

本体表示语言研究综述

岳 静 张自力

(西南师范大学人工智能与智能软件研究室 重庆 400715)

摘 要 随着 Semantic Web 的出现,本体作为人工智能领域中知识的核心已经越来越受到研究者的重视。特别是近十年来,涌现出许许多多本体表示语言,为本体的开发和研究提供了有力的帮助。本文主要介绍了传统的本体语言以及目前常用于万维网的本体语言的特点,并从表达和推理能力等方面对各种本体表示语言进行了分析和比较。在此基础上,总结出本体语言应用的一些规律,对于在开发工作中选择合适的本体语言具有一定的参考价值。

关键词 本体,本体表示语言,表达性,推理机制

A Survey on Ontology Specification Languages

YUE Jing ZHANG Zi-Li

(Laboratory of Artificial Intelligence and Intelligence Software, Southwest China Normal University, Chongqing 400715)

Abstract With the advent of the Semantic Web, ontologies as the core element in artificial intelligence have attracted the attentions of many researchers world wide. Last decade saw many ontology specification languages come into being, providing a strong support for the ontology development and research. This paper first highlights the features of traditional and Web-based ontology specification languages. These ontology specification languages are then compared by their expressiveness and inference capabilities. Based on the comparison, some guidelines are extracted for choosing appropriate ontology specification languages for different applications.

Keywords Ontology, Ontology specification languages, Expressiveness, Reasoning mechanism

1 引言

本体(ontology)是用于描述或表达某一领域知识的一组概念或术语,它可以用来组织知识库较高层次的知识抽象,也可以用来描述特定领域的知识^[2]。近年来,本体逐渐成为计算机领域研究的一个热门问题,尤其是随着 Semantic Web 的提出,本体作为 Web 信息在语义层次上共享和交换的基础,更是显示出其不可忽视的重要作用。目前,本体已经广泛应用于软件工程、自然语言理解、多问题求解、多 Agent 系统等许多领域,它已经成为知识表示、知识管理、知识共享、知识复用的主流技术之一,正成为自然语言处理、Web 信息检索、数据库和知识库的管理、异构数据集成、数字图书馆、GIS、多 Agent 系统、Semantic Web 等研究领域共同关心的一个核心^[2]。

在本体开发过程中,非常关键的一步是选择本体语言。过去十年中,产生了许多用于本体开发的语言,例如,Ontolingua、LOOM、OCML、Flogic 等等,Ontolingua 是这些语言中最具有代表性的一个。一些协议也被用来访问知识表示系统,比如 OKBC。最近几年,产生了一些应用于万维网的新的语言——XML、RDF、RDF Schema,然而,目前这些语言还不够成熟,仍然处于发展阶段,同时还出现了另外一些新的本体表示语言:SHOE、XOL、OIL、DAML+OIL、OWL,它们都是基于前面各种本体表示语言的。

既然出现了这么多的本体表示语言,那么在实际开发工作中,我们应该选择那种语言比较合理呢?实际上,在以往的开发过程中,本体开发工作者往往是根据自己的偏好和经验

来进行选择,而忽略了实际应用的需要和本体语言本身的特点。这样,对于语言的选择带有很大的主观色彩,甚至采用的语言与实际需求有较大差别,大大影响了开发工作的效率。如果开发者对于各种语言的特点了如指掌,并且在开发工作中根据实际需要选择合适的本体语言,那么就能起到事半功倍的效果。

本文正是以此为目的,介绍了传统的本体语言以及基于 Web 的本体语言,分析并比较了各种本体语言的特点,特别是着眼于从表达和推理能力两个方面,比较了各种语言的差异,希望对研究者的开发工作有一定的参考价值。本文在第 2 部分,详细介绍了传统的本体语言以及目前基于万维网的本体语言,第 3 部分对各种语言的主要元件和推理机制等方面作了分析和比较。最后,在分析和比较的基础上总结出一些规律,并提出了在本体语言领域尚待解决的一些问题。

2 本体表示语言

在这部分,我们将依次分析已被广泛使用的本体表示语言(Ontolingua、OKBC、LOOM、OCML、Flogic)、产生于 Internet 环境下并由 W3C 推荐的语言(XML、RDF、RDFS)以及一些最新的本体表示语言(SHOE、XOL、DAML+OIL、OWL)。

2.1 传统本体表示语言

Ontolingua^[1,2]是一种基于 KIF(Knowledge Interchange Format)和 FO(Frame Ontology)的语言,斯坦福大学人工智能实验室的 Ontolingua 服务器就是采用 Ontolingua 作为其本体表示语言。

KIF 是由斯坦福大学人工智能实验室专家在研究本体时

提出的一种基于谓词演算的形式化语言,主要功能包括解决知识表示语言的异质性,定义对象、函数和关系,基于一阶谓词逻辑,并含有一个前缀符号,另外,它还可以表示元知识和非单调推理规则,具有描述性语义,成为采用不同知识表示的程序之间通讯的桥梁。

由于 KIF 其实是一种互换格式,因此从本质上讲,用 KIF 来详细说明本体是非常繁琐的。而建立在 KIF 基础之上的 FO 是一种知识表示本体,它能够以框架的样式详细描述本体,能够提供诸如 class, instance, subclass-of, instance-of 等项,但它不能表示公理。因此,Ontolingua 将 KIF 表达式引入基于 FO 的定义内。

一般来说,Ontolingua 可以用以下三种方式建立本体:(1)仅使用 FO 词汇(不能表示公理);(2)使用 KIF 表达式;(3)同时使用上述两种方式。无论采用哪种表示方法,Ontolingua 都包括 3 个组成部分:定义头部分、用自然语言描述的非形式化定义部分、用 KIF 或者 FO 定义的转换器。这些 Ontolingua 转换器可将由 Ontolingua 表示的本体转换为 LOOM 等语言。

目前,Ontolingua 已经开发出一个推理引擎。如果用户想要开发一个专用的推理引擎,就必须使用 OKBC 应用程序接口。

OKBC^[1](Open Knowledge Base Connectivity)协议,曾经被称为通用框架协议,它具体描述了一个协议,而非一种语言。这个协议对其底层的知识表示系统作了假定,对于支持知识共享的语言是一个有力的补充。

GFP 知识模型是位于 OKBC 底层的隐式表示方式,它能够用以对象为中心的方法来表示知识,还提供了一组具有代表性的构件:常量、框架、槽、面、类、个体和知识库,这些构件都常常出现在框架表示系统中。同时,它还提供了一个问答界面,以使用 OKBC 协议访问知识库,另外,还定义了一些用类 Lisp 语言编写的函数,这些函数可以描述在知识库内部所进行的一些复杂的操作。

目前,已经开发出完全符合 OKBC 协议的 OKBC-Ontology,并已应用到 Ontolingua 中。

OCML^[1](Operational Conceptual Modeling Language)是一种基于框架的语言,它能够表示诸如关系、函数、规则、类和实例等项。为了使这种语言更为有效地执行,它还另外增加了一些逻辑机制以增强推理的有效性。它也提供了一个问答界面,而且可以将它作为阐明事实、检验 OCML 模型内容的一种机制。

在开发 OCML 的过程中还考虑到许多实际因素,其中一个就是与标准(比如 Ontolingua)相容,而且它还提供了理论证明和函数评价功能,因而 OCML 也可以被看作是一种“实用的 Ontolingua”。

Flogic^[1,2](Frame Logic)综合了基于框架的语言和一阶谓词演算,它能够清楚明白地说明面向对象和基于框架语言的大多数结构方面,例如,对象属性、合成对象、继承、多态、查询方法、封装等等。在推理演绎方面,FLogic 有一个理论模型和一个完整有效的基于决策的证明机制。

FLogic 的应用领域很广,从面向对象、推演数据库一直到本体。如果将它与其他一些专门的逻辑(HiLog,事务 Logic)结合起来,还可以利用本体中的信息改进推理机制。

LOOM^[1,2]是一种高级编程语言,由美国南加州大学信息科学学院设计并实现,主要是用来构建专家系统和其他一

些智能系统。它是由 KL-ONE 衍生而来的,基于描述逻辑,这种方法实现了基于规则和基于框架模式的紧密结合。

LOOM 提供了明确定义而且表达力较强的模型描述语言,能描述定义、规则、事实和缺省规则等;提供了有效的推理机制,能利用前向推理、语义一致化和面向对象的真值维护等技术实现推理;提供了编程范例、产生式规则范例和面向对象范例等,并能方便地把范例集成到应用中。LOOM 还提供了知识库服务功能以方便用户对知识库的操作和维护。目前,南加州大学又推出了 LOOM 的第二代 PowerLoom,它继承了 LOOM 的知识表示方法,并且它的推理引擎更为有效。

另外,非常重要的一点是,集中采用描述逻辑的方法来建立本体,它与上述各种基于框架的语言的方法是不同的。用这种方法所下的定义可以产生一个有效的分类器,并用一组限制条件详细描述各个概念。

2.2 Web 标准和 W3C 推荐本体语言

XML^[3,7](eXtended Markup Language)是从 SGML(Standard General Markup Language)发展而来的,由万维网联盟(W3C)的 XML 工作小组开发,现在它几乎已经成为了一种标准。

XML 作为本体表示语言的基础,它主要的优点在于:

- 用 DTD(Document Type Definition)定义通用语法规范;
- 用户能够理解包含在 XML 内的信息;
- 由于 XML 可以嵌入网页中,因此它可以表示包含在多个页面中的分布式信息;
- 允许缺省 XML 标记符之间的信息结构,这样,要在文档内部发现本体成分就很难;
- 有许多现成的分析和处理 XML 文件的标准工具,但没有推理工具。为了用基于 XML 的语言进行推理,必须开发出新的工具。

XML 本身对于本体的规范并没有什么专门的特性,它只是提供了一个简单而有效的方式来详细指明本体表示语言的语法(正因为如此,我们并没有在第 3 部分将 XML 列入比较的范畴)。

RDF^[1,3](Resource Description Framework)是一种基于 XML 的语言,W3C 为了创建描述网络资源的元数据而开发出这种语言。RDF 与 XML 之间的关系非常紧密,事实上,它们是互为补充的,RDF 的一个目的就是以一种标准的、可共用的方式来描述基于 XML 数据的语义。

RDF 数据模型(基于语义网络)包含三种类型:资源(主语)——由 WWW 地址表示的实体;属性(谓语)——定义了用来描述资源的具体特征、属性和关系;声明(宾语)——为某个资源的一个属性赋值。

RDF 有两个重要的贡献:①为创建本体提供了标准语法;②一组建立诸如 instance-of、subclass-of 关系的标准集。

RDF Schema^[3](RDFS)是一种定义 RDF 模型的语言,它其实就是一组标准 RDF 资源和属性的集合。RDFS 为 RDF 提供了基本的类型模式,可以定义对象、类和属性。RDFS 的数据模型与面向对象编程中的数据模型是一样的,它可以定义属性和资源之间的关系,因而用户能够创建自己的词库。

2.3 基于 Web 的本体表示语言

XOL^[1,2](XML-Based Ontology Exchange Language)是由美国 SRI 研究院 AI 中心开发的本体交换语言,为交换本体定义提供了一种格式,因此,它并不能用来开发本体,但却

可以作为在不同数据库系统、本体开发工具和应用程序之间转换本体的中间语言。

XOL 用 XML 语法定义了一个 OKBC 子集,称为 OKBC-Lite。相比而言,OKBC 为访问基于框架的系统定义了协议,而 XOL 更适合通过万维网交换不同系统之间的信息。不过,XOL 有一个主要的障碍——只能定义类、槽、方面,而不能定义框架。

目前,已经有许多 XML 编辑工具可以生成 XOL 文件。

SHOE^[1,2] (Simple HTML Ontology Extension) 最初只是 HTML 的一个扩充,其目的是在 HTML 或其他 WWW 文件中加入机器可理解的语义,使得 agent 能从网页和文件中收集有用的信息,以改善搜索机制和信息收集。要达到这个目标需要经历两个阶段:(1)定义一个本体来描述正确的对象类别以及对象之间的关系;(2)给 HTML 页面加上描述其本身以及其它网页的注释。在 SHOE 中,本体是一个 ISA 的层次关系,包括类、类之间的关系、推理规则。除此之外,SHOE 的另一个重要特征是,它可以声明信息。

DAML+OIL^[1,6] 其实本身是两种语言,DAML 和 OIL,前者是由美国开发,后者是由欧洲开发的,两队开发者发现对方的研究成果以后,决定将两种语言联合起来,也就是现在的 DAML+OIL 语言。最近还开发出 DAML Service,这是一组用 DAML+OIL 表示的本体,其目的是解决 Web 服务中的语义问题。

OIL (Ontology Interchange Language) 是 OnTO-Knowledge 项目组设计的本体语言。OIL 的定义融合了三种范例:基于框架的建模、基于描述逻辑的形式化语义和基于 XML 的语法。OIL 与 XOL 联系非常紧密,甚至可以把它看作是 XOL 的扩展。OIL 中的本体是一个由许多模块组成的三层结构:对象级(处理实例),本体定义(包括了本体的定义)、本体容器(包括了本体的信息,例如该本体的创建者)。使用 OIL 定义的本体能够被映射到描述逻辑的公理,因此能够借助已有的系统执行可靠性和完整性推理。OIL 已经成功地应用于多个领域,如知识管理、电子商务等。

DAML+OIL 本身是由 RDF(S) 扩展而来的。DAML+OIL 中的本体用 XML 编写,也可以用 RDF 中的三元式编写。有许多工具、系统和应用程序使用了 DAML+OIL 本体。不过,其中许多将被改写成 OWL 语言,因为这种语言将会取代 DAML+OIL。

OWL^[3~5] (Web Ontology Language) 是专门为 Semantic Web 而开发的一种本体语言,与 DAML+OIL 不同的是,OWL 是由 W3C 发起的,它的第一个官方版本出现于 2003 年。

OWL 有 3 个层次的语言:OWL Lite,OWL DL,OWL Full。这 3 种语言的表达性依次增强,较高的层次包含了较低的层次。

OWL Lite 可以定义类、属性以及类的实例,它是一种比较简单的语言,适用于只需要分类层次和简单的限制条件的用户。例如,当它限制集合时,只能给集合赋值为 0 或为 1。

OWL DL 是 OWL Lite 的扩充,适用于那些希望在保持计算完整性和可判定性的情况下获得最强的表达性的用户。OWL DL 包含了 OWL 语言的所有构件,但它只能在某种条件下使用,只能定义一个集合的属性而不是一个个体的属性。

OWL Full 又比 OWL DL 更高一级,它不仅可以定义一个集合的属性也可以定义某个个体的属性,它适应于那些只需要最大的表达性和 RDF 语法自由度而不需要任何计算保障的用户。例如,OWL Full 中的类既可以被看作是一组个体的集合,也可以将它本身看成是一个个体。

3 分析与比较

上述各种语言各有各的特点,有的易于表达但是推理机制很差,有的能够实现有效的推理却是以牺牲表达性为代价的,因而我们很难找到一个最优的语言。在实际应用中对表达性和推理性有着不同的需求,某些语言可能比其他语言更适合某种特定的应用领域。当用户为某个应用系统开发本体时,不仅有必要分析该系统对于知识表示和推理的需求,而且应该研究这些语言所提供的知识表示和推理的能力,这样,本体开发者才能避免盲目地选择本体语言。

下面,我们将以表格的形式比较上述本体语言在本体语言主要元件、分类系统、关系和函数、推理机制等几方面的不同。在表中,“+”表示该语言支持这项特性,“-”表示该语言不支持这项特性,“+/-”表示目前不支持但加以改进后可支持这项特性,“ND”表示没有强制要求但是可以实现的特性,“?”表示目前未知的特性。

3.1 主要元件

表 1 对上述各本体语言的主要元件作了大致描述。

从表中我们可以看出,几乎所有的语言都定义了概念、多元关系和实例。在 OKBC 和 FLogic 这两种基于框架的语言中,关系可以用框架来表示,但它并不是这种语言所提供的一种特殊元件。基于 Web 的语言不能定义函数、过程和公理,但是某些特定格式的公理除外,例如,在 SHOE 中可以定义推演规则。表 1 最后一行分析了各语言的形式化语义:一些基于 Web 的语言例如,SHOE、RDF(S) 和 DAML+OIL、OWL 缺少这项,而传统语言和 XOL 提供了这项功能。

表 1 主要元件的比较

	Onto	OKBC	OCML	LOOM	Flogic	XOL	SHOE	RDF(S)	DAML+OIL	OWL
概念	+	+	+	+	+	+	+	+	+	+
多元关系	+	+/-	+	+	+/-	-	+	+	+	+
函数	+	+/-	+	+	+/-	-	-	-	+	-
过程	+	+	+	+	-	-	-	-	-	-
实例	+	+	+	+	+	+	+	+	ND	+
公理	+	+/-	+	+	+	-	-	-	ND	-
产生式规则	-	-	+	+	-	-	-	-	ND	ND
形式化语义	+	+	+	+	+	+	-	-	-	-

3.2 分类系统

如表 2 所示,在分类系统的各项中,只有 subclass of 这一

项是所有语言都定义好的,而只有 Ontolingua 和 LOOM 才定义了其余所有项(not subclass of 除外)。不过,这些项都可以在其余语言中定义成关系,因此在这里对它们并没有特别的

要求。DAML+OIL 允许定义 not subclass of 这一项,因而它也可以定义 disjont decompositions 这项。

表 2 分类系统的比较

分类系统	Onto	OKBC	OCML	LOOM	Flogic	XOL	SHOE	RDF(S)	DAML+OIL	OWL
Subclass of	+	+	+	+	+	+	+	+	+	+
子集分割	+	-	+/-	+	+/-	-	-	-	-	-
分解	+	-	+/-	+	+/-	-	-	-	+/-	+
Not subclass of	+/-	-	-	+/-	-	-	-	-	+	-

3.3 关系和函数

关系是本体中非常重要的一个元件,因此几乎所有的本

体语言都支持这项,但并不是所有语言都实现了需要的特性。某些语言中就不包括函数这项。具体情况如表 3 所示。

表 3 关系和函数的比较

	Onto	OKBC	OCML	LOOM	Flogic	XOL	SHOE	RDF(S)	DAML+OIL	OWL
关系函数	+	+	-	+	+	-	-	-	+	+
一元关系	+	+	+	+	-	-	+	-	+	+
二元关系	+	+	+	+	-	+	+	+	+	+
N 元关系/函数	+	+/-	+	+	+/-	-	+	+	+/-	-
类型限制	+	+	+	+	+	-	+	+	+	+
完整性限制	+	+	+	+	+	-	-	-	-	-
实用定义	-	-	+	+	+	-	-	-	-	?

3.4 推理机制

几乎所有语言中都存在着明显的知识表示和推理机制的差别,但 OCML 除外。例如,Ontolingua 是上述这些语言中表达性最强的,但它并未实现推理引擎。OCML 可以定义一些与推理相关的特性,比如,规则既可以定义成前向的也可以定义成后向的,这样也就明确地定义了链。

只有 FLogic 和 OIL 的推理引擎比较完整有效,这一点正是我们所期望的,但是另一方面,这两种语言的表达性却非常差。

基于描述逻辑的语言(LOOM 和 OIL)实现了自动分类。

总体说来,语言开发者都没有处理好出错处理机制这个问题,FLogic 是唯一一个能处理错误的语言。不过,其他语言在这方面都作了一些相应的改进工作。

除 XOL 之外的所有语言都支持单继承和多继承,不过多继承中的冲突仍然没有解决。

所有传统本体语言都有强制检查这项功能,OKBC 中所有操作都必须进行强制检查,但是也可以设定参数甚至屏蔽到这项功能。SHOE 中没有强制检查功能,因为网络中的错误非常多,要完全解决这些错误是非常困难的。不过,在必要时还是会进行典型的强制检查。

表 4 推理机制的比较

	Onto	OKBC	OCML	LOOM	Flogic	XOL	SHOE	RDF(S)	DAML+OIL	OWL
有效	-	-	+	+	+	-	-	-	+	+
完整	-	-	-	-	+	-	-	-	+	+
自动分类	-	-	-	+	-	-	-	-	+	-
出错处理	-	-	-	-	+	-	-	-	-	+/-
单调性	+	+	+	+	+	ND	+	ND	+	+
非单调性	+/-	+	+/-	+	+	ND	-	ND	-	-
单个继承	+	+	+	+	+	ND	+	+	+	+
多个继承	+	+	+	+	+	ND	+	+	+	+
过程执行	+	+	+	+	-	-	-	-	-	-
强制检查	+	+	+	+	+	-	-	-	-	+
前向	+	+	+	+	+	-	ND	-	-	-
后向	+	+	+	+	+	-	ND	-	-	-

总结及展望 本文主要介绍了各种本体语言的特点,并从本体语言主要元件、分类、函数、推理机制等方面分析和比较了各种语言。从以上分析中我们可以大致总结出各种语言在表达和推理能力上的差别,如表 5 所示。从表达性这个方

面来看,传统的本体语言表达性较强,它们不仅可以表示概念、概念类别、概念之间的关系等基本的元件,还可以表示形式化公理、函数、规则、过程以及其他一些元件;而本体标记语言在这方面就比较欠缺。

表 5 表达性和推理能力的比较

	Onto	OCML	LOOM	Flogic	XOL	SHOE	RDF(S)	DAML+OIL	OWL
表达能力	强	中	中	中	中	中	中	中	中
推理引擎	无	中	强	强	无	中	无	弱	较弱

到目前为止,虽然还没有一种最优的表示语言,但是可以根据实际应用的需要选择最适合的语言。由于某一种语言不可能同时兼备表达性和推理性,只能在两者之间达到一种平衡,但是不同的应用领域对知识表示和推理的需求不同,某些语言可能比其他语言更为合适,因而,在以后的研究中,我们的一个主要任务就是找出影响创建本体的关键因素,这样才能从众多的语言中选取最合适的。

尽管目前还没有评价本体语言的统一的标准,但是我们仍然可以从上述的分析比较中总结出一些规律。

- 需要交换不同应用程序之间的本体时,采用基于 Web 的语言比较适合,因为这些语言都是基于 XML 的,因而比较容易被用户理解和管理。除此之外,这些语言还有一个很大的优势,它们受到各个科研组织的强力支持,因此就有更多可利用的编辑工具。

- 如果对于表达性的要求较高,则使用传统本体语言比较合适,不过,如果只是将本体看作一些类别,那么基于 Web 的语言也可以胜任。

- 由于基于 XML 的语言没有提供推理引擎,因此如果要在 agent 内实现推理,它们并不适合,而一些传统语言不仅提供了推理引擎而且还可以将它们转换成其他可计算的语言。

- 在电子商务领域中,本体通常是用来表示电子商务平台中所提供的产品和服务,并分门别类以便用户浏览。因而对于表达的要求并不高,而对推理的需求就要高一些,例如,自动分类对于自动地将这些产品和服务归类是非常有用的,因此,在这里使用基于描述逻辑的语言是很有帮助的。

- 创建顶层本体要求很强的表达性,对于推理没有太高的要求,因此,顶层本体通常是用基于描述逻辑的语言来表示的,比如 LOOM。Cyc 工程是通用本体中一个最典型的例子,它的知识库采用 CycL 语言表示,这种语言是基于框架和一阶逻辑的。

上述指导原则对于开发者选择适当的本体语言是大有帮助的,但实际工作中情况复杂多变,应该根据实际需要,灵活地进行选择,这样才能提高开发和研究工作的效率。

目前,国际上(尤其是欧洲)在本体研究领域处于领先地位,

推出了一系列的本体开发方法、开发工具和相关标准,而国内在这方面才刚刚起步,研究成果还很少。不过,总体说来,本体技术还不是十分成熟,仅仅从本体表示语言这个小小的领域来看,还有很多问题尚待解决。比如,由于本体语言在表达和推理能力中存在或多或少的缺陷,在不能改变语言本身特性的情况下,我们可以考虑从语言外部来弥补这些缺陷,例如,本体标记语言不能表示形式化公理,但是可以在这些语言基础之上建立逻辑层,这样,就能具有语言本身不具有的逻辑特征。另外,除了以往开发工作的经验和规律,找到影响本体开发的关键因素也是一个需要解决的问题。总之,本体是一个具有很大机遇和挑战性的研究领域,需要广大科学工作者的共同努力。

参考文献

- 1 Corcho O, Gómez-Pérez A. A Roadmap to Ontology Specification Languages. EKA W 2000, LNAI1937, 2000, 80~96
- 2 顾芳,曹存根. 知识工程中的本体研究现状与存在问题. 计算机科学, 2004, 31(10): 1~10
- 3 Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American, 2001
- 4 McGuinness D, Harmelen F. OWL Web Ontology Language Overview. Feb. 2004. <http://www.w3.org/TR/owl-features>
- 5 Smith M, Wely C, McGuinness D. OWL Web Ontology Language Guide. Oct. 2004. <http://www.w3.org/TR/owl-guide>
- 6 <http://www.daml.org/2000/10/daml-ont.html>. Oct. 2004
- 7 Bray T, Paoli J, et al. Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation, Feb. 2004. <http://www.w3.org/TR/REC-xml>.
- 8 Brickley D, Guha R V. Resource Description Framework (RDF) Schema Specification. W3C Proposed Recommendation. Oct. 2004. <http://www.w3.org/TR/PR-rdf-schema>
- 9 Fensel D, et al. On-To-Knowledge: Ontology-Based Tools for Knowledge Management, 2001
- 10 Lim S Y, Song M H, Lee S J. The Construction of Domain Ontology and Its Application to Document Retrieval. ADVIS 2004, 117~127
- 11 Gómez-Pérez A, Fernandez-Lopez, Corcho O. Ontology Engineering. Springer published, 2003
- 12 Corcho O, Gómez-Pérez A. Evaluating knowledge representation and reasoning capabilities of ontology specification languages. In: Proc. of the ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin, 2000
- 13 Su Xiaomeng, Ilebrette L. A Comparative Study of Ontology Languages and Tools. In: Proc. of Conf. on Advanced Information System Engineering (CAISE' 02), Toronto, Canada, 2002, Springer, 2002
- 14 杜小勇,李曼,王大治. 语言 Web 与本体研究综述. 计算机应用, 2004(10): 14~20

(上接第 134 页)

开发环境提供的 Web 服务客户端测试工具 UTC 进行测试,测试结果表明用户所需的 Element 类型数据已经通过 Internet 传递到本地。

Web Service 客户端负责发送请求,调用服务器端的服务,从服务器端取得按照客户端 XML 映射文档模板生成的 XML 文档,取得这个数据以后,客户端就可以根据本地的数据库模式,解析 XML 文档,将数据添加到本地数据库,从而完成数据交换。

最终在客户端得到 Web 服务响应端返回的 Element 类型的数据文档,解析该包含共享数据的文档,将数据添加到本地数据库相应的数据表中。

结论 本文在深入研究现有数据交换技术的基础上,针对其存在的不足,结合现代企业对数据交换的现实需求,提出了一种全新的、基于多层 B/S 模式的、安全易用的数据交换原型系统。在此基础上,基于 J2EE 平台下实现了该原型系统,并在实际应用中得到了验证。该系统以前端 JavaBean 包装 Session Bean,完成数据转换业务逻辑,有效地解决了难以构建满足用户所需的、结构清晰的 XML 文档的难题;以前端

JavaBean 作为 Web 服务的功能实现,生成 Web 服务,达到真正意义上跨平台,无障碍地穿透企业防火墙;以符合 EJB2.0 的 CMP 实体 Beans 类化关系数据表,使得业务逻辑的实现与后端具体数据库管理系统相互独立,降低设计的复杂度,提高系统整体的性能及安全性;保证了从源端系统提取、转换后返回的数据信息符合本地数据库模式,极大地方便了本地数据文档的解析,具有较强的推广应用价值。

参考文献

- 1 Williams K. Programming Professional XML Databases. Wrox press, 2000. 701~722
- 2 Shanmugasundaram J. Bridging relational technology and XML [Dissertation]. Dissertation Abstracts International, Volume: 62-07, Section: B; 3277
- 3 Hojabri B. Algorithms for generating XML documents from hierarchical views of relational databases; [Dissertation]. Masters Abstracts International, Volume: 41-04; 1107
- 4 李军怀,张景,等. 基于 XML 的企业异构数据集成方法研究. 计算机工程, 2002, 28(9): 63
- 5 聂培尧,魏振刚. 一种基于 XML 的数据集成系统及其实现. 计算机应用, 2002. 22~29