

Dexter: 一种超媒体参考模型^{*}

石教英 赵 斌

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

摘 要 目前超媒体系统的应用领域变得越来越广泛,但是这些超媒体系统在设计和实现的过程中往往出现了术语不统一和系统描述方法大相径庭的问题,因此 Dexter 模型的诞生为未来超媒体系统的设计起着参考和指导作用。本文首先对国内容易引起混淆且使用混乱的三个概念——超文本、多媒体和超媒体——予以明确的定义,然后详细介绍 Dexter 模型的三层结构,最后总结了 Dexter 模型的成功与不足之处。

关键词 超文本,多媒体,超媒体,Dexter 模型。

1988 年 10 月 John Leggett 和 Jan Walker 在美国新罕布什尔州的 Dexter 饭店主持召开了一个关于超媒体设计的研讨会,与会者都是一些至少曾参加过设计超媒体系统的专家学者。研讨会的主要目的是讨论如何统一超媒体系统的基本术语和语义描述。在这次研讨会上专门成立了一个 Dexter 研究小组并提出了一个具体化了的超媒体数据和过程模型,这个模型可以作为设计一个新的超媒体系统的参考标准。Dexter 研究小组的两个成员 Frank Halsasz 和 Mayer Schwartz 将这一参考标准模型进一步形式化并在 1990 年 1 月召开的 NIST 超媒体标准研讨会上宣读,将这个模型称之为 Dexter 超媒体参考模型,简称 Dexter 模型^[5]。

Dexter 模型将目前现有的超媒体系统抽象到一个较高的层次,对未来超媒体系统的设计起指导作用。Dexter 模型还统一了传统超媒体系统中使用不规范和容易引起混淆的术语,比如对传统超媒体系统中经常用到但常有争议的“节点”一词,在 Dexter 模型中抽象成“元素”这一术语。

一、超文本、多媒体和超媒体

超文本、多媒体和超媒体这三个概念从表面上来看都涉及文本、图形、声音、动画等多种媒体信息,在使用的过程中常常混淆,实际上超文本、多媒体和超媒体在本质上有根本的区别,图 1 给出了对三个概念比较直观的描述。

如图 1(a)所示,超文本是一种由元素构成且元

素与元素之间以一定的链接关系而组成的网络模型。超文本中的元素可以是文本、图形、声音、或者动画等之中的任何一种,但决不是多种媒体的组合。超文本可以看作是用于多种媒体信息检索定位的一种控制结构,它为在多种媒体信息间按内容检索定位提供了一种非常有效的方法。每一个元素的内部结构、元素与元素间的链接关系视不同的实现方案而不同。当用户指定要访问某个元素(从一定程度上讲,访问某个元素也就是以用户感觉合适的方式显示给用户),超文本系统就在它预先定义的结构中通

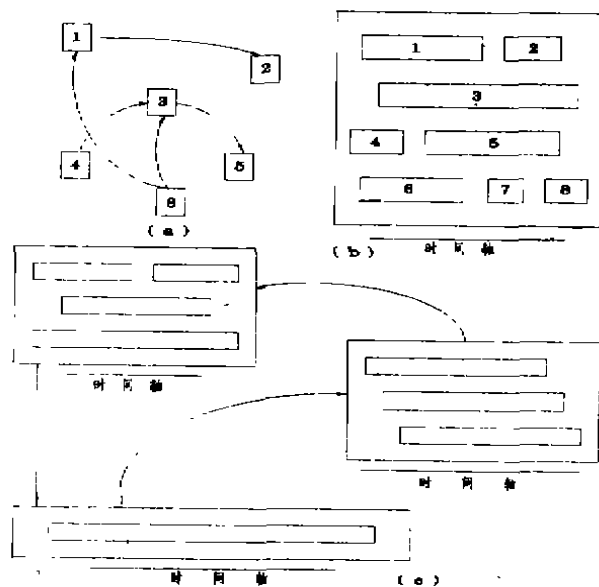


图 1 超文本、多媒体和超媒体的区别

^{*} 本文受到国防科工委跨行业预研基金资助,石教英 教授, 赵 斌 博士生。

过正确的链接关系找到(如果确实存在的话)用户指定的元素。

图 1(b)给出了多媒体的含义。多媒体为用户有效地表示信息提供了一种以多种媒体类型组成的机制。多媒体主要用于显示界面,也就是说在一个多媒体系统中显示给用户的信息不仅仅是文本格式,还加入了图形、声音、动画等多种媒体表示形式。例如在图 1(b)中媒体 1 和 2 可以是图形,媒体 3 可以是动画,媒体 4 和 5 可以是声音,而媒体 6,7,8 则可以是文本。

图 1(c)给出了超媒体的定义。超媒体是对超文本与多媒体结合的一个高层描述。从总体结构来看,它类似于超文本结构,也是由元素与链组成的,但是每一个元素的内部结构不再局限于文本、图形、声音或者动画之中的任何一种,而是多种媒体的组合,因此从单个元素来看是一种多媒体类型。超媒体系统的检索定位方法主要分为两层:第一层是元素与元素间的检索定位,这类检索定位方法基本上与超文本系统的检索定位方法相同;第二层是元素内部的检索定位方法,这类检索定位方法是在元素间检索定位的基础上,参照多媒体系统的检索定位方法而确定的。

二、Dexter 超媒体参考模型

如图 2 所示,Dexter 超媒体模型可分为三层^[5-6],运行时间层(run-time layer),存储层(storage layer)和元素内部层(within-component layer)和两个接口,定位机制(anchoring)和播放规范(presentation specification)。

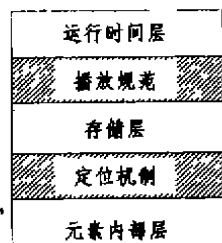


图 2 Dexter 模型的层次

2.1 存储层

Dexter 模型的关键是存储层,因为存储层描述了超媒体系统最基本的,也是最重要的元素间的网状关系。实际上存储层定义了由元素组成的数据模型。这里的元素是对超媒体系统中基本组成单元的

抽象描述。在各个不同的系统中仍采用不同的名称,例如在 NoteCards 和 HyperCard 系统中称元素为卡(card)^[1],在 KMS 系统中称元素为帧(frame)^[1],在 Intermedia 系统中称元素为文件(document)。在元素中信息可由文本、图形、声音、动画复合组成,因此可以看作是一个多媒体系统。对存储层的描述着重于定义元素间的连接关系,而不涉及元素的内部结构。重点讨论元素内部结构是元素内部层的任务,因此从存储层来看,一个仅包含文本的元素和一个包含文本、图形、声音、动画等多种媒体的元素是没有区别的,存储层把它们都当作是同一种类型。

在存储层的描述中,超媒体是由一个有限个元素组成的集合和两个函数组成的。

$$\text{Hypermedia} = (E_1, E_2, \dots, E_n, F_1, F_2)$$

其中 $E_1, E_2, \dots, E_n$ 表示有限个元素, F_1 和 F_2 是两个用于检索定位的函数,一个称为分解函数(resolver),另一个称为访问函数(accessor)。

在存储层中最基本的单元是元素。一个元素可以是原子单元,链,或者是由原子单元和链组成的复合单元,更为复杂的是由复合单元和链组成的复合单元。原子单元是存储层中最简单的单元,原子单元的内部结构在元素内部层讨论。

链是用于表示元素与元素之间关系的一种实体。一般情况下链是由两个或多个元素“节点”组成的点序列。在超媒体系统中,链是用于浏览整个超媒体网状结构的重要组成成份。在 Dexter 模型中,链的形式是多种多样的。最常见的链是由两个节点组成的,一个称之为源节点,一个称之为终节点。由于链是一种元素,因此一个链可以作为另外其它链的“节点”。Dexter 模型还支持多头链(multiheaded link)。多头链主要用于从一个元素同时检索到多个元素的情况。此外 Dexter 模型还支持少于两个节点的链,这种链称为悬挂链(dangling link)。由于 Dexter 模型支持悬挂链,因此在一个超媒体系统中不再需要定义起始链和终结链。

在 Dexter 模型中,每个元素都有一个唯一标识符,称为 UID。从整个超媒体系统全局来看,每个元素的 UID 都是不同的。访问函数的功能就是当用户指定某个元素的 UID 时能够在超媒体系统中成功地定位找到该元素。由于 Dexter 模型中元素的检索定位完全依赖于链,因此有的时候仅指定某个元素的 UID 并不能马上直接找到该元素,这个时候就需要一种分解函数起作用。分解函数的功能是当用户指定某个元素的 UID 而不能直接找到该元素时,需

要将目标元素的 UID 分解成由一个或多个中介 UID 组成的集合,这样访问函数就可根据中介 UID 集合中的成员找到目标 UID 指定的元素。

在存储层中还定义了一个操作集,它是由函数组成的,这些函数的功能主要是实时地对超媒体系统进行访问和修改。操作集中的操作包括在超媒体系统中增加一个元素,删除一个元素,修改一个元素的内容及其附加信息等。关于存储层的全部函数列出如下:

CreateComponent: 创建一个新的元素并把它加到超媒体系统中。

CreateAtomicComponent: 创建一个仅由原子单元组成的元素。

CreateLinkComponent: 创建一个链。

CreateCompositeComponent: 创建一个复合元素。

DeleteComponent: 删除一个元素。

ModifyComponent: 修改一个元素。

GetComponent: 当给定元素 UID 时,返回该 UID 所对应的元素。

AttributeValue: 当给定元素 UID 和属性时,返回该元素的属性值。

SetAttributeValue: 当给定元素 UID、属性和属性值时,将元素的属性设置成给定的属性值。

AllAttribute: 返回所有元素的属性集合。

LinkToAnchor: 当给定一个元素 UID 和该元素内的一个锚号时,返回所有指向该锚的链。

LinksTo: 给定一个元素 UID,返回指向该元素的所有链或由链组成的路径。

2.2 元素内部层

元素内部层定义了各个元素内部不同的内容和结构。元素内部层并不是 Dexter 模型所讨论的核心部分,而是必不可少的组成部分。在一个元素中其内容和结构是没有限制的,从内容上来说可以是文本、图形、声音、动画等任何一种可用的数据模型;从结构上来说,元素可由简单结构和复杂结构组成。简单结构就是每个元素的内部仅由同一种数据媒体组成,而复杂结构的元素内部又由各个子元素组成,而子元素的结构又与元素相同。这种嵌套结构的元素定义为描述复杂的混合类型的元素提供了灵活性和多样性。在 Dexter 模型中,元素内部层是开放的,也就是说 Dexter 模型对元素内部层的实现不作

硬性规定,可由应用程序根据实际情况作出灵活处理。

2.3 运行时间层

在存储层和元素内部层定义的数据及其时序与链结关系对用户来说是不可见的。运行时间层则为用户提供了一种可视可听的工具,它可直接访问和操作在存储层和元素内部层定义的网状数据模型。

在运行时间层中最基本的概念是例示元素(instantiation of a component),例示的过程实际上是将元素播放给用户。在实现的过程中,首先将元素的内容拷贝到缓冲区内,用户可编辑或浏览缓冲区内的内容。当缓冲区内的内容被改变后,运行时间层会定时将缓冲区内的内容备份到存储层。

2.4 定位机制

由于在 Dexter 模型中,描述元素间链接关系的存储层和描述元素内部结构的元素内部层是各自独立的部分,在检索定位的过程中就需要一个接口来维护从存储层到元素内部层或从元素内部层到存储层的检索定位过程。在 Dexter 模型中,介于存储层和元素内部层之间的接口称为定位机制。

定位机制的基本组成成分是锚(anchor)。锚由两部分组成:锚号(anchor id)和锚值(anchor value)。锚号是每个锚的标识符,锚值用来指定元素内部的位置和子结构。从存储层来看锚值是没有意义的,只有元素内部层的应用程序才能解释锚值。

锚号不同于元素的 UID。元素的 UID 从整个超媒体系统全局来看都是不同的,而锚号只是在各个元素内部编号不同。因此从整个超媒体系统全局来指定一个锚则必须使用(UID,锚号)对。

在 Dexter 模型中,锚号是一个相对固定的值,而锚值则是一个经常变化的值。由于超媒体系统的元素内部结构在运行过程中可能会改变,因此元素内部层的应用程序必须实时调整锚值的变化,保持它原有的指向。

锚与链的不同之处在于链仅指向元素,而锚则可指向元素内部的具体内容。

2.5 播放规范

在 Dexter 模型中介于运行时间层和存储层之间有一个重要的接口,称为播放规范。播放规范规定了同一数据呈现给用户的不同播放性质。比如在一个教学系统中,播放给教师看的习题可以带有答案,而播放给学生看的同一习题则没有答案;又比如说给一般用户播放的动画是不允许编辑的,而给系统程序设计者播放的动画则允许系统程序设计员对其

(下转第 16 页)

模型的框架,采用最基本的原子算子和最简单的语法集,达到最强的表达能力,便于进行数学的研究。

三、结论

上面我们分别介绍和比较了并发领域的三个重要模型 Petri 网、CSP、CCS,这三种模型各有千秋,在一些方面,某个模型可能优于其它模型,而在另一些方面,其它模型可能更方便。目前,既有局限于某一模型内的研究又有对三者关系的研究。值得注意的是,某个领域的成果对其它领域的研究具有启发和指导作用。

三种理论虽然比较成熟地解决了并发问题,但是由于它们都是抽象掉某些因素后简化了的模型,现实中还有许多与并发有关的问题尚未解决,而且并发理论本身复杂性太大,使得对它的进一步的理论研究和实践探讨相当困难,因此需要更多有献身

精神的科研人员投入到并发的研究开发中来。

参考文献

- [1] 陆汝铃,形式语义,科学出版社 1992
- [2] 周巢尘,通信顺序进程,北京大学出版社 1990
- [3] Robin Milner, Communication and Concurrency, 1989 published by Prentice Hall International (UK) Ltd 66 Wood Lane End, Hemel Hempstead, Hertfordshire, HP2. 4RG
- [4] Mogen Nielsen, CCS—And its relation to Net-Theory, Advances in Petri Nets 1986, Part I, Lecture Note in Artificial Intelligence 254 1986
- [5] Ernst-Rüdiger Olderog, TCSP—Theory of Communicating Sequential Process, Advances in Petri Nets, Part I, Lecture Note in Artificial Intelligence 255, 1986

(上接第 3 页)

进行编辑。

三、小 结

自 1988 年提出 Dexter 模型后,它对超媒体系统的设计和实现起着重要的指导作用。在设计新一代超媒体系统时 Dexter 模型是一种必不可少的参考模型。但是 Dexter 模型还处于一个早期阶段,它所定义的功能远比现有的系统强。特别是在以下两个方面:

- Dexter 模型支持多头链,而目前在超媒体系统中实现多头链还非常困难。

- 目前绝大部分超媒体系统还不支持复合元素。

因此在设计未来的超媒体系统时,要重点解决以下几个方面的问题:

- 应当设计支持复合结构的超媒体系统。
- 支持协作与共享的超媒体结构。
- 支持与时间有关的媒体。
- 能够支持其它类型应用程序的开放式超媒体系统。

参考文献

- [1] Akscyn, R., et al., KMS: A distributed hyper-

text for managing knowledge in organizations, CACM, Vol. 31, No. 7, 1988

- [2] Kaj Gronbæk, et al., Design issues for a Dexter-based hypermedia system, CACM, Vol. 37, No. 2, 1994
- [3] Kaj Gronbæk, et al., Cooperative hypermedia systems; a Dexter-based architecture, same to [2]
- [4] Halasz, F. G., Reflections on NoteCards, Seven issues for the next generation of hypermedia systems, same to [1]
- [5] Frank Halasz et al., The Dexter hypertext reference model, NIST Hypertext standardization Workshop, Gaithersburg, MD, January 16-18, 1990
- [6] Halasz, Frank et al., The Dexter Hypertext reference model, same to [2]
- [7] Lynda Hardman, et al., The Amsterdam hypermedia model; adding time and context to the Dexter model, same to [2]
- [8] John leggett et al., Viewing Dexter with open eyes, same to [2]