

数据库与超文本系统的连接

Approaches to Integrate Database and Hypertext System

杨继国 杨冬青 唐世渭

(北京大学 北京100871)

摘 要 In this paper we proposed three approaches to integrate database and hypertext/hypermedia system. We pointed out characteristic for each method and compared them with each other, and introduced a versatile WWW interface system for databases-Web2DB

关键词 Hypertext, Hypermedia, WWW, Database

1 引言

超文本/超媒体技术从八十年代以来发展很快,在理论上建立了多种超文本抽象模型,如 HAM、Dexter、Trellis 等,也先后出现了一批应用的超文本系统,如 HyperCard、KMS、Intermedia、Microcosm 等。如今 Internet 上,最流行的服务 World Wide Web(WWW)就是以超媒体的形式来将分布在全球上的惊人数量的信息组织起来的。

随着应用的发展,超媒体技术需要与其他信息技术相互结合,和数据库的结合就是其中最为典型、也最为迫切的一种。其原因如下:

* 随着现有的超文本/超媒体系统的不断扩展,在 WWW 中迫切需要能够存取以前或者是在其他系统中积累的大量数据信息,而这些信息绝大多数都存放在数据库中。

* 绝大多数的超文本参考模型都没有提供节点和链信息的具体存储结构(这当然是为了避免局限于某种具体的存储方法,提高灵活性),然而在实用的超文本系统中,必须要考虑如何有效地存储和管理这些信息,有些系统采用了文件的方法,有些系统则借助于数据库。从发展来看,当超文本系统形成一定的规模时,也就是所谓的 Hypertext-in-Large 系统,存储及其相关的问题就成为极其重要的制约条件,数据库作为存储管理的基础必须要和超文本紧密配合。

* 从数据库技术本身的发展来看,现有的关系数据库(RDB)和面向对象数据库(OODB)在不同程

度上都存在对相互联系着的多媒体信息缺乏表达能力的问题。特别是 RDB、DBMS 对于多媒体信息仅是将它们存储起来,不能够体现多媒体信息的本质属性,以及它们之间的天然联系,也就无法对这种联系提供管理和操纵上的支持,而将任务交给了应用程序,然而信息间的联系却是本质上的、内在的。OODB 对于表达单个多媒体信息的语义比 RDB 出色,但对于表达多媒体信息之间联系的支持也很弱,而超媒体技术则是一种较好的表达途径。因此,将超媒体技术与数据库技术相结合,建立新型的数据库系统,是解决当前所面临的信息支离的窘境的一条较好出路。

超文本和数据库本质上都是现实世界信息的表现和组织方式,只不过组织的方式和传统上的侧重点不同。目前超文本技术和数据库技术基本上还是两个独立发展的领域,关于它们相结合方面所做的研究和工作尚不多,本文试图从体系结构的角度探讨数据库和超文本系统相结合的几条途径,为以后这方面的进一步研究做初步的探索,最后介绍一个我们实现的一个 Internet 与数据库相连接的系统 Web2DB。

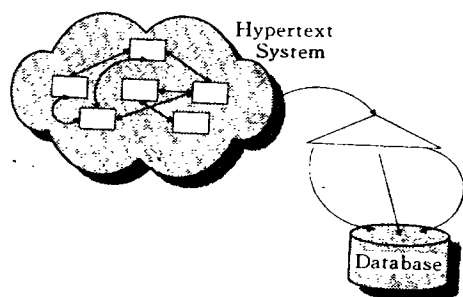
2 连接途径

超文本和数据库都是以现实世界的信息为基础的,它们的连接可以有以下几条途径:

2.1 途径之一:数据库作为超文本系统的一个节点

在这个途径中,数据库被看成超文本系统中一个独立的特殊节点,由特殊的链将数据库节点和其

他超文本节点相连接。如图一所示:



图一 数据库作为超文本系统的一个节点

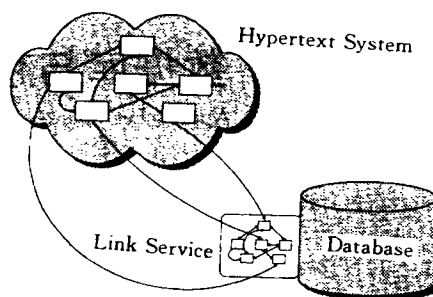
图中的三角形表示一个转换机制,超文本系统中要访问数据库信息,需要将访问请求(如 Follow Link)转换成为数据库的查询语言(如 SQL),其结果也需要进行转化,成为超文本系统所规定的格式(如 WWW 中的 HTML)。指向数据库节点的链是一种特殊的链,它不同于普通的按锚(Anchor)位置定位的链,而是依靠计算或查询来定位的,也就是链所指出的锚实际上要经过数据库系统先执行一个查询来确定。这种结合方式的特点是数据库作为独立的一个节点,其内部机制没有任何变化,仍然依靠原来的 DBMS 来管理数据,不影响其他非超文本应用;同时数据库系统整体作为超文本系统的一个节点,其内部锚的定位机制要依靠数据库系统的查询。

这种途径的优点是结构比较简单,相对容易实现,不需要对现有的系统作很大的变更。然而它有一定的不足:首先超文本系统和数据库系统的信息组织和管理方法不统一,这就造成了在两者访问时需要进行请求和数据的转化,其总体结构对用户来说不好把握;其次数据库整体作为超文本系统的单个节点,使得超文本系统的用户很难与数据库内部的各个信息单元建立连接;再者超文本系统及其用户需要了解数据库系统的管理机制,以及用户数据的模式,才能建立两者间的联系,而对用户来说,实际上要同时理解两个不同的系统,这一点对超文本系统的用户有时是相当困难的;最后由于数据库中的数据的管理实际上仍完全依靠本地的 DBMS,这样使得超文本系统无法从全局的角度对超文本应用进行统一的管理和协调,如并发控制,版本管理,等等。

2.2 途径之二:数据库上附加链服务

这种方式是在数据库系统的基础上,增加一层专为超文本系统设计的链服务,也就是对数据库中的数据建立一个超文本结构,附加在数据库上,然后

与超文本系统相连接,如图二所示:



图二 在数据库上附加链服务

在这个途径中,需要对数据库中的信息进行再组织,形成超文本结构,以便于与超文本系统连接。在数据库中的数据原来是以数据库系统所规定的模式来存放的,如 RDB 中以表为基本结构,在这种情况下,要提供一种机制,使用自动或用户干预,或者两者相结合的方法将在原来以数据库模式存储的数据基础上形成节点、链这样的超文本结构。数据库中的数据被重新组合形成虚拟节点,在此基础上根据信息的内在联系建立节点间的链。这样,超文本系统就能够以更自然的方法与数据库中的信息建立连接,而不需要特殊的机制。

在现代的超文本系统中,一般都采用链和节点内容分开存放的方法,一方面便于信息本身在非超文本系统和与超文本系统之间的共享,另外一方面也为在数据信息上建立多套不同的连接提供了可能。实际上我们可以看出,在数据库上附加链服务实际上与超文本系统中的链没有太大的区别,只不过链所连接的数据是在数据库中存放的。

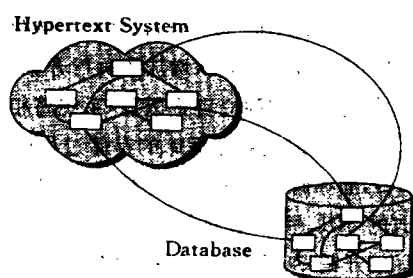
这种方式有以下特点:1)数据库中的数据在逻辑上被再组织形成超文本结构,实际上是在数据库上提供超文本接口,从而使超文本系统与数据库在体系结构上达到统一;2)数据库中的数据被组织成虚拟节点,从而超文本系统能够单独地访问数据库中单个的信息;3)数据库中信息节点间的链定位仍然要用到数据库的查询机制,但通过链服务系统,可以对超文本系统透明,超文本系统用户可以不必再去了解数据库的结构和模式。

然而这种方法也有它的局限性,其根本原因是超文本系统和数据库系统在本质上还是两个结构和管理方式都不同的系统,总结如下:1)需要在数据库上附加链服务系统;2)虽然对于超文本用户来说不需要了解数据库的细节,但对于链服务的建立者来

说,仍是不可避免的;3)虽然超文本用户可以象普通链一样使用数据库链服务中的链,但最终链的定位仍然得依靠数据库的查询机制;4)对已有的数据库中的信息建立链接是一项相当有难度的工作,因为数据库模式所提供的语义信息往往比建立超文本结构所需要的要少得多,甚至根本就没有链的信息。这就需要提供一套机制(一般要有用户的辅助,可能再加上一些自动的方法)来建立链服务。关于链服务的建立机制是一个需要专门研究的问题;5)依然没有解决对超文本系统进行统一管理(并发控制,版本管理…)的问题。

2.3 途径之三:超媒体结构的数据库系统

这是一种全新的方法,它以新型的数据模型——超媒体模型代替传统的关系或面向对象模型来建立数据库,如图三所示:



图三 超媒体数据模型的数据库

在这种途径中,需要对现有的数据模型进行革新,结合超媒体技术,建立适合表达多媒体数据间自然联系的数据模型,以此模型为基础建立数据库系统。这样,超媒体系统与数据库系统之间就不会再有体系结构上的差异,连接更为自然、紧密。

正像我们在引言中提到的那样,建立超媒体模型的数据库还有更广阔的意义。一方面,超文本/超媒体系统本身有数据存储方式的问题,要使 Hypermedia-In-Large 系统能够有效地工作,建立高效的数据库管理系统来存储、管理超媒体系统中的信息(包括节点和链的)是非常关键的。因此将超媒体技术与数据库相结合,建立超媒体模型的数据库对于超文本/超媒体系统来说有着非常重要的意义,并且会随着超媒体系统规模的扩大变得越来越迫切。另一方面,为了从 DBMS 的层次上解决对多媒体信息及它们之间内在联系的语义支持问题,需要克服传统的关系或面向对象模型的表达能力不足的缺陷。而超媒体模型能够较好地表达信息之间的这种内在联系。实际上超媒体本身就是一种信息组织和管理

机制,以它为模型建立数据库也是非常自然的。

超媒体模型的数据库有着重要的意义,从超文本系统和数据库的连接,以及数据库技术发展的角度来说,它有以下优点:1)超文本系统和数据库系统是真正的无缝连接,不需要任何形式的请求和数据转换;2)由于在 DBMS 层次支持超文本链,可以使链的实现更高效、更适合超文本系统的特点(而不是前面所提到的那种借助数据库的原有查询机制);3)利用数据库管理机制可以更加有效地在超文本系统中进行并发控制、安全性控制、版本控制、事务处理等管理工作;4)从数据库的角度来看,这种数据库同时支持非过程性查询(如 RDB)和导航式查询,能满足不同应用的需要。5)媒体模型的数据库可以从 DBMS 层次支持多媒体信息之间的内在联系,提供建模和管理的功能,从而减轻了应用系统的工作,也使得维护系统的一致性更为有效。

这条途径由于是一种新的方法,有关超媒体模型数据库的研究工作尚不多见,还有很多问题需要探讨和研究,主要的困难之处有:首先目前的超媒体参考模型都没有考虑作为数据库系统的特点,而现有的数据库模型又没有结合超媒体的特性。因此如何将两种技术结合起来,建立以超媒体为基础的数据库模型成为最为关键的问题。数据库需要完备的数据模型,而这样的数据模型又必然要有严格的形式化基础,因此要建立超媒体模型的数据库,需要结合已有的形式化工具,同时考虑超媒体技术和数据库的特点,研究设计适合于数据库系统的超媒体数据模型。这方面的研究由于经验很少,需要积极地进行探索;其次建立超媒体模型的数据库,并不意味着抛弃传统模型的数据库,相反,它要吸收传统模型数据库的优点,并要提供最大限度的兼容性。传统模型的数据库已经经过多年实践的检验,尤其是 RDB,在当前市场上还占有主导地位,而且经过多年应用,被证明是成熟可靠的。超媒体模型的数据库应该吸收这些传统模型数据库的长处和经验,如 RDB 的 SQL 查询,OODB 的复杂对象语义表示能力等,同时也要兼容原来在这些数据库上的应用系统,这一点在实践上是非常重要的。为达到这个目标,超媒体模型数据库的设计必须考虑众多兼容性的问题。

3 Web2DB:一个 WWW 与数据库的通用接口系统

Internet 在近几年成为各界关注的焦点,这样

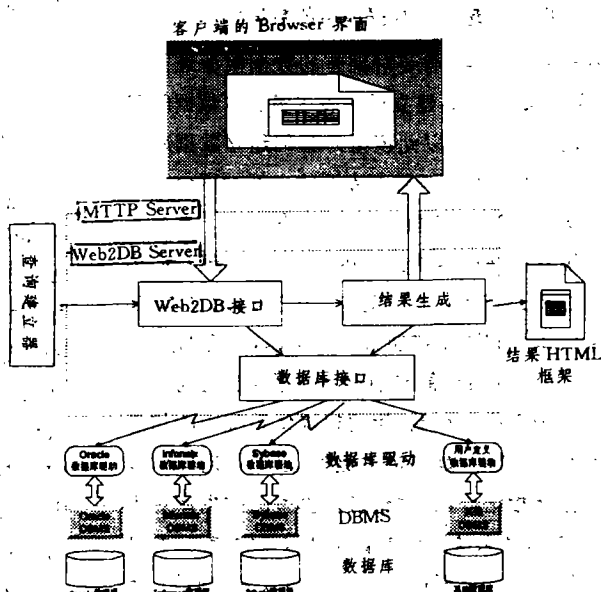
一个全球性的互联网络给人们带来了更多的交流机会。Internet 上的信息资源数量惊人且分布在全球各地,其主要的信息组织方式是 WWW (World Wide Web)。WWW 是一种结构上比较简单的超媒体,但也正是由于其简单性使得 WWW 得到了飞速发展。在很多情况下,WWW 需要检索数据库中的信息,而这些信息可能是由其他应用多年积累下来的。WWW 中与外部系统连接的方式传统上只有 CGI (Common Gateway Interface) 方法,直接用 CGI 与数据库连接有如下的缺点:1)没有通用性,对于每一个访问数据库的应用必须有专用的 CGI 程序;2)结果 HTML 由 CGI 程序硬编码形成,返回结果的表示格式的改变必然要求修改 CGI 程序;3)这种方法的效率不高,由于 CGI 程序是独立于 HTTP Server 的程序,需要由后者频繁地启动 CGI 进程,数据也在进程间经常交换,开销很大;3)对于设计应用的 HTML 编程人员,必须自己编写相应的 CGI 程序,而 CGI 程序的结构相当繁琐,又需要与数据库系统进行交互,因此 HTML 程序员的负担很重。基于这种情况,我们设计实现了 Web2DB,力图在 Internet 上的数据库连接提供一个较好的解决途径。Web2DB 的基本结构如右图。

Web2DB 的软件基本上都集中在服务器端,这是考虑到在用户端已经习惯了用单一的浏览器(如 Netscape 或者 Internet Explorer)来作为界面,一般用户难以接受附加的接口。用户的 CGI 请求(加上了访问数据库的扩展)以标准的 HTTP 协议传输到服务器方,由 HTTP Server 和 Web2DB 接口组成的扩展 WWW 服务器接受用户请求,然后执行由查询建立器形成的存储过程,将数据库中查询出来的数据结合框架 HTML 形成结果 HTML,通过 HTTP 协议返回给客户端的浏览器显示。Web2DB 的基本模块有如下部分:

• Web2DB Server。是系统的主要部分,它负责接受客户端的数据库访问请求,并通过调用查询建立器所建立的过程操纵数据库,再根据结果 HTML 框架形成返回的查询结果。一般情况下 Web2DB 接口与 HTTP Server 结合在一起,形成扩展的 WWW 服务器,这样做的原因是为了减少进程间调用和数据传递的开销,加快执行

速度(速度问题是目前 WWW 访问数据库的主要难题)。如果用户需要坚持自己已有的 WWW 服务器,Web2DB 接口也可以作为一个标准的 CGI 程序使用,用户仍然可以利用它的通用数据库接口,不过效率可能比前一种方式有一定的下降。这两种方式的使用接口是完全一致的。

• 查询建立器。用于帮助用户建立对数据库的操纵(主要是查询)过程。关系数据库访问的通用接口是 SQL 语言,很多 HTML 的设计人员不是非常熟悉,直接书写有一定的困难。同时,如果在运行时直接使用用户书写的 SQL 语句,一方面增加了网络传输(SQL 语句)的通讯量,再者 SQL 语句是一种说明性查询语言,DBMS 在执行时需要分析语法、优化并转换成内部形式执行,而一个应用的访问所使用的 SQL 语句在建立完成后一般就会固定下来,在实际应用中每次都要 DBMS 从头分析、优化和转化相同的 SQL 语句显得没有必要,增加了执行开销。通过查询建立器辅助用户建立查询以后,系统将相应的 SQL 语句序列形成关系数据库的存储过程(Stored Procedure),而存储过程是经过分析、优化并转换成内部形式的 SQL 序列,在执行时免去了类似其他语言“编译”的开销,从而提高了执行效率。当然使用存



储过程需要 DBMS 的支持,目前市场上的主流关系数据库,如 Oracle、Informix 和 Sybase,都支持存储过程的使用。对于不支持存储过程的数据库系统,查询建立器自己将 SQL 语句存储起来。

• 结果 HTML 框架。其实要由应用开发者来编

写,其主要的作用是用于告诉 Web2DB 接口如何将结果数据形成 HTML 文档,也就决定了用户查询结果在浏览器上的显示式样。将结果的格式从程序中分离出来的好处是显而易见的。一方面用户可以更灵活地控制输出的结果,另一方面也为同一个查询能以多种格式输出提供了可能。这个框架由开发者书写,基本的语法格式就是普通 HTML,也可以使用、修改已经存在的 HTML 文档,只需要将要出现结果数据的地方标出即可,除了 HTML 外无须另外的知识。

•数据库驱动。由于目前数据库系统种类较多,各种数据库之间使用的接口是不同的,即使同是关系数据库的不同厂家的产品也是如此。在目前的 Web2DB 版本中,我们只支持 UNIX 下的关系数据库系统,这是由于关系数据库有通用的查询语言 SQL,统一起来相对简单一些。首先,我们对上层模块(Web2DB 接口、结果生成)提供统一的数据库接口模块,即对上层来说不涉及具体的数据库产品,它们的通讯语言是标准的 SQL。对于每种关系数据库产品,Web2DB 通过对应的驱动程序来访问。系统提供了常见的几种主流关系数据库产品的驱动,有 Oracle、Informix 和 Sybase 接口,其他的数据库驱动(如 DB2、Unify 等)也计划提供。对于使用专门的或者目前还没有提供驱动的关系数据库产品的用户,Web2DB 还提供了一个用户自定义数据库驱动的接口,使用户可以编写自己的数据库驱动。由于同时存在的数据库接口可能很多,对数据库驱动如果采用静态链接到 Web2DB 系统的话有不少缺陷:首先,如果有较多的数据库驱动,WWW 服务器会变得很庞大,占用系统资源太多;其次,修改或添加用户驱动势必要重新链接系统,这就要求 WWW Server 系统要关闭下来,很不方便。基于以上原因,我们对数据库驱动采用了动态链接的方法,即哪些数据库驱动作为动态链接库一开始并不装入,只是在需要

的时候才载入内存。这样就避免了上面提到的缺陷,但目前需要操作系统支持动态链接库(如 Solaris, Linux 等)。对于不支持动态链接的操作系统(如 SCO Unix 3.2.4),我们提供静态链接的版本。

总结起来,Web2DB 具有以下一些特点:1)采用虚拟 CGI 方式提供服务,设计者无须额外学习其他技术,同时又获得了较高的运行效率。2)作为一种通用的 WWW 数据库接口,支持多种 DBMS 的访问,同时提供用户书写数据库驱动的接口。3)从上面的讨论可以看出,一个设计者使用 Web2DB 实现访问数据库的 WWW 应用时,在整个过程中他所了解的只是 HTML,无须了解 SQL 和数据库内部的复杂细节,这对于 HTML 页面设计者来说相对负担较轻,使用起来也比较方便。4)查询建立器和结果 HTML 框架的使用可以使得设计者能灵活地控制查询的过程和结果的格式,一个查询结果可以在不同情况下有不同的显示形式(比如在教学系统中显示给教师的习题可以带有答案,而同一习题给学生看时则没有答案);将查询独立出来单独存储使得同一查询可以在多个应用中使用。

参考文献

- [1] Halasz, Frank et al., The Dexter Hypertext reference model, CACM, 37(2)1994
- [2] John Smith et al., HYPERTEXT, CACM, July 1988
- [3] 胡晓峰, 老松杨等, 分布式超媒体系统 HDB 的设计与实现, 小型微型计算机系统, 16(10)1995
- [4] 余盛可, 超文本形式化模型研究, 计算机研究与发展, 32(3)1995
- [5] Akscyn, R. et al., KMS: A distributed hypertext for managing knowledge in organizations, CACM, 31(7) 1988
- [6] Dave Raggett, HyperText Markup Language Specification Version 3.0, Internet draft, IETF
- [7] Ian Graham, Introduction to HTTP and CGI, Online Document of NCSA Httpd