

开源主题图 Ontopia 引擎工作机制研究*

李清茂

(阿坝师范高等专科学校 计算机科学系,四川 汶川 623000)

【摘要】在概要分析OKS知识组件和Ontopia主题图引擎体系结构的基础上,深入分析了引擎API的主要开发包、核心接口之间的构成关系以及实现形式。针对Ontopia引擎接口对主题图创建、存储、数据交换和应用功能开发等方面的支持,并结合具体代码实现方法对引擎工作原理进行了全面分析。

【关键词】主题图;开源主题图引擎;Ontopia主题图引擎;知识组件

【中图分类号】G203 **【文献标识码】**A **【文章编号】**1673-1891(2010)02-0040-05

1 引言

Topic Maps作为一种能描述信息资源知识结构,定位某一知识概念所在资源位置,表示知识概念间关系的一种知识组织工具,自2000年成为ISO/IEC13250^[1]国际标准以来,先后推出了XTM1.0(ISO13250:2002)^[2]、TMQL^[3](ISO 18048: 2003)、TMCL(ISO 19756: 2008)^[4]、XTM2.0^[5]等多项国际标准。与此同时美国和欧洲的一些公司着手开发主题图引擎来支持主题图创建、存储和导航等功能,从而为广大主题图应用程序开发人员提供接口。

目前常用的主题图引擎有TM4J^[6-7]、tinyTIM^[8]、XTM4XMLDB^[9]和OKS^[10]等,然而仅有挪威Ontopia公司的OKS(Ontopia Knowledge Suite)获得了成功。尽管在世界范围内,主题图技术有很多忠实的支持者,但是作为知识组织技术的主题图却很难在挪威之外的其他地方广泛推广。由于OKS原本属于Ontopia公司的商业产品,除了商业用户之外很难有人能够真正了解OKS组件内部引擎API的接口实现细节。为了在世界范围内推广主题图技术,Ontopia公司于2009年7月19日发布OKS 5.0组件,并同时宣布开源代码,因此人们才有机会了解到OKS组件中的Ontopia Topic Map Engine API工作原理和实现细节。Ontopia公司这一战略转变必将对主题图应用技术迅速普及和推广产生深远影响。

2 OKS知识组件概要

OKS知识组件是目前最先进,功能最齐全的商业化主题图构建工具套件,主要用于创建、维护和部署主题图应用程序。

Ontopia主题图引擎位于引擎层的中心,也是整个OKS组件的核心。它有自己独立的API接口,但同时也支持业界TMAPI接口标准。在主题图应用

程序中主题图表示、生成、存储以及与其它数据格式转换都必须在引擎API的支持下才能完成。除此之外,Ontopia引擎还集成了支持tolog查询语言、Ontopia schema语言、全文本查询(Full-Text Search)等功能,用以构建基于主题图的Web应用框架。OKS存储层支持基于内存(In Memory)和基于关系数据库(RDBMS Backend)的主题图后台实现,但其具体实现方法都必须依赖引擎API。在应用层中OKS应用层主要提供了主题图导航框架(Ontopia Navigator Framework)和Web编辑框架(Web Editor Framework),在引擎API的支持下为程序员提供面向Web的主题图应用支持。

3 Ontopia引擎结构及运行机制

3.1 Ontopia引擎概述

Ontopia主题图引擎不仅在OKS组件中处于核心地位,同样在支持主题图应用程序开发方面也非常重要。Ontopia主题图引擎由核心引擎API和主题图工具包组成^[12]。在主题图应用程序中对主题图进行的修改、访问、存储、数据交换等操作都可以通过Ontopia引擎API来实现。利用它还可以实现主题图数据存储无关性,即无论数据存储在内、关系数据库还是以XML文件形式存在都不会影响应用程序的运行。工具包是为特定主题图应用目的而设计的,用于实现诸如主题关联遍历、主题图合并、信息查询等功能。

3.2 Ontopia引擎API基本组成

Ontopia引擎(Topic Map Engine)API提供了多个开放源代码的JAVA开发包,用以对主题图开发的全面支持,包括主题图表示、索引和以XML文档或关系数据库方式存储的主题图数据进行修改和访问,其核心是一组主题图JAVA API包。这组核心

收稿日期:2010-03-19

*基本项目:四川省教育厅科研基金“主题地图(Topic Maps)技术在阿坝州旅游文化信息资源组织中的应用研究”课题部分研究成果(项目编号:07ZC001)。

作者简介:李清茂(1973-),男,四川攀枝花人,副教授,管理学硕士,研究方向:信息管理技术与方法。

API包即是图2所示的 Ontopia Topic Map Engine Core API,应用程序利用它可访问主题图,而无需考虑数据是在内存还是位于数据库中,甚至虚拟视图都不影响应用程序对主题图的访问。

Ontopia引擎API提供的JAVA开发包主要有^[12]:

(1)net.ontopia.topicmaps.core包。该包定义了表示主题图对象的核心接口,包括主题图(Topic Maps)、主题(Topic)、关联(Association)、资源指引(Occurrence)等。由于该包在整个主题图应用程序开发中处于核心地位,因此只包含主题图及其构成元素的抽象接口定义,不包括接口的具体实例,具体实例需要根据主题图数据存储后台不同分别定义。如果采用内存实例则核心接口实例位于net.ontopia.topicmaps.impl.basic包中;如果采用数据库后台存储则核心接口位于net.ontopia.topicmaps.impl.rdbms包中。这两个包中定义的全部集成核心接口中定义的成员变量和操作方法,只需定义具体的对象即可实现接口的实例化。

(2)net.ontopia.topicmaps.core.index包。该包定义的全是主题图通用索引接口,这些接口提供了一个在主题图中查询信息的API。索引接口提供的是

最直接和最有效率的查询方法,例如给定主题类型(Topic Type)检索所包含的所有主题,而不管该主题是用作关联类型或其他。该包中仅包含接口,其实例并不包含在其中,利用引擎API时每个指定的实例可利用其接口。因此,程序人员仅关心与存储结构独立的API。

(3)net.ontopia.topicmaps.utils包。该包主要是用于处理主题图的实用工具包,如TopicStringifiers类用于抽取主题名字符,MergeUtils类用于主题图合并,AssociationWalker用于支持主图遍历等。

(4)net.ontopia.topicmaps.xml包。该包中定义的类主要用于对XTM主题图提供交换和导出支持等操作。

(5)net.ontopia.topicmaps.entry包。该包中仅定义三个高级接口:TopicMapReferenceIF、Topic Map Repository IF、TopicMapSourceIF。主要用于在在文件系统中定位主题图以提供持久存储和管理功能。

3.3 Ontopia引擎核心接口结构

核心包net.ontopia.topicmaps.core在整个引擎API中作用非常重要,其核心接口体系结构如图1所示。

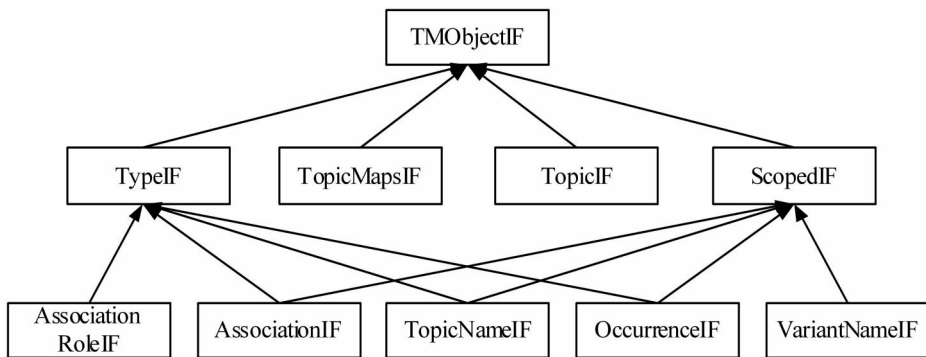


图1 核心接口体系结构图

从图1可知,TMOBJECTIF接口是核心API的基接口,引擎中所有其它主题图对象都从它派生而来,因此是一个抽象接口,用于表示所有子接口的公共功能。该接口有四个直接子接口:TypeIF、TopicMapsIF、TopicIF、ScpoedIF。其中,TypeIF和ScpoedIF接口与TMOBJECTIF接口一样都是抽象接口。下面分别对这些接口进行分析:

(1)TypeIF:它的直接父接口是TMOBJECTIF,其直接子接口有AssociationIF,AssociationRoleIF,OccurrenceIF,TopicNameIF。该接口由代表资源指引(Occurrence)、关联(Association)、主题名称(TopicName)和关联角色(AssociationRole)的主题图对象来实现。该接口定义了两个抽象方法:TopicIF getType()和void setType(TopicIF type)。分别用于

获取和设置主题对象类型。

(2)ScpoedIF:它的直接父接口是TMOBJECTIF,其直接子接口有AssociationIF,OccurrenceIF,TopicNameIF,VariantNameIF。该接口由具有主题范围约束(Scope)的主题图对象实现。

从图1可知,AssociationIF,OccurrenceIF,TopicNameIF这三个接口都有两个共同的父接口TypeIF和ScpoedIF,因此在具体的主题图存储后台实例化这三个接口时,应全部实现TypeIF和ScpoedIF接口中定义的所有方法。这样的设计架构有助于实现平台无关性,即无论主题图采用内存还是关系数据库作为存储后台,其操作方法都是一样的。

(3)TopicMapIF:它的直接父接口是TMOBJECTIF,用于表示整个主题图对象,核心引擎为其定义了

三类方法。

与标识符相关的方法: getObjectByItemIdentifier (LocatorIF locator)用于获取指定标识符的主题图对象; getTopicBySubjectIdentifier (LocatorIF locator)用于获取指定议题标识符的主题对象; getTopicBySubjectLocator (LocatorIF locator)用于获取表示可寻址议题的主题对象。

与主题元素相关的方法: getAssociations()获取主题图中所有的关联; getTopics()获取主题图中所有的主题; getIndex (java.lang.String name)用于获取一个名字索引; clear()通过清除所有的主题和关联来清除主题图; getObjectById (java.lang.String object_id)通过指定指定对象 id 来获取主题图对象。

与事务相关的方法: getBuilder()用于获取主题图构造器相关事务; getStore()用于获取主题图事务相关的存储对象。

(4)主题元素接口:在引擎核心 API 中主题图的基本元素分别用几个接口来表示。其中 TopicIF 接口由主题图模型中表示主题的对象来实例化, AssociationIF 由主题图模型中表示关联的对象来实例化, OccurrenceIF 接口由主题图模型中表示资源指引的对象来实例化; TopicNameIF 接口由代表主题名称的对象来实例化; VariantNameIF 接口由代表主题变量名的对象来实例化。

3.4 Ontopia 引擎核心其它接口

在 Ontopia 引擎 API 中,除图 3 所示具有直接继承关系的核心接口外,在 net.ontopia.topicmaps.core 包中还有一些与主题图构建、存储以及输入输出相关的接口,这些接口对于利用引擎 API 构建应用程序也至关重要。这些接口的功能如下:

(1)TopicMapBuilderIF:主题图对象结构创建辅助器,利用该接口的构造器可以创建主题图对象。该接口中定义了如下这些方法:

getTopicMap():返回主题图对象所属的构建器;

makeAssociation (TopicIF assoc_type):在当前主题图中为指定类型的主题创建一个新的关联;

makeAssociationRole():为一个具体的关联指定关联角色类型;

makeOccurrence():为指定的主题创建内外外部资源指引;

makeTopic():为当前主题图创建一个新的主题。

(2)TopicMapStoreFactoryIF:创建新的主题图存储对象构造器接口。该接口根据不同的主题图存

储后台用不同的类来支持其实例化。其中 InMemoryStoreFactory 类用于实现基于内存的主题图存储实例, RDBMSStoreFactory 类用于实现基于关系数据库的主题图存储实例。该接口中只定义了一个抽象方法 createStore(),用于创建主题图存储对象。

(3)TopicMapStoreIF:表示主题图对象与特定存储实例之间的关系。该接口功能用于从特定的存储实例中检索主题图,回存已修改数据。该接口实例可以支持不同类型的存储实例,一般可以是基于内存和基于关系数据库作为存储实例。需要说明的是该接口支持简单的事务功能。一个主题图存储对象可以调用 Open()或 getTopicMap()方法打开;为了支持主题图对象的持久化可调用 commit()方法为其提供支持。

(4)数据交换接口: TopicMapImporterIF、TopicMapReaderIF 和 TopicMapWriterIF:主题图输入与输出接口。利用该接口允许主题图以不同的数据格与应用程序交换数据,其支持的主题图格式有 ISO CTM、LTM、RDF、XTM 以及 RDBMS 后台连接,其操作方法与交换格式独立。

4 Ontopia 引擎 API 应用支持

4.1 创建主题图

利用 Ontopia 引擎来构建主题图应用程序,首先要创建主题图对象。在 Ontopia 引擎中每一个主题图对应一个主题图存储对象,存储对象代表主题图对象与具体存储器之间的联系。在 Ontopia API 中利用 TopicMapStoreIF 接口来定义主题图存储管理器,通过存储管理器调用 getTopicMap()方法获得主题图对象。

由于 Ontopia 引擎支持两种不同的存储后台,因此 TopicMapStoreIF 接口有两个实现类: InMemoryTopicMapStore 和 RDBMSTopicMapStore,分别用于支持基于内存和基于关系数据库的存储后台。

InMemoryTopicMapStore 类位于 net.ontopia.topicmaps.impl.basic 包中, RDBMSTopicMapStore 类位于 net.ontopia.topicmaps.impl.rdbms 包中^[3],在实例化主题图存储接口时需要声明相应包。

当创建完成主题图后要向主题图中添加数据,一般不直接实例化主题图对象(无论是来自 InMemoryTopicMapStore 类,还是 RDBMSTopicMapStore 类)。不这样做的原因是:如果直接实例化主题图对象会把代码限制在一个引擎实例中,这必然会降低主题图应用程序选择引擎实现的灵活性。因此,在向主题图中添加数据时,首先从 TopicMapIF

接口中获得一个 TopicMapBuilderIF 对象,并利用该对象的构造器来创建新的主题图对象,然后再调用该对象的相应方法来配置对象。

4.2 支持 XTM 格式数据交换

在 Ontopia 主题图应用中,应用程序与主题图交换数据是通过引擎 API 提供的 Reader/Writer 接口来实现的。支持的数据格式如前所述,对于每一种数据格式 Ontopia 引擎都提供了相应的实现类来支持抽象接口,如图 5 所示。

(1) 从 XTM 文件中读取主题图

为了从 XTM 文件中读取主题图,Ontopia 引擎在 net.ontopia.topicmaps.xml 包中用 XTMTopicMapReader 类来实现。具体使用时 XTMTopicMapReader 构造器将阅读器与文件绑定,调用 read() 方法可获得主题图,阅读器自动创建 InMemoryTopicMapStore 对象,并把主题读进一个空的主题图对象中。

利用 XTMTopicMapReader 类创建的阅读器不仅要负责创建 XML 解析器和主题图存储对象,还会利用 mergeMap 和 topicMap 元素实现主题图的自动合并。

除此之外阅读器 Reader 可以适应多种不同的应用环境。例如,可用 TopicMapImporterIF 中的 importInto() 方法将 XTM 文件读入指定的主题图对象中;如果 XML 文档中包含了多个主题图文档,即包含了多个 topicMap 元素,则调用 readAll() 方法同时调入多个主题图。

(2) 输出 XTM 文件

在 net.ontopia.topicmaps.xml 包中利用 XTMTopicMapWriter 类可以输出由引擎创建的主题图。当 XTMTopicMapWriter 类配合 java.io.Writer, 直接调用 write(TopicMapIF topicmap) 方法把主题图输出到 writer。因此,在应用程序代码中应包含下面这条语句:

```
new XTMTopicMapWriter ("numbers.xml").write(topicmap); // topicmap 为要输出的主题图对象, numbers.xml 为输出的存储文件。
```

4.3 应用工具支持

Ontopia 引擎设计的主导思想是利用核心接口来表示主题图数据,其他接口则建立在核心接口之上,这意味着如关联遍历、主题图合并、信息查询等功能都置于核心接口之外。利用 Ontopia 引擎开发应用程序时 utils、entry 和 events 包提供了丰富的功能接口用于对应用工具提供支持,这样的优点是使得这些接口中的功能与主题图的实现方式无关,因此便于主题图应用程序的移植。

下面是关于引擎工具包对主题图应用功能支持分析:

(1) 对象名称抽取器 (Stringifiers): 在主题图应用中经常需要提取主题及其关联的名称或类型。一方面根据主题图的技术标准,主题类型、关联及其关联类型都认为是一种主题;另一方面由于主题图允许一个主题可以有任意多个不同类型名字,因此从主题中抽取主题名操作非常复杂。

在 Ontopia 引擎中可以利用 net.ontopia.utils 包中的 StringifierIF 接口定义一个主题名称抽取器 stringifiers,再结合 net.ontopia.topicmaps.utils 包中的 TopicStringifiers 类提供的方法可以从主题中抽取名字。

(2) 事件监听 API: 在主题图应用程序中,当主题图发生改变时应用程序往往希望获得反馈信息。Ontopia 引擎中提供了事件监听机制 (event API) 可以帮助应用程序进行事件监听注册,从而可以获得主题图发生改变的反馈信息。

Ontopia 引擎中 Event API 位于 net.ontopia.topicmaps.core.events 包中,该包中包含一个 TopicMapListenerIF 接口和 TopicMapEvents 类。其中 TopicMapListenerIF 接口必须由事件监听器实现, TopicMapEvents 用于注册监听器。

目前 Event API 支持三种事件:主题添加 (Topic Added)、主题修改 (Topic Modified) 和主题删除 (Topic Deleted)。

需说明的是基于内存和基于关系数据库的主题图存储事件激活方式存在差异。在内存中实现主题图只要处于 auto-commit 模式;事件就会自动激发,在 RDBMS 实现中只有提交事务时才激发事件。

(3) 主题图来源配置及引用支持: 在主题图应用程序中需要访问的主题图一般是以文件的形式存储在某个文件目录中或在关系数据库中,Ontopia 引擎称为主题图来源。应用程序要访问主题图首先要知道其来源即存储位置,然后再创建一个对象指向其来源称为主题图引用。

在 Ontopia 引擎中提供了 TopicMapSourceIF 和 TopicMapSourceIF 两个接口来处理主题图来源和引用。其中 TopicMapSourceIF 接口支持多种不同格式的主题图数据源,如 RDBMS、RDF、URL 以及 XTM 等,XTMPathTopicMapSource 类是该接口的 XTM 数据源实现类;每个主题图引用都是 TopicMapReferenceIF 接口的对象,指向一个主题图存储对象,因此主题图引用就间接指向主题图。主题图引用对象通过调用 createStore (boolean) 方法来创建主

题图存储对象,为应用程序访问主题图提供服务。

如果应用程序中需要处理多个不同位置和不同形式的多个主题图数据来源,则可以通过 TopicMapSourceIF 接口的另一个实现类 XMLConfigSource 来处理,利用它可以读取包含多个主题图数据源的 XML 配置文件。

5 结束语

主题图技术作为一种具有表达语义关系的知

识组织工具,目前已越来越受到业界的关注。主题图引擎作为主题图表达、存储和应用的关键,对主题图应用技术的推广起着至关重要的作用。Ontopia 主题图引擎作为一种成熟的商业主题图知识套件 OKS 的重要组成部分,无论是在主题图标准支持,还是应用程序开发功能方面都提供了良好设计的 API 接口,因此随着 Ontopia 引擎开源战略转变必促进主题图应用技术飞速发展。

注释及参考文献:

[1]Michel Biezunski.ISO/IEC 13250:2000 Topic Maps: Information Technology [EB/OL], (1999.12.09) [2009-10-9].http://www.y12.doe.gov/sgml/sc34/document/0129.pdf.

[2]TopicMaps.org Specification.XML Topic Maps(XTM)1.0.[EB/OL]. (2001.08.26) [2009-11-23].http://www.topicmaps.org/xtm/.

[3]Robert Barta.TMQL Language Features[EB/OL].(2008-4-1)[2009-10-16].http://kill.devc.at/system/files/language.html.

[4]ISO/IEC JTC 1/SC 34:Information technology—Topic Maps Constraint Language. (2008.8.7) [2009-12-2].http://www.itscj.ipcj.or.jp/sc34/open/1053.pdf.

[5]ISO/IEC JTC1/SC34:Information Technology—Document Description and Processing Languages[EB/OL], (2006.6.19) [2009-12-19].http://www.isotopicmaps.org/sam/sam-xtm/.

[6]Kal Ahmed.TM4J—Topic Maps for Java[EB/OL],(2004-12-15)[2009-21-13].http://tm4j.org/

[7]朱良兵.开源主题图引擎 TM4J 应用研究[J].现代图书情报技术,2006(10):66-70.

[8]Kal Ahmed.tinyTIM—a tiny in memory TMAPI implementation[EB/OL],(2005-2-7)[2010-1-22].http://tinytim.sourceforge.net.

[9]Kal Ahmed.XTM4XMLDB—Topic Maps for XML databases[EB/OL],(2004-12-4) [2010-1-16].XTM4XMLDB—Topic Maps for XML databases.

[10]Ontopia: Ontopia releases OKS 5.0[EB/OL].(2009.07.19)[2009-07-26].http://www.ontopia.net/download/index.html.

[11]Ontopia: The Ontopia Knowledge Suite [EB/OL].(2009.07.19) [2009-01-26].http://www.ontopia.net/solutions/products.html.

[12]OKS 5.0 Releases: The Ontopia Topic Maps Engine—A Developer's Guide[EB/OL].(2009-07-19) [2009-11-10].http://www.ontopia.net/solutions/engine.html.

[13]OKS 5.0 Releases: The RDBMS Backend Connector—A Developer's Guide[EB/OL]. (2009-07-19)[2009-11-10].http://www.ontopia.net/solutions/rdbms/devguide.htm.

Research on the Mechanism of Open Resource Ontopia Topic Map Engine

LI Qing-Mao

(Department of Computer Science, Aba Teachers College, Wenchuan, Sichuan 623000)

Abstract: On the basis of a synoptic analysis of the Ontopia knowledge suite and the architecture of the Ontopia topic map engine, this paper does an in-depth analysis of the interrelations between the development package and the core interface of engine API and the ways of implementation. In view of the support of the Ontopia engine interface for Topic Maps in its establishment, storage, data exchange, development of application, etc, the author makes a comprehensive analysis of the mechanism of the engine API in combination with specific code.

Key words: Topic maps; Open source topic map engine; Ontopia topic map engine; Ontopia knowledge suite