语义桌面上的本体存储研究*)

余翔宇

(武汉大学软件工程国家重点实验室 武汉 430079) (中国地质大学地空学院 武汉 430074)

摘 要 作为一种新的个人信息管理方式,语义桌面将语义 Web 技术用于个人计算机,极大地方便了用户对个人计算机上的数据搜索和管理。与语义 Web 技术一样,基于 RDF(Resource Description Framework)格式的本体存储与管理技术在语义桌面中也起着至关重要的作用。本文深入分析了语义桌面环境中的本体特点,提出设计专门针对语义桌面本体存储的 RDF Repository,并给出了具体的实现方案。

关键词 语义桌面,语义 Web,个人信息管理系统,本体存储,RDF Repository

Ontology Storage on the Semantic Desktop

YU Xiang-yu

(The State Key Lab of Software Engineer, Wuhan University, Wuhan 430079, China)[1]

(Dept. of Geophysics and Geomantic, China University of Geosciences, Wuhan 430074, China)[2]

Abstract As a new Personal Information Management, Semantic Desktop applies the semantic Web technologies to the personal computer and help users to search one's own personal computer efficiently. Like the semantic Web, the RDF (Resource Description Framework) Repository still plays an important role in Semantic Desktop. After analyzing the difference between the RDF repository for Semantic Deskto Pand Semantic Web counterpart, this paper presents a special RDF repository for the semantic desktop, and then gives its implementation in detail.

Keywords Semantic desktop, Semantic Web, Personal information management, Ontology storage, RDF repository

1 引言

当今社会,随着信息的爆炸性增长,如何有效和快速地寻找到自己所需要的信息成为一个日益突出的问题。然而,伴随着互联网上搜索技术的一次又一次革命,个人电脑上的搜索技术却仍然停留在一个相对非常落后的水平。一方面,磁盘容量的增长使得个人电脑上的数据越来越多;而另一方面,传统的桌面搜索技术变得越来越无法适应人们的需求。个人信息管理(PIM)的研究者们于是将目光投向了 Web,希望通过将 Web 上的搜索技术如关键字全文检索和本体推理等引入到桌面搜索中来。于是在 2003 年,德国的 Leo Sauermann 教授提出了语义桌面(Semantic Desktop)[1]的概念,旨在将语义 Web 的技术应用扩展到个人信息管理领域。从此,语义桌面迅速引起了人们的关注。在开始阶段,语义桌面的研究者们简单地将语义 Web 的标准和技术移植到桌面系统中。然而,随着研究的深入,人们逐渐认识到语义 Web 和语义桌面之间应用环境的差异,并感觉将语义 Web 上的技术直接照搬套用到语义桌面中效果并不理想。在文献[2]中提出应当在语义桌面中建立适应个人信息管理需要的本体及其本体语言。而对于语义 Web 技术的本体存储容器 RDF Repository 而言,其应用环境的变化也给其带来了新的挑战。本文首先介绍语义桌面的定义及其体系结构,然后介绍语义 Web 应用中的本体存储方法和种类,接着详细论述语义桌面中的本体特点,最后说明如何针对语义桌面系统设计和实现专门的本体存储。

2 语义桌面

语义桌面的定义如下:"语义桌面是用来整合用户个人计算机上所有信息的一种方案,这些信息包括文档、多媒体和消息等都被作为语义网资源来进行统一标示,并可通过 RDF 格式来进行查询和读取。用户能获得互联网上的资源,也可以把自己创作的内容与他人共享,并且应用程序也可通过本体建立起信息之间的语义联系进行互操作。语义桌面应用程序具有比传统的桌面应用程序更好的整合和交互能力,用户将从中获益[3]。"

我们以一个实际的例子给上述定义作一个解释。假设怀特先生是一位美国教授,他曾被邀请来中国参加一个会议,但却很遗憾地因为一些原因而未能成行。在后来的一次聚会中,怀特先生与朋友们聊到了中国,他忽然想起那次会议的邀请邮件中附有一张中国的风景照。于是他想把这张相片找出来给朋友们看看。但很不巧,怀特先生忘记了这封邮件的附件保存在哪里以及这张照片的名称。加上其电脑上的文件又太多,使用传统桌面搜索方法必然要花费很长的时间。那么怀特先生如何根据自己还记得的这些信息快速地找到这张照片呢?这就是语义桌面要解决的主要任务之一。

客观来讲,目前的语义桌面系统开发还处于比较初级的阶段。Gnowsis[4]是目前最具代表性的语义桌面系统。它将桌面数据资源整合到统一的 RDF 图中,并提出 PIMO(Personal Information Model Ontology)用来表示用户的个人主观模型,使得桌面搜索结果更加精确和个性化。另一个主要的语义桌面研究项目是 Haystack[5]。和 Gnowsis 相比,它的研究重点在于其集成了许多应用程序的功能,包括文字处理、图像处理、邮件传输、实时通讯等等,并且提供了一个从用户界面到数据库的完整语义编程环境。类似的语义桌面系统还有 Fenfire[6]。它提供了一个可视化的 RDF 图形编辑界面,允许

*)基金项目:中国地质大学(武汉)优秀青年教师资助计划资助项目,项目编号:CUGQNL0731。余翔宇 讲师,博士生,研究方向为 Web 技术、数据库。

用户来建立各种信息资源的内在联系，从而帮助人们按照自己的需要来对各种类型的数据资源进行组织。

一个通用语义桌面系统的体系结构如图 1 所示。其中：RDF repository 是语义桌面系统中的一个核心组件，主要用来存储和管理语义桌面中以 RDF 格式表示的语义知识；语义搜索模块主要用来提供语义桌面系统中与用户相关的查询功能接口，其中也包括了关键字全文检索功能接口；本体编辑模块主要给用户提供一个对资源之间进行关联以及对本体进行编辑的用户操作界面；语义抽取模块主要对计算机中的数据资源进行标注以及自动获取资源内容信息。

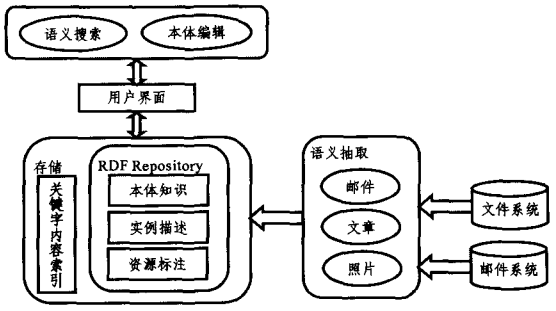


图 1　语义桌面体系结构

3　本体存储

如同语义 Web 中的应用一样，基于 RDF 格式的本体存储管理模块即所谓的 RDF Repository 在语义桌面中同样占有着重要的地位。由于语义 Web 的发展使得 RDF Repository 已经产生了较多成熟的产品，因此在语义桌面系统中人们通常选择复用这些已有的本体存储产品，如 Gnowsis 在较早版本中使用 Jena＋mysql 对本体进行存储，现在的版本中则使用的是 Sesame2。

依照所采用的存储方式，可以将 RDF Repository 产品区分为以下 3 类[7]。1. 基于文件的存储方式。这类产品通常将 RDF 数据直接存储在 XML 格式文件中。在使用时首先将文件中的数据装载进内存并按照某种结构进行组织，然后在该结构上执行数据的查询、编辑操作，使用完毕后再将内存中的数据结构写回到文件里面。这类产品有 OntoEdit，Protégé 等。它们通常主要用来编辑和建立本体，而并不适宜为本体数据提供存储和查询管理服务。2. 基于数据库的存储方式。这类产品的特点是将 RDF 格式数据按照一定的策略组织在数据库中，并利用现有的数据库系统对数据的操纵和管理能力来进行本体的存取。由于关系数据库技术发展成熟，大多数现有的 RDF 数据管理工作使用关系或对象-关系数据库管理系统作为后台存储，其代表系统包括 Sesame，Jena，3store 等等，其中 Jena 还提供对嵌入式数据库 Berkeley DB 的支持。由于关系数据库并不是针对 RDF 数据的特点设计的，RDF 数据复杂的图形结构与关系数据简单的扁平结构之间存在着巨大的差异。因此这种存储方式的关键问题在于如何将复杂的本体图模式匹配成简单的关系模式。3. 基于文件系统建立的 native storage 存储方式。这类产品的特点是对 RDF 三元组进行某种索引，然后设计专门的存储模式来将 RDF 数据及其索引写入磁盘，从而由底层向上提供对 RDF 数据查询与管理的支持。其代表系统有 kowari[8] 和 sesame2[9]。这类产品通常需要对磁盘进行不断的读写操作，所以对速度会有一定的影响；但由于其减少了数据冗余，因而从另一个方面提高了查询效率。

4　语义桌面中的本体特点

从本体的管理规模来看，语义桌面主要应用于个人电脑之上，更多的是为单机提供服务；而语义 Web 的应用环境却是整个万维网，显然语义桌面中的本体数量较少。这也意味着语义 Web 对 RDF Repository 产品的要求更高。这样看来，选用基于数据库的本体存储方法在语义桌面中显得并不合适。首先，由于语义桌面是面向个人用户的，因此数据库的许多功能如并发操作等实际上就被浪费掉了；其次，强制用户在本机上安装数据库也显得并不友好。而使用 native RDF Repository 在语义桌面中进行本体存储显得最为合适。首先，它十分有利于无缝地整合到语义桌面系统中；其次，也有利于开发人员根据自己的需要对原有的存储模式进行改进。这也是 Gnowsis 系统改为使用 Sesame2 的主要原因。

除此以外，语义桌面还具有独特的本体分层管理特点。这是因为语义桌面主要针对的是用户个人电脑上的信息，所以它必须将用户强烈的个人主观意愿表现出来。换句话说，在语义桌面中除了要表达语义 Web 中关于“这个事物是什么”的知识陈述，还必须表达出用户对于“我认为这个事物是什么”的描述。这样，语义桌面中就对本体进行了不同层次的划分，不同的文献中对其划分不尽相同，但总的来说存在着个人主观本体、领域本体、资源本体这三种类型。我们利用图 2 来分别对它们进行解释。

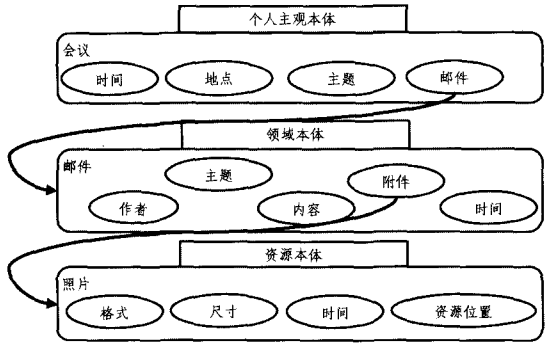


图 2　语义桌面中的本体分层

首先，个人主观本体(Personal Mental Ontology)源于个人主观模型 PMM(Personal Mental Model)，它用来说明在用户的概念中所存在的东西。对于同样的一个概念，不同的用户其主观模型常常存在着差别。在语义桌面中，个人主观模型可以通过个人电脑中的文件夹建立方式、文件的存放位置等等信息来捕获[10]，但更多的还是通过用户自己来建立。例如在上面的例子中，我们可以知道在怀特先生的电脑里一定存在着一个叫做会议的主观概念模型。这个会议模型的定义则存在于怀特先生的主观意识里。与此对应，领域本体则用来描述某一个领域内的公共知识，它有着比较明确的定义。比如对于图 2 中的邮件本体，在不同的个人电脑中其定义都是统一的。领域本体通常会随着桌面系统的应用程序一起被引入进来，比如用户电脑上的 outlook 软件决定了其 PC 机上引用的邮件本体。而资源本体主要用来对文件资源进行标注和描述，说明该资源是什么，和前面两个本体的不同之处在于，资源本体中直接包含了对资源位置的引用。

语义桌面中这种本体的分层机制主要是由用户自顶而下的查询模式决定的。我们通常在建立文件夹的时候就遵照了这种模式，如子文件夹名通常是父文件夹名的子概念或组成部分。在语义桌面中，个人主观本体占有着最重要的地位，这也反映出语义桌面具有的强烈的个性化特点。

另外，与语义 Web 中的本体描述不同，语义桌面中关于谓词的表述是很模糊的。就是说用户并不关心太详细的概念与概念之间的具体关系，而更多希望表明的是一种概念与资源之间的关联“related”和包含“contain”。比如某个会议和某人有关，某个文件和某人有关等等。这也促使语义桌面的研究者们研究出新的本体语言和推理法则来对语义桌面的特征进行更详尽的描述。

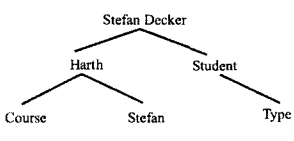
5 设计与实现针对语义桌面的本体存储

虽然在语义桌面中可以简单地复用为语义 Web 设计的 native RDF Repository，但为了更好地适应语义桌面中的本体查询和管理需要，笔者专门针对其特点设计与开发了一个轻量级的 native RDF Repository。它能够更贴切地支持语义桌面系统的需求。以下将详细介绍它的设计与实现过程。

通常的 native RDF Repository 设计思想是：直接将 RDF 三元组即主语(Subject)、谓词(Predicate)、宾语(Object)进行索引并以合理的方式存储起来以满足查询和更新的需要[11]。根据上述语义桌面的特点，我们需要将待存储的本体进行分类，然后再将它们的定义与实例分开分别进行存储。因此在笔者的系统中所设计的基本存储单位是一个由 Subject，Predicate，Object，Type，Context 组成的五元组(S,P,O,T,C)。其中，前三个元素与 RDF 三元组一致。Type 字段表示的是该元组所描述的类别。比如 Type 值为 1 时代表该元组描述的是领域本体的定义，值为 2 时描述的是领域本体的实例。Context 则可认为是一个备注字段，它的具体内容由实际应用来决定。在语义 Web 中，通常 Context 字段用来描述本体信息的来源。而在面向语义桌面的存储中，我们可以根据 Type 值的不同来赋予不同形式的 Context。由于在桌面搜索中，时间序列也通常被作为搜索的重要条件之一，因此笔者在系统中利用 Context 来标明元组的建立或修改时间。

为了提高查询效率和节省冗余，我们用 64 位长整数 NodeOID 作为索引来表示五元组(S,P,O,T,C)中的各个元素。该索引的建立我们参照了 kowari 的实现方法，利用二叉平衡树来对元素字符串进行排序和组织。除此以外，我们还建立了 OIDNode 的反向索引来根据字符串取得所对应的整形值，这样做也可以方便字符串之间的相互引用。然后，再进一步利用 lucene 等工具来对字符串进行分词，各索引关系的简单示例如图 3 所示。由于在语义桌面中资源的绝对路径值常常含有许多重复的成分，因此这种索引方法能取得非常好的效果。

key	value
0	Stefan Decker
1	Harth
2	Student
3	Course
4	Stefan
5	Type



key	value
Stefan	0,4
Harth	1
Student	2
Course	3
Decker	0
Type	5

图 3 索引示例

在 RDF 的查询过程中，查询的顺序模式起着非常重要的作用，即用户是根据已知的主语 Subject 来查找与之匹配的项，还是从宾语 Object 出发来进行条件匹配。在这里，我们也需要认真考虑语义桌面的查询顺序模式。与语义 Web 相比，语义桌面最主要的查询目的是为了找到用户想要的资源，因此它总是会遵循一个自顶向下的查找顺序。正如上文怀特先生的例子所示，其查找的目标始终是照片文件。无论是通过会议来找到照片，还是通过邮件来找到照片附件都是试图从一个相对较大的概念或实体出发找到较小的实体的资源。由于笔者所设计的五元组(S,P,O,T,C)中的 T 值能反映出本体的层次类别，因而以 T 值来建立顺序索引在语义桌面的搜索中能起到很好的作用。另外，以主语和宾语作为已知条件的查询顺序模式在语义桌面中显得比较突出。因此在系统中，我们主要设计了 SPOT，POT，OTS，TSP 这四组索引。和语义 Web 不一样，语义桌面的查询语言通常是通过界面操做隐式执行的，因此在设计系统时也可以事先根据界面设计和功能设计来决定用户的查询顺序模式，并建立相应的索引。

另外，为了存储概念之间的继承关系，在语义 Web 的一些 native 本体存储系统中常常会用二叉树结构来对其进行描述和存储[12]。但在笔者的系统中并没有专门针对继承关系设计特定的存储机制。其原因主要是我们认为对个人用户而言，其电脑上的数据一般不会精确到需要去做很细的概念分类。

索引建立好后，我们利用轻量级的开源 B+树底层 JDBM[13]来实现其存储。最后在其之上根据用户界面操作来完成对本体进行存取和查询的函数实作。

结束语 由于语义桌面系统对本体的存储和管理提出了新的需求，简单的复用语义 Web 中的本体存储容器既显得大材小用，又不能起到很好的效果。这就促使笔者想到自底向上地开发一个轻量级 RDF Repository 来专用于语义桌面系统。在完成了系统初步的设计与实现之后，作者将进一步将其进行完善，使之更好地符合语义桌面的特点。

可以预计的是，随着人们对语义桌面的逐渐关注，越来越多专用于语义桌面的技术必将被开发出来，使得语义桌面最终能被普遍和有效地利用到日常的个人电脑信息管理中。

参 考 文 献

[1] Sauermann L, Bernardi A, Dengel A. Overview and outlook on the semantic desktop// Proc. of Semantic Desktop Workshop at the ISWC. 2005

[2] Sauermann L. Pimo-a pim ontology for the semantic desktop (draft). Draft, DFKI, 2006

[3] 李胜，胡和平，卢正鼎. 语义桌面——个人计算机技术的未来发展方向. 计算机科学，2007，34(5)

[4] Sauermann L, Grimnes G A. Semantic Desktop 2.0: The Gnowsis Experience // The 5th International Semantic Web Conference at the ISWC. 2006

[5] Quan D, Huynh D, Karger D R. Haystack: A platform for authoring end user semantic web application// International Semantic Web Conference. 2003

[6] the fenfire project. http://fenfire.org.

[7] 鲍文，李冠宇. 本体存储管理技术研究综述. 中国科技论文在线. http://www.paper.edu.cn

[8] Wood D, Gearon P, Adams T. Kowari: A Platform for semantic Web Storage and Analysis // Proceedings of the 14th International WWW Conference. 2005

[9] Seseame2. http://www.openrdf.org/. 2006

[10] Chirita P A, Gavriloaie R, Ghita S, et al. Activity Based Metadata for Semantic Desktop Search// Proceedings of the 2nd European Semantic Web Conference. 2005

[11] Harth A, Decker S. Optimized Index Structures for Querying RDF from the Web// Proceedings of the 3rd Latin American Web Congress. 2005

[12] Chen Y, Ou J, Jiang Y, et al. HStar-a Semantic Repository for Large Scale OWL Documents // Proceedings of the First Asian Semantic Web Conference. 2006

[13] JDBM. http://jdbm.sourceforge.net/