DOI:10.13656/j.cnki.gxkx.1995.03.019
广西科学 Guangxi Sciences 1995，2（3）：75～78

基于 Dexter 的协同超媒体系统设计
Design of A Dexter-Based Cooperative Hypermedias Systems

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　论述了基于 Dexter 超文本标准模型的协同超媒体系统的通用体系结构设计中的问题。该结构包括开发与 Dexter 兼容的超媒体系统的通用框架。其中的编辑进程、客户进程和 OODB 服务器进程分别对应于 Dexter 模型的各个层。OODB 提供了灵活的锁定和通告机制，这些机制使得 OODB 更为通用且可独立于基于 Dexter 的进程和数据模型而有所变化和扩展。文章最后给出了一些协同系统使用的典型例子。

关键词　超媒体　协同设计　进程　锁定　通告

Abstract　Issues for the design of a general architecture for cooperative hypermedia systems based on the Dexter Hypertext Reference Model have been discussed. The architecture includes a generic frame-work for developing Dexter-Compatible hypermedia systems. In the frame-work, the Editor process, Client process and OODB server process be designed correspond to the layers proposed by the Dexter model seperately. The OODB provides support for flexible locking and event notification mechanism. Developing such support within the OODB makes it general and independent of changes and extensions to the Dexter-based process and data models. At last, some examples of abstract use of scenarios are given.

Key words　hypermedia cooperative, design, process, locking, notification

1　概述

大型工程项目的计算机系统往往具有分布在不同时间、空间和硬件平台上等协同工作的特征，这种协同性对超媒体系统提出了一些要求，如共享数据库的建立、对共享材料的有效管理、系统的移植性、扩充性和剪裁性，各应用的集成性等。尽管有许多协同设计方面的关键问题未解决，如：如何支持几个用户共享超文本和成员，Dexter 各层如何与分布在超媒体结构上的多个用户相联系等，但 Dexter 仍不失为一个开发协同超媒体系统的基础模型和参考标准。Dexter 模型将系统的存储层和运行层明确分离开来，将内部成员层的结构对外开放，且能区分超媒体系统中所集成的各软件的职责，这些都为超媒体协同设计提供了一个稳定的框架。

1995-06-21 收稿。

2　系统在共享材料上的协同方式

一种协同超媒体也是面向对象的超媒体框架是以 Dexter 原理为出发点，由模拟 Dexter 不同层的一般类层次构造而成，通过定义一般类就可开发出一个特定系统。这其中，一个必不可少的组成部分是一个通用的基于 MBS 的面向对象数据库（OODB），它可用于所有独立于应用领域的对象类型，是协同超媒体系统设计的基础。

在大型的工程设计和创作中，往往涉及到各个部分设计之间的协调，这种协调可以两种方式实现：显式通讯调整；通过共享材料进行的隐式调整，而在共享材料上的协同性可有以下六种不同方式：

·职责分离　设计资料被分为不相交的部分，每一部分至多被一个用户操作。用户可观察被其它用户操作的部分。此处的协同关系是非常松散的，主要由使用其它用户所开发的设计的某个用户组成。

·轮流进行　类似职责分离方式，但每一部分可在不同的用户间改变，某一时刻只允许一个用户修改给定的部分。这种方式要求更多的支持以协调操作同样部分的用户。

·动态改变　在一个会话期间，用户可动态地改变各部分：用户 A 可能要修改正被另一用户 B 锁住的某一部分，则用户 A 向用户 B 发生请求，使得锁在会话期间传递给用户 A。

·可选择的版本　不同的用户可对同一部分产生不同的版本，然后在最后对这些不同的版本进行综合。

·交互对话　两个或更多的设计者可在同一时间工作在同一部分上，同时，一些直接通讯的通道是开放的，每一用户所做的全部操作都会立即在此部分的一共享拷贝上修改，协同提交功能会在合适的时候修改 OODB 中的内容。这种协同方式的一种变形是每一用户所做的修改不立即提交给系统，但在没有其它用户察看的情况下，可取消这种方式。

·完全同步会话　与交互对话一样，只是工作在同一部分的几个用户使用一个共享（全局）窗口，在这种方式，所有用户完全共享资料的同一视图。

在以上这些方式里，超媒体开发的重点主要是放在支持共享材料上的隐式和异步协同关系，从职责分离到动态改变是超媒体系统支持的典型的异步协同方式。这些方式需要告诉用户谁正在共享资料上做什么。资料的各个块用链连接，协同可通过其它用户所开发的部分上的注解来进行。这些方式要求一种灵活的借助于存储超媒体对象的底层数据库的锁定方案，这就是 OODB。可选择的版本方式也是一种异步方式，对这种方式的支持是为了能够共享某些类型的资料。交互对话和完全同步会话方式表达的是共享材料上协同关系的同步方式，他们的主要区别在于是否维持一个共同的视图，在交互对话里，几个用户可以编辑超媒体中的同一成员而不必维持同一视图；同步会话里，所有的用户都拥有正在编辑的超媒体成员的同一视图。这些同步方式要求扩展超媒体系统以支持对 OODB 的改变的共同提交部分。

3　协同超媒体系统基于 Dexter 的体系结构

在协同超媒体系统里，以进程的形式对应于 Dexter 模型中的各个层，图 1 给出了与 Dexter 模型各层次相关的进程关系。

图中主要有以下三种进程：

·编辑进程　这些进程是用超媒体集成的最终用户编辑器，包括文本编辑器、图形编辑器、图像编辑器和超媒体浏览器。一个编辑器考虑一种特定数据对象（如构成文本成员内容的文本对象）。编辑器可将数据对象存贮在 OODB 之外的独立文件中，编辑器亦可用来编辑 OODB，被这些编辑器操作的数据对象属于 Dexter 模型中的内部成员层，编辑器的功能也主要是属于内部成员层的。编辑器的超媒体功能仅仅通过与相应的运行实例对象的通讯而存在。一个编辑器代表了特定类型成员的内容的运行管理，支持与运行进程通讯的协议。超媒体浏览器则借助于组合来实现，并可编辑组合的内部成员层对象。

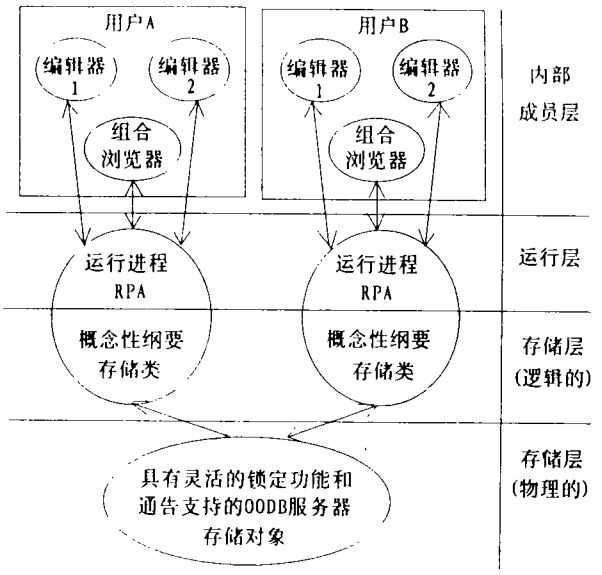

图 1　协同超媒体系统结构

Fig. 1　Architecture of cooperative hypermedia systems

·运行进程（RP）　运行进程（RP）为一系列用户正在使用的编辑器进程提供超媒体服务，运行进程负责处理链、锚和在运行时间里的成员，它是一个与编辑器通用的服务器，也是一个 OODB 服务器的客户。RP 创建了由实现 Dexter 运行层概念的一般类及特定类定义的对象的实例，为创建和处理、实现 Dexter 存储层概念的成员和链对象提供了与编辑的无关的操作，各运行进程亦负责将从 OODB 服务器接收到的事件通告分送到各编辑器上。

·OODB 服务器　OODB 服务器进程为超媒体对象提供了永久的物理存贮，被存贮的对象是从实现 Dexter 存贮层概念下一般类特别指明的各个类的实例。

4　系统在不同平台上的分布

在一个分布式环境里，OODB 服务器和 RP 可运行在不同的计算机上，对于超媒体的每一个活动用户

来说都有一个活动 RP，RP 是 OODB 服务器的一个客户，共享运行在不同硬件平台上的客户间的超文本是可能的，存贮在 OODB 中的对象结构可共享，但对内部成员层上数据对象的共享则依赖于编辑不同平台上同样类型内容的对象的编辑器的可用性。

RP 和编辑器原则上可运行在不同的计算机上，但实际上它们通常运行在用户的工作站上。为了用超媒体支持协同设计和创作，用户需要协调他们在共享资料上的工作。技术上，这种协调是由 OODB 服务器发布的事件通告来支持的。

5 OODB 服务器的事件通告机制

OODB 服务器能够通知它的客户有关存贮对象上的事件通告，OODB 服务器的客户可以预约各种类型的事件。下面给出了 OODB 支持这些功能的方式。

• 协同支持：基于 OODB 的事件通告和灵活的锁定　有两种支持，一种是对在用户中建立的有关共享资料上发生事件的洞察；一种是对资料各部分的交换的支持。

• 给 OODB 客户的事件通告　事件通告机制意味着用户借助于编辑器或浏览器可对发生在共享资料上的各种事件进行预约。RP 会向 OODB 服务器发出请求，以获得有关与其它用户相关的其它 RP 所产生的对象的改变。例如，在给定情况下，几个用户可访问同一超文本，如果一个用户改变了超文本中的一个成员，则这些改变对于那些用读访问打开这个成员及预约了这些变化的通告的用户来说是可见的，在一个特定的超媒体应用里，某些事件类型的预约可自动进行，也可由用户手工进行。

• 对象访问与锁定　OODB 提供了对象访问和锁定机制来支持超媒体的开发，这实际上也是另一种形式的通告机制，目前的锁定值仅有读和写两种。并通过以下几种操作来实现：建立（Create）操作告诉数据库如何存贮一个对象及其传递方式，此时客户在所建立的对象上获得一个写操作；获取（Get）操作可从 OODB 服务器中检索出一特定对象并使用，此时客户获得的是在读对象上的读锁定。

如果一个对象改变了，那些改变会存贮在数据库中，更新（Update）操作可在数据库中该对象的实体上进行修正，然后将最终结果存贮。在此期间，更新操作会告诉数据库将此对象的每一变化都保存下来。更新操作仅能用在具有写锁定的对象上，如果仅具有读访问的对象进行了更新操作，则会产生异常。

对于已从数据库中检索出来的对象，亦可动态地改变其锁方式，这可通过使用换锁（Changelock）操作来实现，具有比当前锁方式更高许可权的锁方式改变意味着可隐式地提取被锁住的对象；许可权较低的改变则意味着设置了一个隐式检测点，可检测这个对象是否已被改变。如果一个写锁被废弃，则另一个客户可获得写锁，改变对象，并将这些改变提交给 OODB。

6 协同超媒体系统中 OODB 的作用

在协同超媒体系统中，由于 OODB 的存在，而使得运行进程可将超文本作为永久对象进行存贮和检索。系统里的超文本，成员和锚均具有如谁是创建者，谁是最后一个修改者等信息的属性，也具有说明存贮对象是禁用的，是属于某一组的还是属于某个特定用户的信息的属性，这些属性允许一个超文本的会话有选择地仅仅表示当前用户想要使用或有权使用的对象。OODB 的基本事件通告机制使得以下事件成为可能：

• 建立、删除或更新整个超文本；

• 建立、删除或更新超文本的成员（原子、链或组合）；

• 建立、删除或更新成员内的锚和属性；

• 锁住超文本和成员上的改变。

也可以对有关事件的启动和其它 RP 提交的或夭折的活动事件的通告进行预约，这种事件通告可使用户了解共享超文本的状态及其内容的变化。

通告可以用户所选择的形式（如图示，声音）出现在用户界面上，这使得用户能够监控同一超文本上后面的用户的活动，用户可检查通告登记，亦可检查超文本和成员的属性以获得有关对象修改的信息；获得了超文本上写锁定的用户可以在一个事件期间修改超文本。为了记录这样的改变，其它的用户通过他们的 RP 可预约被超文本的其它用户引起的对象检索、建立、更新和访问等变化的通告。其中，对象检索可通过读访问或是写访问进行，这种锁定信息亦是事件通告的一部分。

预约可用于特定对象或整个对象，并可通过一成员浏览器或一特定编辑器进行。

7 协同超媒体系统使用情况的例子

上面给出了协同超媒体系统中的一些基本概念与机制，下面给出几种协同系统使用的具体例子，以便于读者对协同系统使用情况的了解。

超媒体系统的协同方式可有以下几种：

• 立即更新　甲和乙都启动了同一超文本 H1 上的一个会话：甲获得的是成员 C1 上的写锁定；乙

则用读访问打开 C1 并预约当 C1 被其它用户修改时立即更新。此时，不论何时甲对 C1 作了修改，这些修改都会提交给数据库，C1 的实例会通过重新获得存贮在 OODB 中的 C1 的最新版本来重新自动更新自己，并使更新后的内容出现在乙的屏幕上。

• 登录事件　几个用户（甲、乙、丙）已启动了超文本 H1 上的一个会话：甲用一读访问打开 C1 并预约对 C1 的改变进行登记；乙用写锁定打开 C1，此时，甲的控制台上会出现一条消息："某年某月某日某时，乙用写方式打开 C1"。当乙修改并存贮 C1 后，在甲的屏幕上又会得到一条消息："某年某月某日某时，乙修改了 C1"。与此相似，当乙释放写锁，丙又用写锁打开 C1 时，甲均会收到相应的消息。

• 对超文本了解的通告　甲和乙均启动了超文本 H1 上的一个会话，预约有关谁使用了 H1 的通告，则控制台上会显示已执行了 H1 上"Get"操作的用户名单，如果丙此时也启动 H1 上的一个会话，则甲和乙的控制台上会出现丙的标识。

• 成员和组合成员的了解通告　几个用户工作在由甲负责管理的同一情形下，他们在对应的超文本 H1 上启动了一个会话。甲产生一包含成员 C1、C2、C3 和 C4 的组合体 CSI，作为当前的活动文档。甲使用 "Composite subscribe……"（组合预约）菜单命令来预约发生在 CSI 包含的成员上的变化的通告。此时，无论何时另一个用户执行了在 C1、C2、C3 和 C4 上的更新锁定等操作，甲都会被告知。

• 关于特定对象的建立/删除通告　几个用户启动了超文本 H1 上的会话，甲预约登记 H1 上文本成员的建立，乙为 H1 建立了一个新的文本成员 C1，编辑其内容并存起来，此时，甲从控制台上获得消息："乙建立了文本成员 C1"。

• 锁交换　几个用户启动超文本 H1 上的一个会话。甲获得成员 C1 上的写锁定；乙和丙用读访问打开 C1；丙预约登记 C1 的变化。在某一时刻，乙使用菜单命令"改变锁……"，这个菜单项会通知他，甲在 C1 上拥有写锁定，然后，乙通知甲并问他是否要将其改动存起来并释放 C1 上的写锁定。甲同意这样做，并存起改动，将写锁改为读访问，于是乙立即可获得写锁。在此交换期间，丙会获得消息："甲已存起改动"、"甲释放 C1 上的写锁"、"乙已获得 C1 上的写锁"。甲然后预约登记 C1 上的所有变化。乙于是做出一些修改并将它们提交给 OODB，同时触发送给甲和丙的通告消息。

同步连接　甲和乙启动了超文本 H1 上的一个会话，用读访问打开文本成员 C1，并预约 C1 上的立即更新。甲建立了从 C1 的文本区到成员 C2 的文本区的公共链，甲提交这个变化，对 C1 的锚清单产生一短暂的写锁定，立即用新的链标识更新乙的 C1 视图。乙在 C1 和成员 C3 之间产生了另一公共链，并提交这些改变。甲的 C1 视图也以同样的方式立即更新。

以上例子说明了从基于 Dexter 超媒体结构开发出的系统所支持的几种协同类型。

8　结束语

基于 Dexter 的框架和结构已成为在大型工程项目中进行协同超媒体系统开发的基础，但这个框架和结构也还需进一步扩展和完善，包括某些特定超媒体原型系统的实现。

参考文献

1　Gronbeak K, Hem J A., Cooperative Hypermedia Systems: A Dexter-Based Architecture Communicaions of the acm. 1994, 37 (2).

2　Gronbeak K, Trigg R H. Design issues for a Dexter-based hypermedia system. Commun. acm 37, 2.

3　Ahmed S., Wong A., Sriram D., Logcher R A. comparison of object-oriented database management systems for engineering applications. Res. Rep. R91-12, 1991.