

语义 Web 应用程序开发方法及实例分析

李新龙, 黄映辉

(大连海事大学 信息科学技术学院, 辽宁 大连 116026)

摘要:随着语义 Web 技术的不断发展, 语义 Web 应用程序越来越受到重视, 但现在国内对语义 Web 应用程序的研究却比较少, 缺少语义 Web 应用程序的开发方法。文中通过对语义 Web 应用程序的研究, 结合与 Web 应用程序的对比分析, 给出了语义 Web 应用程序的定义、架构以及开发方法, 并详细说明了基于数据层、逻辑层和表现层三层架构的语义 Web 应用程序的结构特征和构建过程, 进而通过构建一个语义 Web 应用程序实例对所提出的开发方法进行了验证, 取得了预期的成果。

关键词:语义 Web 应用程序; 开发方法; 实例分析; Jena; 信息语义检索

中图分类号: TP301.2

文献标识码: A

文章编号: 1673-629X(2013)12-0086-06

doi: 10.3969/j.issn.1673-629X.2013.12.021

Semantic Web Application Developing Method and Example Analysis

LI Xin-long, HUANG Ying-hui

(College of Information Science and Technology, Dalian Maritime University, Dalian 116026, China)

Abstract: With the continues development of semantic Web technology, more and more attention has been drawn to semantic Web applications. However, less research has done on the semantic Web applications in the domestic, lacking of development methods. By study, the definition of the semantic Web application is given, as well as the basic framework and the developing method. The structure characteristics and building process of each layer in semantic Web application based on data layer, logic layer and presentation layer, are also described. Through constructing a semantic Web application example, a novel development method is verified and achieve the prospect result.

Key words: semantic Web program; developing method; example analysis; Jena; information semantic retrieval

0 引言

随着语义 Web 技术^[1]的不断发展, 目前对语义 Web 的研究已经逐渐从理论研究延伸到实际应用层面, 基于 RDF 和 OWL 的语义 Web 应用已经开始应用到商业产品中, 数据库开始支持 RDF 和 OWL 文档的存储, 浏览器也能够支持 RDF 和其他本体文档的解析, 语义 Web 技术的应用已经越来越广泛, 因此对语义 Web 应用程序的研究逐渐受到重视。

目前国内外对语义 Web 应用程序的研究大多处于理论阶段, 缺少语义 Web 应用程序开发的具体方法, 面向用户的语义 Web 产品也比较少, 因此对语义 Web 应用程序的研究具有非常重要的意义。相对于当前运行在 Web 上的应用程序, 语义 Web 应用程序能够更加充分地利用语义 Web 的优势, 突出了“以数

据为中心”的程序设计思想, 将语义从信息中提取出来, 实现信息的最大化利用。简而言之, 语义 Web 改进了应用程序的功能, 使之能够在 Web 这样的尺度上高效地利用大量多元化的信息^[2]。此外为了能够支持丰富的信息操作, 语义 Web 上的信息描述是通过一种结构化、标准化的方式来实现的。

1 语义 Web 应用程序

1.1 定义

现有的 Web 是面向人类的。面对信息的爆炸性增长, 计算机无法对信息进行完全符合人类请求的处理。因此 Web 的发明者 Tim Berners-Lee 在看到 Web 的不足之后提出了语义 Web 的概念。他认为语义 Web 并不是另一个 Web, 而是对现有 Web 的一种扩

收稿日期: 2013-02-28

修回日期: 2013-06-05

网络出版时间: 2013-09-29

基金项目: 国家自然科学基金资助项目(60972090)

作者简介: 李新龙(1987-), 男, 山东潍坊人, 硕士研究生, CCF 会员, 研究方向为智能信息处理; 黄映辉, 教授, CCF 高级会员, 研究方向为智能信息处理。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130929.1522.019.html>

展,它主要是在 Web 的基础上添加了一些可以被计算机理解的语义信息,在方便人类使用的同时也使计算机可以理解并处理这些信息,因此它面向的对象不再仅仅是人还包括机器。

Web 必然被语义 Web 替代,原有的 Web 应用程序是基于 Web 的 B/S(浏览器/服务器)架构的产物,已经不能适用于语义 Web。因此需要将 Web 应用程序迁移到语义 Web 上^[3-4],这就形成了语义 Web 应用程序^[5-6]。

语义 Web 是对 Web 的一种扩展,Web 应用程序迁移到语义 Web 上之后形成的语义 Web 应用程序也是 Web 应用程序的一种扩展。和经典的 Web 应用程序相同,语义 Web 应用程序同样是基于 B/S 架构的产物。语义 Web 应用程序是以数据为中心,并将语义直接含于数据当中,它能够适时地通过语义 Web 访问和共享丰富的信息并对信息进行动态地处理。

1.2 架构

语义 Web 应用程序是由 Web 应用程序迁移而来,为了减少新技术的应用所带来的工作量的陡增,应尽可能地减少这种迁移所带来的改变。因此语义 Web 应用程序同样采用 Web 应用程序常用的三层架构模式^[7],即数据层、逻辑层和表现层。语义 Web 应用程序三层架构如图 1 所示。

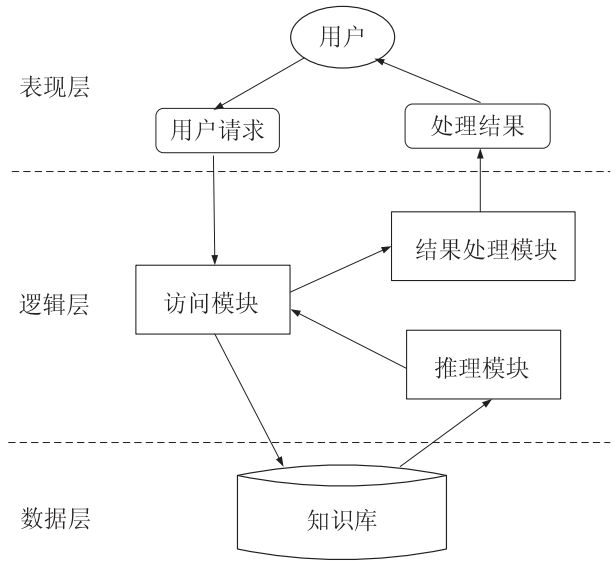


图 1 语义 Web 应用程序的架构

1.2.1 数据层

数据层主要负责对原始数据进行操作,它是语义 Web 应用程序的数据中心,这一层主要通过知识库中的本体来体现。

本体(ontology)是语义 Web 应用程序的基础与核心。针对不同的语义 Web 应用程序有不同的本体,即通常所说的领域本体。目前在很多领域已经存在很多不同的比较完善的领域本体,可以直接整合到语义

Web 应用程序中,同时也可以对已有的本体进行修改,得出语义 Web 应用程序需要的本体。一个构造良好的本体支持跨语义 Web 应用程序的交流,同时语义 Web 应用程序也能够利用不同的本体。

知识库(knowledge base)主要是负责对语义 Web 应用程序所需要的本体进行存储和管理,并提供对本体进行访问和推理的接口。知识库类似于 Web 应用程序中的关系数据库,但却有明显的不同,例如知识库提供多维度的关系,关系数据库只有一种关系即外键。两者的比较如表 1 所示。

表 1 关系数据库与知识库不同点

特性	知识库	关系数据库
结构	本体	模式
数据	实例	记录
管理语言	本体语言	DDL
查询语言	SPARQL	SQL
关系	多维度	单一外键
惟一性	URI	主键

1.2.2 逻辑层

逻辑层主要负责对用户的请求进行处理,并根据处理之后的请求与数据层进行交互,然后将处理之后的结果返回给表现层。这一层主要通过三个模块来体现,即推理模块、访问模块和结果处理模块。

访问模块主要作用是 与知识库进行交互。访问模块通过接收到表现层的用户请求后对知识库进行访问,并将推理模块推理之后的结果返回给结果处理模块。

推理模块主要作用是进行知识推理。当访问模块对数据层的知识库进行访问时,推理模块结合用户的请求以及推理规则对知识库中的知识进行推理,并将结果返回给访问模块。

结果处理模块主要作用是将访问模块返回的结果进行处理,并根据用户的请求对结果进行排序,并按照一定的顺序将结果返回给表现层。

1.2.3 表现层

表现层主要负责 人机交互,这一层主要通过浏览器页面来体现。

Web 应用程序面向的对象是人,所以 Web 应用程序表现层主要是用于人机交互,即用户提出请求,表现层将请求传给 Web 服务器,然后将 Web 服务器返回的结果呈现给用户。但语义 Web 程序面向的对象已经不只是人,还要与其他的语义 Web 应用程序进行交互,所以表现层中需要有面向其他语义 Web 应用程序的接口,便于它们之间的交互。

2 语义 Web 应用程序的开发

2.1 开发框架

语义 Web 应用程序开发框架能够为语义 Web 应用程序提供一个一致且完备的编程环境。由于现有的框架所使用的程序设计语言并不是 OWL,因此开发框架需要将语义 Web 应用程序中的类和实例等转换成编程中所使用过的类、对象和方法。

目前有许多优秀的程序设计框架可供选择,例如 Jena^[8]、Sesame^[9]等。

Jena 是惠普实验室提供的开放资源,它是用于开发语义 Web 应用程序的一个 Java 框架,它为 RDF、RDFS、OWL 提供一个程序开发环境。首先 Jena 中拥有对 RDF 文件和模型进行处理的 RDF API 和 OWL API,以及对 RDF、RDFS、OWL 文件(基于 XML 语法)进行解析的解析器,可以用于数据层开发;其次 Jena 中拥有支持信息语义检索 SPARQL 查询语言的 ARQ 模块,用于知识推理的基于规则的推理机子系统 Jena 推理机等,可以用于逻辑层的开发。因此选择 Jena 框架用于语义 Web 应用程序的数据层和逻辑层的开发。基于 Jena 框架的语义 Web 应用程序三层架构如图 2 所示。

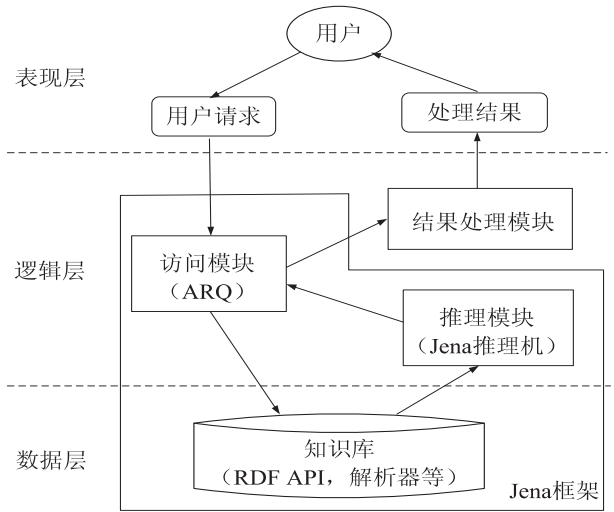


图 2 基于 Jena 框架的语义 Web 应用程序的架构

2.2 数据层开发

数据层的开发主要包括建立知识库和填充知识库两步。建立知识库主要是获得一个结构化模型(存储空间),用以存储语义 Web 应用程序所需要的本体;填充知识库则是将需要的语义 Web 应用程序的数据以本体的形式填充到知识库当中。

2.2.1 建立知识库

在 Jena 框架中,建立知识库就要创建 Model 对象,Model 类包含了对语义 Web 数据进行管理的方法。每一个语义 Web 应用程序可以使用多种本体类型的模型以便获取多种不同层次的推理支持。

利用 ModelFactory. CreateOntologyModel() 创建一个模型的方法为:

```
Private void aquireMemoryForDate() {
    modelMem = ModelFactory. CreateOntologyModel();
}
```

该方法还可加入参数,用于指定模型内出现蕴含的方法。

2.2.2 填充知识库

在创建模型之后,就需要用知识来进行填充。填充模型的方式有很多种,可以通过本体文件来填充,也可以通过直接加入陈述来填充,还可以通过已经存在的模型来填充。其中通过本体文件来填充是最常用和最简单的方式。

利用本体文件 Ontology. rdf 来填充抽象模型的方法为:

```
Private void addDateFromFile() {
    InputStream inOntology = FileManager. get(). open("Ontology. rdf");
    modelMem. read(inOntology, defaultNameSpace);
    inOntology. close();
}
```

对于使用本体文件填充模型,首先要创建一个 InputStream,将其作为 read() 方法的参数,进而来填充 modelMem 模型。

2.3 逻辑层开发

逻辑层的开发主要包括访问模块开发、推理模块开发和结果处理模块开发三步。访问模块主要是对知识库进行访问,并返回请求的结果;推理模块主要对知识库中的知识进行推理,来获得隐含的知识;结果处理模块主要是为了对结果进行符合用户请求的处理,获取最佳排序结果。

2.3.1 访问模块

访问模块通过 Jena 的 ARQ 模块来实现。

访问模块使用 Jena 中 ARQ 模块所支持的形式化语言 SPARQL^[10],它是 W3C 推荐的标准语言,有大量的可用的处理程序,而且可以根据访问形式的不同可以返回不同格式的结果。

利用 SPARQL 进行知识库访问的方法为:

```
Private void queryData() {
    Query query = QueryFactory. create(query, modelMem);
    QueryExecution qexec = QueryExecutionFactory. create();
}
```

首先利用 QueryFactory. create() 方法创建一个对象,然后使用 QueryExecutionFactory. create() 方法对知识库进行访问。

2.3.2 推理模块

推理模块使用 Jena 推理机来实现。

当访问模块对知识库进行访问时,很多隐含的知识无法得到,这就需要推理模块中的推理机进行推理以获得这些隐含的知识;同时还可以给推理机添加自定义推理规则,使推理机可以支持超出逻辑演绎能力的推理功能。

推理机主要有内置于 Jena 框架中的内部推理机,以及以外部 Java 文件形式提供的外部推理机,例如 Pellet。

Jena 内部有多个推理机,文中将使用 Jena 内部默认的 OWL 推理机,推理机一直保持和模型的绑定,并能对模型内部的知识进行推理以获取新知识。获取 Jena 推理机并绑定到模型的方法为:

```
Private void bindReasoner() {
    Reasoner reasoner = ReasonerRegistry.getOWLReasoner();
    Reasoner = reasoner.bindSchema(modelMem);
}
```

首先调用 ReasonerRegistry.getOWLReasoner() 方法获取 Jena 内部的 OWL 推理机,然后利用 reasoner.bindSchema() 方法将其绑定在前面创建的模型 modelMem 中。

Jena 同样支持用户根据语义 Web 应用程序的需要添加自定义的规则,一条 Jena 规则包含主题项列表和头部项列表,而且每条规则可以选用一个名称。Jena 规则示例如下。

```
rules = "[ (? m Book;Name ? name) , strConcat(? name, ? lit);regex(? lit, \"q\")→(? m sameIndividual ? p);]"
```

同时 Jena 还可以利用外部的 Pellet 推理机代替默认的推理机,调用外部推理机的方法为:

```
Private void bindPellet() {
    Reasoner reasoner = PelletReasonerFactory.theInstance().
    creat();
    Reasoner = reasoner.bindSchema(modelMem);
}
```

2.3.3 结果处理模块

结果处理模块可以通过倒排索引技术^[11]来实现。

经过推理机的作用,许多隐含的信息作为请求的结果被发现,但是由于结果输出之后并没有按照用户的需求进行排序,所以并不是最优的输出,需要对结果进行相关性排序,结果处理模块可以使那些与用户请求相关性更高的结果出现在前面。

倒排索引技术因功能强大、实现简单而被广泛应用于检索系统中。文中将倒排索引技术应用到知识库实例的检索中。倒排索引是把实例解析为概念的集合,并且在索引中记录包含概念的所有实例及概念在实例中的频率、位置等信息。

当在知识库中进行实例检索时,只对概念表进行扫描,通过概念与实例的倒排索引可以快速对应到包

含该关键字的实例,然后根据概念在实例中的频率得出更加符合检索请求的排序结果。

2.4 表现层开发

表现层的开发主要包括浏览器页面的开发。语义 Web 应用程序表现层的主要作用是用于人机交互以及语义 Web 应用程序之间的交互,由于语义 Web 应用程序之间的交互比较复杂,文中仅限于讨论人机交互的作用。

表现层主要是使用 JSP 和 HTML 技术编写的浏览器页面,用于显示数据和接收用户输入的数据,为用户提供一种交互式操作的界面。例如语义 Web 应用程序要实现检索功能,则表现层需要有一个供用户输入检索请求的浏览器页面,然后浏览器将用户的请求通过 Jena 接口传递给逻辑层,最后浏览器页面要显示用户请求的结果。

3 语义 Web 应用程序实例分析

3.1 信息语义检索

信息语义检索^[12](information semantic retrieval)是一种基于信息语义理解的信息检索方式,是对传统的信息检索的一种语义化扩展。

信息语义检索能够对用户的检索请求进行语义层面的理解,同时对用户的检索请求进行语义扩展,从而使查询的结果更加符合用户的需求,也提高了查准率和查全率。

类比信息检索的过程,信息语义检索的基本过程如图 3 所示。

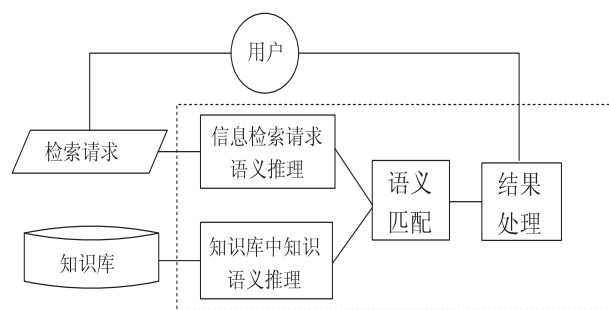


图 3 信息语义检索的基本过程

3.2 信息语义检索的语义 Web 应用程序

由于领域本体的限制,信息语义检索的语义 Web 应用程序只能局限于某一领域之内,因此将其应用于图书检索方面。数据层主要是构建一个图书方面的知识库,并集成到 Jena 框架中;逻辑层利用 Jena 自带的推理机和 ARQ 模块对知识库进行推理和访问,并利用倒排索引技术对结果进行排序;表现层主要通过 JSP 和 HTML 技术构建一个简单的查询页面,实现用户的输入输出。其中知识库的构建和自定义推理规则的加入是重点和难点,其余的都可通过已有的 Jena 模块实

现,因此文中将重点介绍知识库的构建和自定义推理规则的加入,并对实验结果进行分析。

3.2.1 知识库的构建

首先构建一个图书领域的本体,选择使用 Protégé^[13] 软件来构建本体。

Protégé 允许创建类、类的继承关系和实例,前两种功能用于确立语义关系,后一种用于检索输出。首先在 thing 类下创建子类 Book,然后给类 Book 添加属性 Name、Author、PublicationDate 和 SameAuthor,最后为类添加实例,即把所有书的信息作为实例输入到其中。文中选择将当当网传记类的图书作为实例,输入到所创建的本体中,生成一个本体文件 book. owl,以便于对检索结果进行对比分析。

所构建的图书本体的类和属性如图 4 所示。

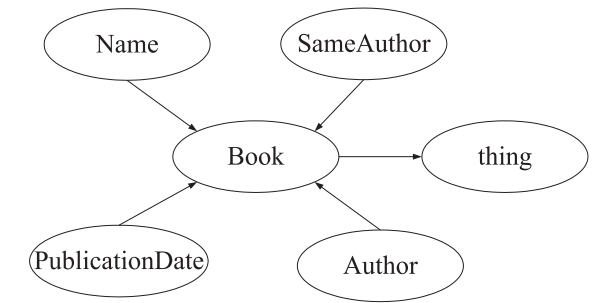


图 4 图书本体的类和属性

本体构建完成之后,就需要利用构建的本体文件来填充知识库。利用本体文件 book. owl 填充知识库的方法为:

```
Private void addDateFromFile() {
    InputStream inOntology = FileManager.get().open("book. owl");
    modelMem.read(inOntology, defaultNameSpace);
    inOntology.close();
}
```

3.2.2 自定义推理规则的加入

在图书检索过程中存在着不同的词却代表相同的意思,普通的关键字检索无法建立两者之间的联系,例如“爸爸”和“父亲”,因此需要给推理机加入一定的自定义推理规则,以便在检索时可以使检索结果更加全面。

假设每当输入有关“爸爸”、“父亲”、“dad”和“father”都可以得到相同的初步结果,则加入的推理规则如下所示。

```
(1) rules1 = "[ (? m Book; Name ? name) , strConcat (? name, ? lit); regex(? lit, " * 爸爸 * ") -> (? m sameIndividual ? p); ]"
```

```
(2) rules2 = "[ (? m Book; Name ? name) , strConcat (? name, ? lit); regex(? lit, " * 父亲 * ") -> (? m sameIndividual ? p); ]"
```

```
(3) rules3 = "[ (? m Book; Name ? name) , strConcat (? name, ? lit); regex(? lit, " * father * ") -> (? m sameIndividual ? p); ]"
```

```
(4) rules4 = "[ (? m Book; Name ? name) , strConcat (? name, ? lit); regex(? lit, " * dad * ") -> (? m sameIndividual ? p); ]"
```

以第一条规则为例,其中 rules1 为规则的名称,m 的书名为 name,利用内置规则函数 strConcat() 将 name 转换为文字项,然后利用 regex() 来匹配常规表达式“爸爸”,如果以上规则都符合,则 m 和 p 是相同的个体。

3.3 实验结果及分析

3.3.1 实验结果

根据语义 Web 应用程序开发方法,对建立的图书知识库进行检索,当使用“我的爸爸”作为关键词时,部分检索结果如表 2 所示。

表 2 检索结果表

序号	书名
1	包玉刚 我的爸爸
2	我的父亲;邓小平(套装全 3 册)
3	我的父亲毛泽东——我的父辈丛书
4	我的父亲牛书申
.....

同时每本书之后会显示更多的关于这本书的具体内容,包括书名、出版年月、作者、作者的其他书籍等一系列信息,如表 3 所示。

表 3 每条结果内容

属性	内容
书名	我的父亲;邓小平(套装全 3 册)
作者	邓林
出版年月	2004-08-01
作者其他书籍	邓小平-女儿心中的父亲

3.3.2 实验结果分析

对照表 2 可以看出,文中构建的语义 Web 应用程序在检索结果的查全率上相对于仅仅以关键字进行检索有了明显的提高,以关键字进行检索时只能检索到有“爸爸”关键词的图书,对于其他的各条信息因为关键字不对应所以不能检索到,而语义 Web 应用程序则可以检索到与“父亲”有关的图书。

对照表 3 可以看出,文中构建的语义 Web 应用程序不仅能检索出所需要的图书的详细信息,还可以检索出一些隐含的信息,例如可以通过 SameAuthor 检索到其他的与本书作者相同的书籍,使检索内容更加的全面。

可见语义 Web 应用程序相对于普通的应用程序

在信息的处理方面具有更大的优势,更加符合用户的需求。

4 结束语

随着语义 Web 的不断发展,语义 Web 势必取代 Web,语义 Web 应用程序在数据处理方面的优势将更加明显。语义 Web 应用程序将得到更加迅速的发展。文中基于 Web 应用程序给出了语义 Web 应用程序的定义和架构,并提出了语义 Web 应用程序开发方法,进而按照这种方法构建了一个语义 Web 应用程序,经过实验得到了良好的结果,证明了语义 Web 应用程序相对于普通的 Web 应用程序在信息处理方面具有更加良好的性能。但由于所选用的本体规模较小,实验结论说服力尚欠有力,有待进一步的推进。

参考文献:

[1] Berners-Lee T, Hendler J, Lassila O. The semantic Web[J]. Scientific American, 2001(5):34-43.

[2] Hebel J, Fisher M, Blace R, et al. Semantic Web programming[M]. Indianapolis, Indiana: Wiley Publishing, Inc, 2009.

[3] Rovani L, Baranovi M. Migrating Web-based applications into semantic Web[C]//Proc of the 28th international conference on information technology interface. [s. l.]: [s. n.], 2006: 159-164.

[4] Wu J, Ping L, Ge X, et al. Cloud storage as the infrastructure of cloud computing[C]. USA: IEEE, 2010.

[5] Qian L, Luo Z, Du Y, et al. Cloud computing: An overview[J]. Cloud computing, 2009, 5931: 626-631.

[6] 魏青松. 大规模分布式存储技术研究[D]. 成都: 电子科技大学, 2004.

[7] 王全修. 一种基于用户体验值的副本冗余度管理系统的设计与实现[D]. 成都: 电子科技大学, 2009.

[8] 张丽娜, 周润景. Matlab 与自适应神经网络模糊推理系统[M]. 北京: 电子工业出版社, 2010.

[9] Jang S H, Kim I K, Lee J S. Node availability-based congestion control model using fuzzy logic for computational grid[C]//Proc of FGCS. USA: IEEE, 2007.

[10] Xu J, Zhao M, Fortes J, et al. Autonomic resource management in virtualized data centers using fuzzy logic-based approaches[J]. Cluster computing, 2008, 11(3): 213-227.

[11] Xu J, Zhao M, Fortes J, et al. On the use of fuzzy modeling in virtualized data center management[C]//Proc of fourth international conference on autonomic computing. USA: IEEE, 2007.

[12] Zhou J, Yu K, Chou C, et al. A dynamic resource broker and

[4] Rovani L, Jagus T, Baranovi M. Defining categories and functionalities of semantic Web applications[J]. International journal of systems applications, engineering & development, 2011, 5(2): 245-253.

[5] Enrico M, Marta S, Sofia A, et al. Toward a new generation of semantic Web applications[J]. IEEE intelligent systems, 2008, 23(3): 20-28.

[6] 孙希进, 黄映辉. 语义 Web 应用程序: 性质与开发模型[J]. 计算机应用与软件, 2012, 29(11): 56-60.

[7] 刘 燕, 王作云. 基于 Web 的三层 B/S 模型的探讨与应用[J]. 物探化探计算技术, 2004, 26(3): 278-282.

[8] Apache Jena. Jena documentation overview[EB/OL]. 2011-11. <http://jena.apache.org/documentation/>.

[9] Aduna B V. User guide for Sesame[EB/OL]. 2011-09-23. <http://www.openrdf.org/doc/sesame/users/index.html>.

[10] W3C. SPARQL 1.1 query language[EB/OL]. 2012-11-08. <http://www.w3.org/TR/sparql11-query/>.

[11] 邝 砾, 邓水光, 李 莹, 等. 使用倒排索引优化面向组合的语义服务发现[J]. 软件学报, 2007, 18(8): 1911-1921.

[12] 樊 皓. 粗糙本体支持的信息语义检索研究[D]. 大连: 大连海事大学, 2012.

[13] Knublauch H, Fergerson R W, Noy N F, et al. The protégé OWL plugin: An open development environment for semantic Web applications[C]//Proceedings of the third international semantic Web conference. Hiroshima, Japan: [s. n.], 2004: 229-243.

(上接第 85 页)

fuzzy logic based scheduling algorithm in grid environment[J]. Adaptive and natural computing algorithms, 2007(1): 604-613.

[11] 王丹力, 赵 剡, 邱治平. Matlab 控制系统设计仿真应用[M]. 北京: 中国电力出版社, 2007.

[12] Wang W, Zhang H. A load balancing schedule strategy of web server cluster[C]//Proc of e-business and information system security. USA: IEEE, 2009.

[13] Zbigniew M, David B F. 如何求解问题-现代启发式方法[M]. 北京: 水利水电出版社, 2003.

[14] 代 栋, 周学海, 杨 峰, 等. 一种基于模糊推理的 Hadoop 异构机群自动配置工具[J]. 中国科学院研究生院学报, 2011, 28(6): 793-800.

[15] 张化光, 刘鑫蕊, 孙秋野. MATLAB/SIMULINK 实用教程[M]. 北京: 人民邮电出版社, 2009.

[16] 闻 新, 周 璐, 李东江, 等. Matlab 模糊逻辑工具箱的分析与应用[M]. 北京: 科学出版社, 2001.

[17] 刘正军. MATLAB 科学计算与可视化仿真[M]. 北京: 电子工业出版社, 2009.

[18] 王 琦, 徐氏蕴, 赵睿涛, 等. MATLAB 基础与应用实例集粹[M]. 北京: 人民邮电出版社, 2007.

语义Web应用程序开发方法及实例分析

作者：[李新龙](#)，[黄映辉](#)，[LI Xin-long](#)，[HUANG Ying-hui](#)
作者单位：[大连海事大学 信息科学技术学院, 辽宁 大连, 116026](#)
刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2013(12)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjfz201312021.aspx