

基于 Dexter 的超媒体系统的基本要素 链、锚和成员

The Essential Factor of A Dexter-Based Hypermedia Systems: Links, Anchors and Components

熊瑞萍

Xiong Ruiping

(广西计算中心广西软件新技术实验室 南宁市星湖路 32 号 530022)

(Guangxi New Software Technology Lab., GuangXi Computing Center,

32 Xinghu Road, Nanning, Guangxi, 530022)

摘要 重点介绍了基于 Dexter 的超媒体系统中的几个重要对象类:链、锚和成员;详细论述了他们的构成、作用及扩展形式,对这些概念的清理与扩充,无疑有助于在开发一超文本系统时这些对象类的设计。总的说来,成员是基础、链是核心、锚是关键。

关键词 链 锚 成员 组合 会话 实例

Abstract The focus of this article is on the central object classes of a Dexter-Based Hypermedia system: Links, Anchors and Components, and describes in details their structure, action and extended form. Cleaning up and expanding the concepts is useful for developing a hypertext system. In short, component is basic, Link is heart, anchor is key.

Key words link, anchor, component, composite, session, instance

1 简介

Dexter Hypertext Reference Model (简称 Dexter) 是从一组典型的超文本系统里吸取了最好的设计思想,概括和综合了他们许多最好的特性而建立的一个独立的超越于数据和过程之上的模型,是一种衡量与比较超文本系统的标准,也是开发超媒体系统的依据和基础。目前有不少超媒体系统是以 Dexter 为平台开发的,如丹麦的 Farhus 大学计算机工程系在 Devise 项目中所开发的超媒体系统(简称 DHM),他们都不可避免地涉及到了链、锚、成员等基本的超文本概念。对这些概念的深入理解可以说是把握以 Dexter 为基础的超文本信息管理技术的本质的关键。本文分析了 Dexter 中这些概念的形式、类型及其本质,并提出了存在的问题及对其功能的某些扩展,以此为欲在其上开发超媒体系统的设计者提供一些参考。有关 Dexter 基本结构和构成可参看《Dexter—超文本系统的一种标准模型》一文。

2 链

链是超文本系统的核心,它将各种信息以一种直接的方式组织起来,为信息的管理和检索提供了一种有效的手段,由链构成的可往返移动的网络结构使得超文本与其它组织信息的方法区分开来。目前的超文本系统已以几种方法实现了链,不同的系统对链的定义和使用会不同,但在 Dexter 中均由链成员(Link Component)统一起来。在 Dexter 里,链是作为成员来处理的,链成员的内容是若干说明符的清单,并包括有一个表示说明及成员和锚的标识符。链的类型除了典型的源/目标链外,还有多头链,悬空链等类型。另外, Dexter 通过使用说明符还支持可算链(Computed Link)和静态链(Static Link);也可通过在链成员中增加属性来支持简单的“类型定义”形式。由于链是成员,所以又可作为其它链的端点。

Dexter 链存在的问题主要是对悬空链的限制和链方向性语义的定义。下面就这两方面加以讨论。

2.1 悬空链

所谓“悬空链”是只有一个端点的链。尽管原则

上 Dexter 的链可以少于两个端点,但模型的语义则明显限制。然而悬空链在许多情况下还是很有用的。首先,悬空链允许跟踪成员和锚删除的惰性修改(Lazy updating)和无用单元的回收,这在被删除(或修改)的链存在于另一个机器或正被另一用户锁住时是非常有用的。第二点,某些情形下会产生超媒体控制之外的数据对象(如:成员数据需要被移去或删除的文件),这时就必然造成悬空链的存在;第三,悬空链的悬空端点可被重新链接到另一成员或锚上而不必重新建立整个链,这对于多头链来说特别有用;最后一点,当所需要的端点成员或锚还不存在时,可特意建立悬空链作为位置占有者。在一个基于 Dexter 的超媒体系统里,以下四种情形会产生悬空链:

- 端点的成员已被删除
- 端点成员的锚已被删除
- 被成员内容关联的相关数据对象不可用
- 锚值无效

在前面两种情形,删除操作修改了对象,以致后来调用 Follow Link 产生异常。成员删除可通过清除锚清单和成员内容,并设置一“删除”标识来实现,锚的删除过程类似成员的删除。

后两种情形通常是超媒体控制的外部操作的结果。例如,如果成员的内容是文件标识符,而此文件已不依赖于超媒体地被移去或删除,也就是用户在超媒体系统之外对此文件进行了移去或删除操作,则构成成员内容的数据对象不可用,在这种情况下,进行 Follow Link 操作会获得文件系统异常的信息,并将其作为悬空链异常传送给用户。再如,当使用超媒体外部的编辑器对成员内容的锚值相关部分进行修改时,则由锚值说明的数据无效,但这种情形通常难以检测,因为锚值的查找/计算过程中不会产生异常。

2.2 链的方向性

在 Dexter 里,每条链的说明符一般是使用常量 FROM、TO、BIDIRECT 或 NONE 来指明方向的,具体使用哪个常量则依赖于端点的情况:源、目标,既是源又是目标,或者两者都不是。但每一条链都必须至少有一个说明符为 TO,也就是说每一条链都至少应该有一个目标,这也是 Dexter 模型的不足之处,从而限制了悬空链情形。另外,尽管 Dexter 模型是最早将方向值引入建立现存系统的方向性模型的,但对所有的方向值,以此模型为基础的系统设计者必须自己做出解释。通常对于值 TO 和 FROM 来说,含义是明确的,对于 NONE 和 BIDIRECT 则不太清晰,不同的系统会有不同的解释。如在前面提到的 DHM 将 BIDIRECT 用于表示同时为源和目标的端点,对于

NONE 则尚未给出一个明确的语义,但用它来标识在跟踪链时暂时不会被显示的端点。

然而,这些常量尚不足以反映现实中人们解释链方向性的各种方式。其中主要反映在链的三个方向上:

- 语义方向 这涉及到由链显示的各成员间的语义关系。如,一个连接成员 A 和 B 的“支持”链具有一个正常读取的方向;成员 A 中的变量支持成员 B 的申请。

- 建立方向 对应于被建立的链端点的顺序。所建立的第一个端点是链源,最后一个端点是链目标。

- 浏览方向 这个方向说明链是如何被往返移动的,如有的系统仅能从源到目标移动,有的则能双向移动。

原则上,链方向的这些语义是正交的。例如在某个方向上用户可以浏览一条链而不需要依赖链的语义方向,但许多系统还是强制实现相关性。如在 NoteCards 中,建立方向与浏览方向一致。

Dexter 模型并不明显地支持语义方向,但允许建立不同类型链的通用属性机制,也能够支持链语义方向的分配。

3 锚

将锚作为把网络结构连接到个别成员内容上的粘接剂是 Dexter 的主要贡献之一。锚是涉及到内部成员层的一种可控手段,也是一种间接寻址实体。没有锚,链就仅能连接到整个成员而无法连接到成员内的具体内容上。但使用 Dexter 模型中的锚有许多问题未解决,最大问题在于锚不是严格地与组合成员相关,这也就是说,在 Dexter 里,尽管一个组合成员的内容是可视的,但没有提到锚是如何与组合内基本成员相关的。因而,在使用 Dexter 开发超媒体系统时,必须对锚有所扩展以更好地利用锚。一个典型的例子就是 DHM 系统,首先是使用了动态的指针来代替锚,这意味着链说明符可以直接指向成员锚而不需访问函数。另外, DHM 还定义了三种锚类型:

- Whole-Component 锚支持未在成员内容内锚接的链的端点的退化(简化)情形。

- Marked(标识)锚是一个直接内嵌在成员内容内的对象,该对象在 Dexter 里称为链标志(Link marker),它可以是可见的或不可见的——实际上,某些链标志也许永远都不可见。在链标识区域内的命令提取会产生在对应的标识锚的链上的追踪。

- unmarked(无标识)锚无链标识,通常必须计算他们在成员内部的位置。

一般来说, marked 锚可以直接拾取, unmarked 锚则需要进行搜索, 每个 unmarked 锚都必须被询问是否是当前所选择的。

4 成员

成员分为原子成员和组合成员。组合是 Dexter 的一个重要概念, 反映的是若干个成员之间的层次结构, 并提供一种手段来捕捉信息的非链组织结构。在 Dexter 模型里, 组合的概念有计算组合 (Computed composites) 和虚组合 (virtual composites) 两种, 计算组合的内容是超文本上的一个结构化询问的结果, 询问返回的是所遇到的节点和链的集合。虚组合在运行时应用户要求而建立, 但不保存在数据库中。

下面我们主要讨论超媒体系统中成员内容的构成。

在超媒体系统里, 成员内容通常是由各种数据构筑而成, 可以包括外部数据对象或引用其它的成员。

4.1 原子成员

图 1 显示了一个原子成员和一个被锚接数据对象之间的两种可能关系。

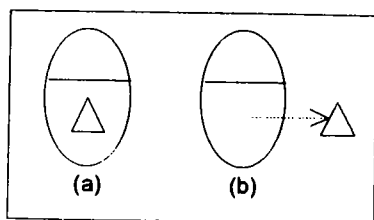


图 1 原子成员与数据的关系

Fig. 1 The relation between atom component and data

(a) 是传统情形, 在此, 成员和它的数据对象一起存储; 在 (b) 中, 包含在成员中的数据对象与成员分开存储, 并只能用成员的内容来引用。

一般情况下, 一个原子数据对象正好就属于一个原子成员, 这时用 (a) 的方式存储即可。但在两个或更多成员需要共享数据, 而这些成员又可能属于不同类型和有不同锚集的情况下, (b) 的方式更有效, 并可支持数据对象的多重视图。

4.2 组合成员

组合成员与数据对象之间的结构关系往往比较复杂。在 Dexter 中组合仅能包含被封装的数据对象, 这就显出 Dexter 模型的不足之处。一般来说, 组合需要涉及到外部数据对象或其它成员。图 2 给出了涉及到各种类型成员的组合的情形。

下面讨论的是常见的几种组合类型。

• 包含成员的组合

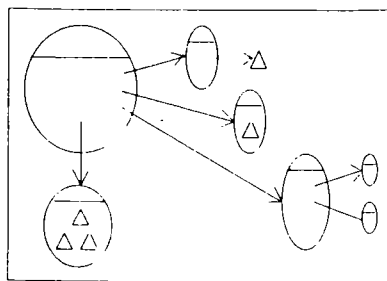


图 2 具有任意类型的内容的组合

Fig. 2 A composite referring to components of arbitrary type

如图 2 中所示, 组合的内容是其它的成员。

• 数据对象被封装的组合

这是一种直接与数据对象有关的组合, 如图 3 所示, 三角形描述的数据对象被封装在一个包含体对象中 (正方形所绘), 在这种情况下, 正方形对象的内部结构对于超媒体系统是可见的。因而, 组合和它的内嵌成员可同时与周围的对象及它的内部结构有关。

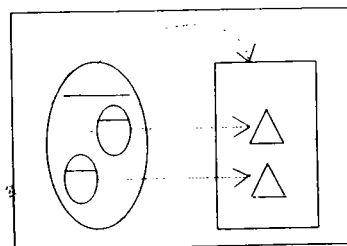


图 3 成员内容指向被封装的数据对象的组合

Fig. 3 Typed composite with nested components points at encapsulated data objects

• 结构化组合

这是一种具有结构化内容的组合, 如图 4 中使用具有由成员引用的键控表所构成的内容的组合。这种组合可与基本对象有关, 同时也与其它关系 (结构化组合成员) 有关。而且这种组合的实例能支持链到链

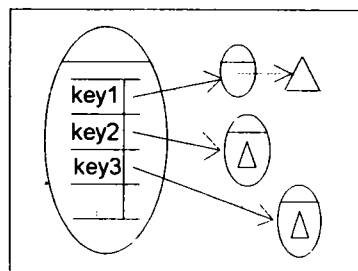


图 4 具有结构化内容的组合

Fig. 4 A composite with structure contents

端点的显示,这里的端点指的是被组合的封装结构所指向的成员。

如何将存储层的成员和内部成员层的数据对象统一起来,及如何将成员和组合的内部统一起来是在开发开放式的基于 Dexter 模型的超媒体系统时能够支持成员内容的大范围浏览所必需解决的问题。下表对成员内容作一个小结。

成员内容三要素

结 构	类型/位置	定 义
<ul style="list-style-type: none"> • 原子型 • 非结构化集合 • 结构化集合 —检索清单 —关键字表 —树 ... 	<ul style="list-style-type: none"> • 数据对象 —在成员内部 —在成员外部 • 成员 —具有严格类型的 —无限制的 	<ul style="list-style-type: none"> • 封装在此成员内 • 全局可见

5 结束语

Dexter 模型虽然已成为超文本系统设计的一个标准。但作为一个初步的抽象模型,还有许多方面需要完善,许多概念需要清理与扩展。本文所论述的问题仅涉及到链、锚、成员等主要超文本对象类的设计,可作为开发基于 Dexter 的超媒体系统时的参考。

参考文献

- 1 Kaj Grdnbaek Randall H. Trigg; For a Dexter-Based Hypermedia System. Communications of the acm, 1994, (2): 37.
- 2 Frank Halasz, Mayer Schwartz; The Dexter Hypertext, Communications of the acm, 1994, 37 (2).
- 3 John J. Leggett, Johnl. Schnase; Dexter with open eyes. Communications of the acm, 1994, 37 (2).

(上接第 70 页 Continue from page 70)

文本系统,它与 Dexter 标准模型是否一致?一种方法就是将实际超文本系统中的数据类型和函数映射到 Dexter 中的每一种数据类型和函数上,且充分论证,对于 Dexter 模型中的每一个抽象值在实际超文本系统中都有一个有效的表示,换句话说就是,在模型中可表达和可实现的每一件事在实际系统中都必须是可表达和可实现的。

但事实上,由于考虑到将来超文本系统的发展,Dexter 模型的功能远远超出目前超文本系统所必需的,主要表现在以下几个方面:

- 允许多路链接和链到链的连接
- 具有组合的概念
- 具有同时拥有显式源和目标的链

在这里,要解释的是组合这个概念,这里的组合不是指节点,链等原子的简单组合体,在 Dexter 模型里,这种组合体和节点,链一样,都属于基本成员。所谓组合是指包含基本成员及一些相关的描述成员性质而不是内容的成员信息,同时还包括一个能索引成员的锚系列、将成员显示给用户的表示说明和一系列任意数目的属性-值对。属性-值对可将任意性质附加到一个成员上,这种组合的概念是目前已有的各种超文本系统所不具备的。

6 结束语

Dexter 模型目前尚处于开发的早期阶段,许多方面还未能很好完善,但功能及结构定义已能满足并远远超过现存的超文本系统,一些概念,如 n 元链和组合成员的构造就是为了适应将来超文本系统的发展和设计的。因而,作为衡量及判断超文本系统的标准、Dexter 模型可以说是一种理想的模型,对于指导我们开发超文本系统具有重要的指导意义。

参考文献

- 1 Halasz F., Schwartz M. The Dexter Hypertext. Communications of the acm. 1994, 37 (2).
- 2 Aksyn R., McCracken D L, Yoder E A, KMS; A Distributed Hypertext for Managing Knowledge in Organizations Commun, ACM 31, 1988, 7.
- 3 Shneiderman b; Hypertext on Hypertext, Addison Wesley, New York, 1989.
- 4 Panel O T.; Interchanging hypertexts. In Proceedings of Hypertext 89, 1989, 5~8.
- 5 Dick L H., Bulterman C A, Guido Van Rossum; The Amsterdam Hypermedia Model, Communications of the acm, 1994, 37 (9. 2).