

一个开放式超媒体系统的设计与实现

The Design and Implementation of an Open Hypermedia System

99-102

黄晓春 姚航 姚卿达

TP391

(广州中山大学软件所 广州510275)

摘要 This paper introduces an open hypermedia system. First, we define the open hypermedia computation environment, emphasizing the power of hyperlink in integrating environment. Then we focus on our hypermedia system—LNOHS, proposing an object-oriented data model and the operation on the model. Finally, we facilitate the flag taxonomy to classify our system.

关键词 Hypermedia, Open hypermedia, Dexter hypertext reference model, Flag taxonomy

一、引言

超媒体系统的潜力在于组织和管理结构化、非结构化的信息,它为用户在包含各种媒体的信息仓库中进行浏览或信息检索提供了方便的手段,允许用户在小块的信息(节点)之间通过任意连接(链)来扩充自己的信息空间。

人们通常使用一些特定的软件包处理与工作相关的数据。例如,文字编辑,发送、接受电子邮件,任务计划,工作安排等。这些工作产生了大量的数据,并存储于计算机上,如何在这些数据之间浏览呢?通常的做法是将文件目录结构作为定位数据的主要依据,这就带来了很大的不方便。用户必须自己组织和管理这些数据,即使它们之间存在着某些联系。

超媒体技术的链接和浏览机制提供了灵活的信息组织与访问方式。采用超媒体技术进行信息集成是超媒体界的研究目标。本文在对超媒体、开放式超媒体系统分析的基础上,首先定义了开放式超媒体计算环境,然后提出了开放式超媒体系统的体系结构,并讨论了该系统的数据模型和链接协议,最后,利用 Flag 分类法对该系统的开放性进行了评价。

二、开放式超媒体

超文本技术作为一种非线性的信息管理技术,它以节点为信息的基本单位,通过“链”将具有相关性信息的节点连接形成网状的结构。通过浏览机制在节点之间迁移。超媒体就是多媒体超文本,它将超文本的节点与链的形式推广到多媒体形式。但一般

对超媒体和超文本这两个概念不做区别,本文也如此。

超媒体技术向用户提供了最大、最灵活的信息访问方式。用户可以在信息空间中自由浏览和访问信息,而不象传统的线性方式那样,只能以特定顺序浏览。

然而,另一方面,传统的超媒体系统仅用于文档管理和访问,其功能主要面向信息表现,例如, KMS, Mosaic, NoteCards 等超媒体系统以其各自的格式创建、显示和组织节点,并提供在节点间浏览的手段。这样的超媒体系统是封闭的超媒体。在这一超媒体系统之外,其它系统无法利用系统的节点信息,另外,超链接的强大的信息管理功能仅仅局限于超媒体系统,从而限制了超链接的应用。能否将超链接功能向其它应用系统开放,而不是仅仅局限于某一特定的超媒体系统呢?因此,近年来超媒体界提出了开放式超媒体计算环境。

目前,对于开放式超媒体系统(Open Hypermedia System)尚无统一的定义,一般从对超媒体系统的开放性要求来定义。即开放式超媒体应具备如下几个特点^[1]:1)系统可伸缩性;2)与第三方应用的可集成性;3)数据模型的可扩展性;4)分布性。

从超媒体对各种媒体数据处理的功能看,我们认为,开放式超媒体计算环境是指这样一个计算环境,在这个环境下,实现了对各种媒体数据的采集、加工(编辑、处理)、链接、存储、管理、传播和利用。

在具备了超媒体功能的计算环境中,超媒体服务是典型的中间件。计算环境中的各应用软件从不

同方面采集、加工、存储信息,这些从不同角度获得的信息共同描述了一个信息空间,借助开放式超媒体系统提供的超链接服务,计算环境中的应用软件成为“超媒体使能”,即具备了超媒体特性。虽然信息空间由不同角度的信息构成,但是通过超链接可以在表现了相关信息的应用软件之间迁移,而不需改变信息本身。

三、LNOHS 开放式超媒体系统

LNOHS——岭南开放式超媒体系统,是在我们开发的一个多媒体编播平台 MEDS 基础上发展起来的系统。在 MEDS 中,实现了超链接功能,然而,和其它传统的超媒体系统一样,该系统最大的缺点就是封闭性。因此,我们考虑将超链接功能独立出来,向计算环境下的其它应用软件提供一个统一的、标准化的超链接功能。

3.1 设计原则

设计之初,我们首先提出了一些自始至终应遵循的设计原则。

1) 开放的原则。我们认为,一个系统可被称为开放的,是指它规定并提供了其系统功能的接口,供其他应用程序扩充使用。因此,在 LNOHS 中必须提供这样的接口。

2) 结构与内容相分离的原则。我们注意到,在微软的 Office97 中,利用 WWW 技术提供了超链接功能。即在其任何一个应用工具环境下(Word、Powerpoint 等),都可以定义超链接,并可通过浏览机制迁移到另一应用环境。然而,我们认为,虽然 WWW 技术具有广泛的分布性,但是,WWW 技术采用嵌入链接方式(HTML 格式),将所有链的信息嵌入相应的节点,格式简单。具有如下不足:

a) 不易扩充。应用程序必须把 HTML 格式作为其基本格式,才可能支持类似 WWW 的浏览方式。

b) 产生悬链问题。由于采用嵌入链接格式,对节点的调整将影响浏览结构。例如,对一个节点对应的 HTML 文件的改名,删除操作,将使其他指向该节点的节点产生悬链。

3) 数据一致的原则。由于结构与内容相分离,因此必须考虑数据一致性问题。对锚的操作必须与设置了锚的区域上的内容(文本、图形等)的操作保持一致。即对锚区域上的内容的增删改操作必须同时对锚进行相应操作。

3.2 设计方法

1) 面向对象框架。面向对象的思想表现在其可

重用性上。基于开放性的考虑,我们认为这一方法对于开放式超媒体系统的设计十分有效。

2) 尽量利用现有技术。重新构造一个系统非常耗时,而且是不现实,不必要的,现有技术为实现开放式超媒体系统成为可能。利用现有技术使我们更专注于关键技术的研究和设计工作。事实上,我们利用了现有的 DBMS 技术和 DCOM 技术。用数据库管理超媒体数据库,而用 DCOM 作为实现开放性的基础技术。

3.3 体系结构

LNOHS 开放式超媒体环境分为几个部分:链接服务器,会话代理和处理各种信息的应用程序。除了提供超链接服务外,还必须让应用程序感知这一服务的存在。因此,必须对应用程序进行扩充,在应用程序层增加一个运行层。体系结构如图1所示。

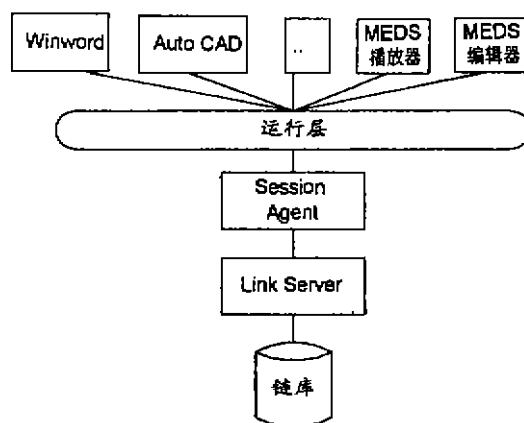


图1

1) 运行层。是对应用程序的扩充。超媒体界认为,要使现有的应用程序具备超媒体功能,必须对它们扩充,即增加一运行层。运行层主要负责定义、解释‘锚’在节点的位置。由于不同应用程序有不同定位机制,例如:x,y 坐标,对象标识,锚所在区域对应的内容等。因此,将对‘锚’位置的定义权、解释权交运行层处理。

另外,运行层必须保证数据的一致性。在对含锚的位置的区域进行增加、删除和修改等操作时,运行层应及时通知链接服务器,以便对锚信息进行相应操作。此外,也应保证对节点操作的数据一致性,即对节点的变动(增、删、改等)应及时通知链接服务器进行相应的修改。

2) 会话代理。它一方面与运行层联系,另一方面,与链接服务器连接,实现运行层、链接服务层的

会话。做为软件代理,具体负责定位链宿节点所对应的文件,并激活相应的应用程序。

3)链接服务器。通过与 DBMS 相连,提供链接服务,包括打开和关闭数据库和对链接数据的存储和检索,并向 DBMS 请求数据库服务。具体的数据库操作包括对节点、锚的增加、删除、修改等操作。链接服务器主要采用 DCOM 技术向应用程序提供链接服务,我们将在下一节详细描述链接服务器。

4)链接数据库。对关于链接的数据进行集中管理。

3.4 链接服务器(Link Server)

链接服务器主要对超媒体数据库进行管理,对外提供链接服务。包括对链、锚、节点等的增加、删除、修改等操作。对于链接服务器,我们将从数据模型及它所提供的链接服务两方面描述。

1. LNOHS 的数据模型,传统的超媒体系统通常从静态角度设计超媒体系统,例如,Dexter 模型^[2]认为,超媒体系统的主要元素有:锚(Anchor)、链(Link)、成员(Component,相当于节点 Node)等元素组成。

LNOHS 在 Dexter 模型的基础上,不但从静态(存储)的角度分析,而且从动态(运行)的角度设计系统的数据模型。链接数据主要由四个基本结构组成:成员(Component)、锚(Anchor)、实例(Instantiation)和锚标识(Marker)。数据模型如图2。

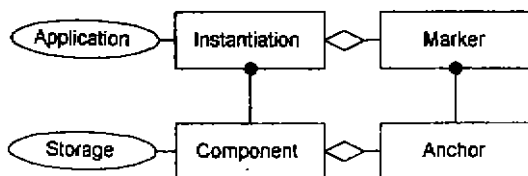


图2

应用程序,集成在开放式超媒体环境中的应用程序;

存储,存储具体成员的内部信息;

成员,又称节点,具有一个 ID 号唯一标识,另外还包括成员所对应的文件名、路径和文件的基本信息,例如文件创建者、创建时间、修改者、修改时间等。

锚,具有一个唯一的锚标识符及锚的基本信息。例如锚的创建者、创建时间、修改者、修改时间等。锚还有两个非常重要的域:位置和价值。位置描述了锚在成员中的具体地点,而值向链接服务器提供了计算

链宿成员的依据,即链接服务器通过计算锚的值获得链宿成员的信息。

实例,是成员运行时的表现。

锚标志,是锚在运行时刻的表现形式,包括锚标志的 ID,锚标志对应的锚 ID、锚的表现描述等信息。

实例/成员,实例和成员构成了多对一的关系,即一个成员可由多个实例表现,但一个实例只能表现一个成员。

锚标志/锚,与实例和成员的关系对等,锚标志和锚也构成了多对一的关系。即一个锚可为多个运行时刻的锚标志所表现,但锚标志只能对应一个锚。

成员/锚,成员和锚构成了簇聚关系。即锚是成员的组成部分,成员内包含了多个数量的锚。

实例/锚标志,与成员和锚相应,实例与锚标志也构成了簇聚关系。

对上述四种结构数据的操作是通过链接服务来实现的。

2. 链接服务。我们将链接服务器封装为 COM 对象,向外部应用程序提供链接服务。COM(Component Object Model)是 Microsoft 提出的软件构件方面的一个机制,COM 是一个面向对象的编程模型,在这个模型中,构件是一个二进制形式的可重用代码,它可以轻而易举地与其它软件供应商的构件组合。COM 为软件开发提供了如下优点^[2]:1)可重用;2)易扩充;3)版本管理容易。

COM 提高了软件间的互操作性。结合对开放式超媒体系统的要求(可集成性、可扩充性、可裁剪性等),我们采用构件对象模型设计和实现了链接服务器。所提供的链接服务主要包括:

- 1)Create Link:在节点之间创建链接;
- 2)Follow Lmk:按照链接关系,在节点之间迁移;
- 3)Compute Link:通过给定的锚的值,计算出链宿节点;
- 4)Get Anchor List:向应用程序提供一个节点所含的锚信息;
- 5)Get Structure:根据给定的节点计算与该节点有链接关系的所有节点。

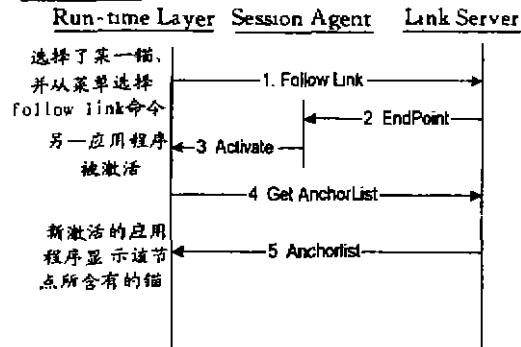
由于采用了构件对象模型技术,链接服务可根据需要进行扩充,例如增加,修改,替换链接服务器中计算链宿节点的算法等,而不影响服务的其它使用者。另外,外部应用程序也可容易地扩充为“超链接使能”,因为应用程序只需知道如何使用链接服务,而不必了解链接服务的具体实现情况。

基于以上链接服务器的特点,就可根据具体需要,通过使应用软件感知链接服务,从而将其集成到超媒体环境中。而对于新软件,可使用链接服务,直接加入超媒体环境,我们以前开发的 MEDS 编辑器和播放器与 LNOHS 超媒体环境达到了紧密耦合。

3.5 链接协议

让我们通过一个具体的链接服务流程看一看在超媒体计算环境下,如何将各种信息集成。

Application



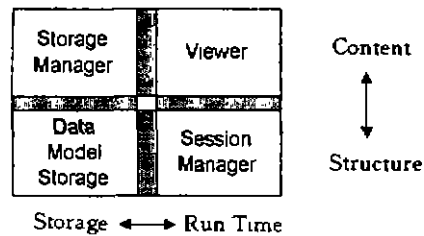
某一应用程序选择了一个锚并触发了一个“迁移锚”命令:

- 1) 运行时层向链接服务器发出请求执行按链迁移的链接服务 FollowLink;
- 2) 链接服务器根据“锚”信息,确定链宿节点后,向会话代理请求显示链宿节点;
- 3) 会话代理根据链宿节点的位置和类型,激活相应的(显示该链宿节点的)应用程序;
- 4) 应用程序向链接服务器请求该链宿节点所包含的所有“锚”信息(GetAnchorList);
- 5) 链接服务器返回相应节点的锚列表。

应用程序获得锚列表后,显示链宿节点的同时,在“锚”位置区域显示“锚”。应用程序转而与用户交互,直到需要其它链接服务时,再向链接服务器请求链接服务。

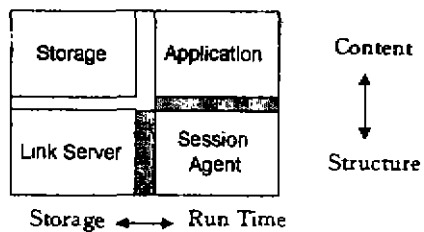
四、开放性评价

Flag 分类法^[4]从另一角度描述了开放式超媒体系统的结构,提供了划分和评价超媒体系统的方法,其体系结构如下:



对应于 Flag 分类法,LNOHS 的结构如下:

LNOHS 交由第三方应用程序存储和表现它们自己的数据,因此,在存储和表现层之间没有协议控制,而第三方应用程序可通过链接协议自由地访问链接服务(水平协议),而链接服务器暂时还未建立与存储层的协议,但由于链接服务器用 COM 对象实现,我们将对链接服务器进行扩充,实现与存储层的通信。



结束语 超媒体系统以其独特的信息组织和管理方式--非线性网状方式,为信息管理、系统集成带来了新的思路。WWW 的广泛应用更推动了对开放式超媒体系统的研究。我们相信,在不久的将来,将会形成一个开放式超媒体的计算环境,将从不同方面与处理信息的各种应用程序紧密集成在一起。

参考文献

- [1] OHSWG Compendium, <http://www.cSDL.tamu.edu/ohs/>
- [2] Halasz, F., Schwartz, M., The Dexter Hypertext Reference Model, NIST Hypertext Standardization Workshop, February 1990
- [3] The COM Specification, Microsoft Corporation
- [4] Oterbye, K./Wul, U. K., The Flag Taxonomy of Open Hypermedia Systems; Hypertext'96 Proc., ACM Press, 1996