、语义标注和本体编辑等功能^[19]。语义桌面可建立知识的语义关联,提高资源管理的效率,实现语义 Web 中知识的隐性化。

(2)语义门户。语义门户是集成分布式语义 Web 中的知识与隐性知识的重要手段。语义门户的核心就是用本体的理念组织知识资源,为机构用户的信息交流提供管理机制和信息集成,所以语义门户具有了语义浏览和语义查询功能^[22]。

根据上述对语义 Web 知识集成过程的探讨和图 1 所示,基于语义 Web 的知识集成具有以下特征:①语义 Web 知识经过与显性知识、应用系统和隐性知识进行集成才能发挥作用;②通过知识之间的集成,既可以发现新知识,也可以验证所获取知识的有效性。虚拟视图、公共数据仓库和网格技术为已获取知识与显性知识的集成提供了技术手段;③为了运用应用系统进一步自动处理语义 Web 反馈的知识,开发了多种编程 API 接口,如 Jena 和 Protégé-owl API 等;④语义 Web 中的知识与人的隐性知识的集成是显性知识隐性化的过程。人理解知识是一种典型的知识集成过程,用已知知识理解新知识往往是在无意识情况下进行的。

综上所述,语义 Web 是一种人机结合的网络,充分利用了人与计算机的优越性。首先,人参与知识的表示过程,将各种知识以计算机可理解的形式提供给计算机,降低了处理知识的技术难度。其次,计算机快速、精确的计算和逻辑推理特征扩展了人在知识检索、推理、获取和集成等方面的能力,加快了知识处理过程。最

后,人对语义 Web 的管理和反馈为语义 Web 的健康发展和不断优化提供了重要保障。

3 基于语义 Web 的知识处理流程的技术框架

从技术维度看,语义 Web 是传统 Web 的一种扩展,不需要重新建设。结合上述对语义 Web 知识处理流程的分析,可以将语义 Web 中的知识处理技术分为知识表示层技术、知识互联层技术、知识发现层技术、知识获取层技术和知识集成层技术五个层次,图 2 是基于语义 Web 的知识处理流程技术框架。

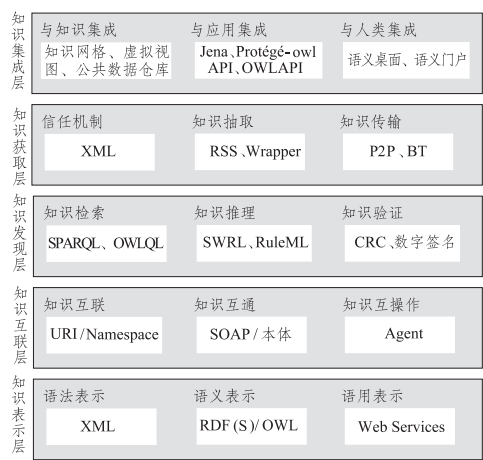


图 2 基于语义 Web 的知识处理流程技术框架

3.1 知识表示层技术

该层技术的主要功能是将知识以“计算机可理解”的形式进行表示。语义 Web 中的知识表示技术必须具备平台独立、语义表示能力强的特点,并且需要实现知识的内容、结构和显示之间的独立。常见的知识表示层技术有 XML 语言、RDF(S)模型、OWL 语言和 Web Services 技术。其中,XML 语言是语义 Web 中知识表示的语法基础,RDF(S)是知识建模的主要思路,OWL 是语义表示的常用语言,通过 Web services 可以将语义 Web 中的知识包装为语义服务。

3.2 知识互联层技术

该层技术的主要功能是实现语义 Web 的核心功能——建立计算机可理解的数据网络。语义 Web 中知识互联层技术可以进一步分为互联技术、互通技术和互操作技术三种。URI 和 Namespace 技术为知识互联提供了基础条件;SOAP 协议和领域本体为知识供需双方的达成共识和成功交换知识提供了重要技术手段;而智能 Agent 则是语义 Web 中知识互操作的常用技术。

3.3 知识发现层技术

该层技术的主要功能是根据知识需求方的提问对语义 Web 进行检索、推理和验证。在知识检索方面,除了现有的知识检索技术外,语义 Web 还提供了 SPARQL、OWL-QL 等检索知识原子或知识片断的方法。常用于语义 Web 中的知识推理技术有 SWRL、RuleML 等,CRC、数字签名等为语义 Web 中知识的验证提供了技术手段。

3.4 知识获取层技术

该层技术的主要功能是为知识用户提供授权、抽取和传递技术。在知识授权方面,语义 Web 主要采用了信任机制。RSS 和 Wrapper 技术是语义 Web 中知识抽取的重要技术,常用的语义 Web 传递技术有 P2P 和 BT。

3.5 知识集成层技术

该层技术的主要功能是实现从语义 Web 抽取知识集成到显性知识、应用系统或人的隐性知识中。知识网格、虚拟视图、公共数据仓库技术是语义知识与显性知识进行集成的常用技术;而通过 Jena、Protégé-owl API、OWL API 接口,可以实现语义 Web 中的知识与应用系统的集成;语义桌面和语义门户为语义 Web 知识与隐性知识的集成提供了重要手段。

4 结论

综上所述,语义 Web 在知识的表示、联网、发现、获取和集成方面均体现了区别于传统 Web 的特性。语义 Web 的知识表示层和知识互

联层技术降低了计算机对 Web 知识处理的难度,使现阶段的人工智能技术能够适应 Web 知识的处理活动。本文提出的基于语义 Web 的知识处理流程及其技术框架较好地结合了人脑和电脑在知识处理方面的不同优势。从知识处理的流程维度看,基于语义 Web 的知识处理流程对人类和计算机进行了合理的分工和有效的定位,实现了人机在知识处理流程中的互补作用;从知识处理的技术维度看,基于语义 Web 的知识处理流程的技术框架建立在传统 Web 技术的基础上,在多层次上集成了知识处理技术,为人机结合处理知识提供了新的技术方案。

参考文献:

- [1] McElroy M W. The new knowledge management: Complexity, learning, and sustainable innovation [M]. KMC Press, 2003: 9-12.
- [2] Berners-Lee T, Hendler J, Lassila O. The semantic web [R/OL]. (2001-05-15) [2008-10-05]. http://www.ryerson.ca/~dgrimsa/courses/cps720_02/resources/Scientific%20American%20The%20Semantic%20Web.htm.
- [3] Berners-Lee T, Fischetti M. Weaving the web: The original design and ultimate destiny of the World Wide Web [M]. HarperCollins Publishers Inc., 2000: 157.
- [4] 顾基发,王浣尘,唐锡晋. 系统集成方法体系与系统学研究[M]. 北京: 科学出版社, 2007: 9-10.
- [5] Antoniou G, Van Harmelen F. A semantic web primer (Second Edition) [M]. The MIT Press, 2008: 1-3.
- [6] Baillinger K. NET web services: Architecture and implementation [M]. Pearson Education, Inc., 2004: 15-31.
- [7] Martin D, Domingue J, Brodie M L, et al. Semantic web services, Part 1[J]. IEEE Intelligent Systems, 2007(5): 12-17.
- [8] Kashyap V, Bussler C. The semantic web: Semantics for data and services on web [M]. Springer Press, 2008.
- [9] Fensel D, Kerrigan M, Zaremba M. Implementing semantic web services: The SESA framework

[M]. Springer, 2008.

- [10] 宋炜,张鸣. 语义网简明教程[M]. 北京: 高等教育出版社, 2004: 111.
- [11] Davies J, Fensel D, Van Harmelen F. Towards the semantic web ontology-driven knowledge management [M]. John Wiley & Sons Ltd., 2003.
- [12] Davies J, Studer R, Warren P. Semantic web technologies trends and research in ontology-based systems [M]. John Wiley & Sons Ltd., 2006.
- [13] Passin T B. Explorer's guide to the semantic web [M]. Manning Publications Co., 2004: 213-217.
- [14] Leuf B. The semantic web crafting infrastructure for agency [M]. John Wiley & Sons Ltd., 2006: 97-111.
- [15] 温有奎,徐国华,赖伯年,等. 知识元挖掘 [M]. 西安: 西安电子科技大学出版社, 2005: 30.
- [16] Fikes R, Hayes P, Horrocks I. OWL-QL-A Language for deductive query answering on the semantic web [J]. Journal of Web Semantics, 2005, 2(1): 19-29.
- [17] Boley H, Groszof B, Tabet S. RuleML tutorial [R/OL]. (2001-05-13) [2008-10-15]. <http://www.ruleml.org/papers/tutorial-ruleml-20050513.html>.
- [18] Horrocks I, Patel-Schneider P F, Boley H, et al. SWRL: A semantic web rule language combining OWL and RuleML [R/OL]. (2003-11-19) [2008-11-13]. <http://www.daml.org/2003/11/swrl/>.
- [19] Staab S, Stuckenschmidt H. Semantic web and peer-to-peer decentralized management and exchange of knowledge and information [M]. Springer, 2006.
- [20] 顾进广,陈萃萌. 基于语义的 XML 信息集成技术 [M]. 武汉: 武汉大学出版社, 2007.
- [21] 马文峰,杜小勇. 知识网格研究 [J]. 图书情报工作, 2007(10): 64-68.
- [22] 田晓迪,宋文. 语义门户的框架研究 [J]. 情报科学, 2007(1): 401-701.

朝乐门 中国人民大学信息资源管理学院博士研究生, 北京石油化工学院讲师。通讯地址: 北京大兴区黄村滨河北里那尔水晶城 18-3-601 信箱。邮编 102600。

(收稿日期: 2009-03-06)