

科技部科技基础性工作专项资金重大项目 研究成果

项目名称：我国数字图书馆标准规范建设

子项目名称：复合数字对象规范

项目编号：2005DKA43503

研究成果类型：研究报告

成果名称：复合数字对象规范调研报告

成果编号：[暂缺]

成果版本：征求意见稿

成果提交日期：2007-07-20

撰写人：张俊娥（CALIS 管理中心/北京大学图书馆）

郑小惠（清华大学图书馆）

曾燕（中国科学院图书馆）

周静怡（中国科学院图书馆）

项目版权声明

本报告研究工作属于科技部科技基础性工作专项资金重大项目《我国数字图书馆标准规范建设》的一部分，得到科技部科技基础性工作专项资金资助，项目编号为 XXXXXXXX。按照有关规定，国家和《我国数字图书馆标准规范建设》课题组拥有本报告的版权，依照《中华人民共和国著作权法》享有著作权。

本报告可以复制、转载、或在电子信息系统上做镜像，但在复制、转载或镜像时须注明真实作者和完整出处，并在明显地方标明“科技部科技基础性工作专项资金重大项目《我国数字图书馆标准规范建设》资助”的字样。

报告版权人不承担用户在使用本作品内容时可能造成的任何实际或预计的损失。

作者声明

本报告作者谨保证本作品中出现的文字、图片、声音、剪辑和文后参考文献等内容的真实性和可靠性，愿按照《中华人民共和国著作权法》，承担本作品发布过程中的责任和义务。科技部有关管理机构对于本作品内容所引发的版权、署名权的异议、纠纷不承担任何责任。

《我国数字图书馆标准规范建设》课题组网站（<http://cdls.nstl.gov.cn>）作为本报告的第一发表单位，并可向其他媒体推荐此作品。在不发生重复授权的前提下，报告撰写人保留将经过修改的项目成果向正式学术媒体直接投稿的权利。

目 录

1. 背景	4
2. 简述	4
3. OEB——电子图书数字对象	5
3.1 OEB对象的组成	5
3.2 OEB的元数据	6
4. EAD——档案编码描述标准	6
4.1 EAD 简介	6
4.2 EAD的适用范围、描述对象的界定、著录对象和著录单位	7
4.3 EAD规范的使用者	8
4.4 EAD规范的结构	8
4.5 EAD文件类型定义（EAD DTD）	8
4.6 EAD 标签库(EAD TAG LIBRARY)	10
4.7 EAD的应用规则	12
4.8 EAD评价	12
5. SCORM——内容可共享的数字对象	13
5.1 内容模型	13
5.2 内容包装	13
5.3 元数据	15
5.4 评价	15
5.5 附录 1：IEEE LOM元数据元素	15
6. METS	16
6.1 METS对象的组成	17
6.2 METS对象各组成部分之间的关系	18
6.3 评价	21
7. FEDORA	21
7.1 FEDORA简介	21
7.2 FEDORA数字对象模型	21
8. MVD——多价文档	24
8.1 MVD概述	24
8.2 MVD结构：层和行为	24
8.3 MVD文档树	25
8.4 MVD批注功能	25
8.5 MVD类型图	26
8.6 MVD评价	26
9. 参考文献:	27

1. 背景

在数字图书馆时代，通过将多种类型的信息有机地组织起来，同时封装完善的行为方法与元数据，建立内容集成、结构复杂的数字信息对象，即复合数字对象，可以全面揭示信息存在的多样性、复杂性特征，从根本上解决信息组织与服务之间的矛盾。

复合数字对象在结构特征，功能定义，描述方法，组织原则以及服务应用等方面都提出了新的思想。一篇文献是一个对象，一种服务是一个对象，一个系统是一个对象，而一个对象集合还是一个对象。复合数字对象是针对数字图书馆多样化需求而产生的，它改变了传统的信息组织观念，必然导致信息组织本身的性质，目标，内容以及工具的变化。在当前的数字图书馆建设中，既要重视复合数字对象的基本理论研究，同时要加强应用技术研究与实践。

本课题旨在研究主流复合数字对象的标准规范，提出推荐方案及其应用指南。

子任务将对数字媒体、数字教育、数字科研、数字出版、数字图书馆等领域的复合数字对象描述与封装标准规范的现状与发展趋势进行分析，针对数字图书馆建设的需要，针对数字科研、数字教育、数字出版、数字媒体和数字图书馆的集成汇聚的趋势，提出我国数字图书馆建设中复合数字对象标准规范的推荐方案及相应的应用指南，并提出必要的测试实验要求。

2. 简述

下表中是国内外研究较多的复合数字对象模型，本课题着重调研这些复合数字对象的描述和封装标准规范。

名称	简要说明
METS	METS (Metadata Encoding and Transmission Standard) 是在 MOA2 的工作基础上发展而成，不同的是，METS 中增加了对音频视频对象的支持，在形式上，使用 XML SCHEMA 取代了 DTD。
OEB	电子图书数字对象，着重于对象的兼容性，交互性与可移动性研究，实现发现资源，内容过滤，阅读方向导引等功能
VEO	面向保存的数字对象，采用封装技术，实现数字对象的文档自构，自足以及系统无关性
FEDORA	The Flexible Extensible Digital Object and Repository Architecture, 面向信息集成数字对象。
SCORM	THE Sharable Content Object Reference Model, 内容可共享的数字对象
MVD	Multivalent Documents, 支持批注与协作写作的数字对象
MPEG-21	Moving Pictures Experts Group , MPEG-21 提供了异构元素组合的描述语言和标准
CDR	Compound Document by Reference Framework, 定义了一般的基于引用的复合文档描述框架，为整合任意文档合适定义了一种独立的语言处理模型
EAD	Encoding Archival Description 是国外档案界处理档案信息最常使用的元

数据标准。

3. OEB——电子图书数字对象

1998 年成立的开放电子图书论坛是由 IT 界、图书馆、出版发行机构共同发起的电子图书标准制定组织，该组织在 1999 年发布了电子图书框架结构标准 OEBPS1.0,其目的是为了更好的表现电子书的层次内容，为电子图书系统开发和制作者提供电子书文档结构的标准。2002 年 7 月，OEB 论坛发布了 OEBPS1.2 版本，是目前仍在使用的当前版本。

3.1 OEB 对象的组成

OEB 规范主要适用于一本电子书的制作过程，由于一本书的结构本身是个复合对象，书由封面信息，扉页，目录，章节，封底等组成，在制作电子书的过程中，要遵循一本书的物理结构，有描述该书的书名，作者等描述信息，有一本书所包含的所有页（文件）信息，有页与页之间连接顺序的信息，也有特定页的突出显示信息。根据一本书的这些特性，标准规定了一个符合上述特征的 xml 文档。该文档描述了一本电子图书的完整框架，包括文档、图象和其他对象以及他们之间的相互关系。根据规范的要求，该文件由 metadata(元数据)、manifest(文件清单)、spine(书脊)、tours(导读)、guide(指南) 构成。

例如：

```
<?xml version="1.0" encoding="utf-8"?>
<package unique-identifier="01023190"
xmlns:oebpackage="http://openebook.org/namespaces/oeb-package/1.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <metadata><dc-metadata xmlns:dc="http://purl.org/dc/elements/1.0/">
    <dc:title>钦定全唐文</dc:title>
    <dc:creator>(清) 董诰 编校</dc:creator>
    <dc:creator>(清) 戴衢亨 编校</dc:creator>
    <dc:subject>钦定全唐文</dc:subject>
    <dc:publisher>武英殿</dc:publisher>
    <dc:type>古籍</dc:type>
    <dc:format>Image/Djvu(.djvu)</dc:format>
    <dc:identifier>01023190</dc:identifier>
    <dc:language>chi</dc:language>...
  </dc-metadata>
</metadata>
<manifest>
  <item id="content1"
href="ptiff\00000001.tiff"media-type="text/oeb-document"/>
  <item id="content2" href="ptiff\00000002.tiff"
media-type="text/oeb-document"/>...
<spine>
```

```

        <itemref idref="content1"/>
        <itemref idref="content2"/>...
    </spine>
    <tours>
        <tour id="tour1" title="欽定全唐文目录">
            <site title="卷七百九十三" href="ptiff\00000006.tiff" />
            <site title="卷七百九十四" href="ptiff\000000054.tiff" />
        </tour>
        <guide><reference type="cover" title="封面"
href="ptiff/00000001.tiff"/><reference type="Other" title="封底"
href="ptiff/00000102.tiff"/>
        </guide>
    </package>

```

3.2 OEB 的元数据

metadata 部分包含电子图书的描述元数据，例如题名、作者等。电子图书的描述元数据已在标准规范一期项目中完成，可以直接引用。manifest 元素包含的是组成电子书的所有文件列表。子元素 item 用于描述文件类型和地址，每个 item 都有唯一的 ID 号标识。例如<item id=content1" href="ptiff\oooooooo1.tiff" media-type="text/oeb-document"/>表示这本电子书的第一页文件。保存路径为 ptiff\00000001.tiff, 媒体类型是 text/oeb-document。spine 元素按顺序从 manifest 列表选取部分媒体类型为 text/oeb-document 的文件，提供书籍的一种线形阅读次序。Idref 属性值对应 manifest 列表中目的文件元素的 id 属性值。如 <itemref idref="content1"/>表示书籍的首页对应 manifest 列表中 item 元素的 id 属性值为 content1 的文件，即 ptiff\00000001.tiff。tours 元素可根据读者水平或目的，按一定次序，选择电子书的部分页面组成导读。guide 元素包含多个 reference 子元素，子元素的 title 属性值依次列出电子书的特定页面，例如封面、目录页、序言等，href 属性值指向文件保存地址。

目前 OEB 标准在电子书行业里多有应用，国内的中美百万册书计划是遵循 OEB 标准来实施。清华大学中文数字图书馆项目中的电子书结构也是按照 OEB 标准来进行封装。由于电子书本身具有的特性以及它的单一性，使这个标准在本行业的开放标准中占有重要地位。

4. EAD——档案编码描述标准

4.1 EAD 简介

EAD (Encoding Archival Description) 是国外档案界处理档案信息最常使用的元数据标准。

EAD 的前期项目始于 1993 年的美国加州大学伯克利分校图书馆的伯克利检

索工具计划，最初受联邦教育部 IIA 项目的资助，开发的目的是建立一种通用于档案馆、图书馆、博物馆各种馆藏记录的数字化网络应用。1995 年该项目被正式称为 EAD。

1995 年 9 月，美国档案工作者协会(SAA)在华盛顿召开了年度会议。小组成员开始研究并确定检索工具编码标准专业使用与维护的恰当方法，并把设计原理和修改的数据模式送交到 SAA 的档案信息交流委员会(CAIE)。CAIE 应邀正式加入 EAD，并建立 EAD 工作组。SAA 正式提议在 EAD 经终审成为一项标准后，由美国国会图书馆网络发展与 MARC 标准办公室(ND / MSO)负责 EAD 的维护。

EAD1.0 版发布之前，采用的是 SGML 编码技术，现在 EAD 2002 Schema 也已经面世，采用的是 XML 及其相关技术。

4.2 EAD 的适用范围、描述对象的界定、著录对象和著录单位

EAD 的目标是让很多机构的档案资源可以为公众存取。为了实现该目标，EAD 将广泛包容国际上不同的著录实践，该标准将适用于不同档案管理机构的需求。

EAD 描述了开放的档案资源信息，它并不是支持档案馆藏管理(包括档案移交、保存、展示、利用、储存和技术加工等各个环节)的系统。

需要强调的是，EAD 是数据结构标准而非内容著录标准。它没有描述每个数据元素取值的规范性要求——这通常是国内或国际内容著录标准的功能。EAD 的标签库说明了数据类型，这些数据类型有可能与特定的内容标准中的某个元素相对应。与此同时，人们应该确保与外部标准这种相适用性。

EAD描述对象既可以是一份文档、一个案卷，又可以是一个类别、一个全宗的组成部分，几乎适用于每个级别内容信息的著录。下图显示了EAD描述对象的成分及不同成分之间的关系。

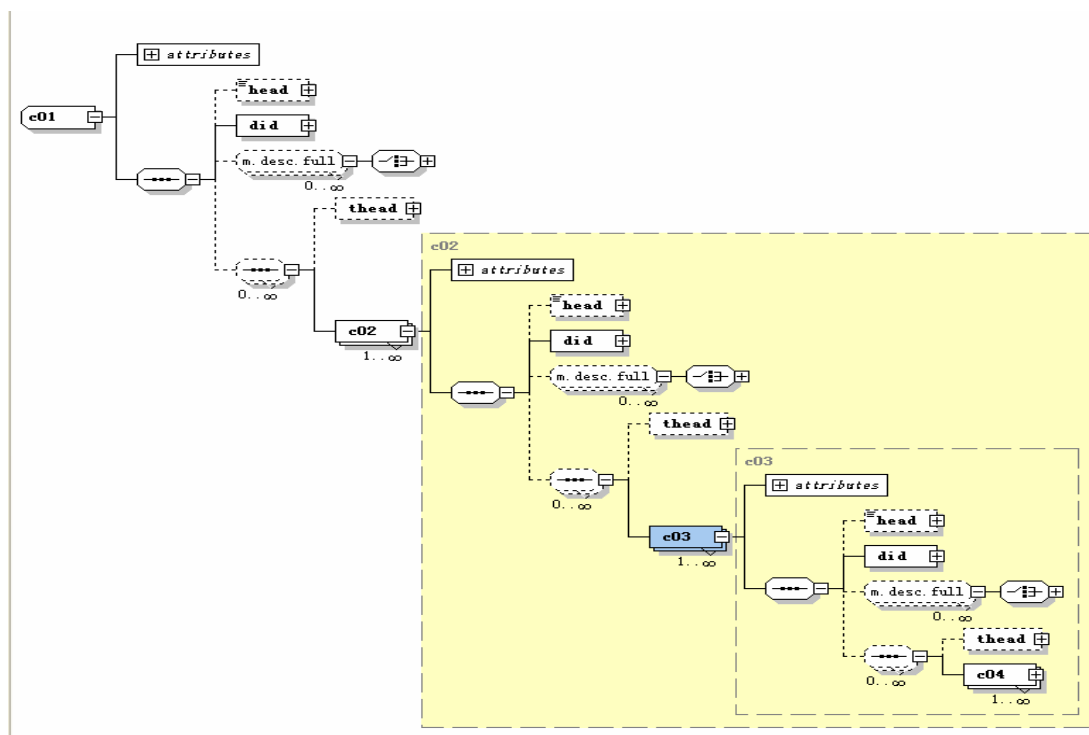


图 EAD 描述对象结构图

图注：EAD 描述的对象可以分为不同成分(component，简称为 C)，从 C01, C02,直到 C12(因篇幅所限，上图只显示到 c04 级)，代表了不同层级对应的描述对象。C01 至 C12 对应的档案描述层级可以是如下层级的一种：Collection 馆藏/特藏、Fonds 全宗、Class 类、Recordgrp 文件组合、Series 集丛、Subgrp 子群组、Subseries 副集丛、File 卷宗、Item 文件、Otherlevel 其他层级。

4.3 EAD 规范的使用者

EAD 规范的使用者主要是档案界，不难看出，EAD 规范描述信息非常之多，描述信息的层次性也非常显著，它充分反映了档案著录的来源原则、尊重全宗原则和反映管理单位的各层次原则。

4.4 EAD 规范的结构

EAD 更加侧重于编码（Encoding），因此，它不仅定义了元素(element)和元素的属性，还定义了相应的标签（tag）。

EAD 2002 共包括 3 份文档，分别是 EAD schema, EAD 文件类型定义（简称 EAD DTD）, EAD 标签库（包括元素与属性的定义，简称 EAD Tag Library）

4.5 EAD 文件类型定义（EAD DTD）

EAD DTD 将整个 EAD 元素集合定义为若干层次的元素组合，高层元素有 3

项：EAD 头标、前项和档案描述。每一高层元素之下再细分为若干元素，某些分元素再分为若干子元素。整个元素集都包含在<ead></ead>中，构成 EAD 文件。

(1) EAD 头标(eadheader)

EAD 头标(eadheader)为必备项，作为 EAD 或检索工具的识别。此项目以“文本编码创始项目”即 TEI(Text Encoding Initiative)为原型，主要用于抽取有关档案的产生、修订、出版与传播的有关信息。其下属元素包括四个分项以及若干子项：

表 1 EAD 头标的下属元素表

EAD 标识符	提供唯一的标识号
文件描述	提供文献作者、题名、版本、出版者等信息
范围描述	记录由谁在何时用何种语言创建了该 EAD 文件,包括档案创建信息需要和档案使用语言
修订描述	记录有关 EAD 文件版本变动的信息

文件描述还包括以下子项：题名叙述，包括作者、题名、副题名、赞助者等项目；版本叙述，出版项叙述，包括地址、日期、数量、段落、出版者；丛刊叙述；附注叙述。

EAD 头标可用于产生电子形式或印刷形式的题名页。在 DTD 中，对 EAD 头标中元素和子元素在 EAD 文件中出现的顺序做了详细规定，不可更改，以利于作为检索工具使用。

(2) 前项<frontmatter>

EAD 的前项为可选项。由于 EAD 头标中各元素在 EAD 文件中必须按规定顺序出现，为使各个档案馆生成符合各自需求的、要素顺序不同的题名页，EAD 定义了前项，主要的项目以题名页为主，结构为：题名；馆藏数量；出版者；记载档案馆的名称；列表；可用若干<item></item>标签对，描述需要在题名页给出的其他信息，如地址、电话等。

(3) 档案描述<archdes>(archival description)

EAD 档案描述是对档案内容和相互关系信息的具体描述部分，为必备项。这些信息以分层方式组织一起，包括文件的内容、上下文关系以及可选增补信息，其中有以下 12 个要素：

表 2 EAD 档案描述包括的要素表

描述标识符<did>(descriptive identification)	必备项，提供描述档案时的信息项目
附属描述资料<add>(adjunct description data)	可选项，提供附加的描述信息，可协助档案的使用，但并非档案描述本身
管理机构信息 <adminifo>(administrative information)	有关机构采集、处理、管理的信息；
编排<arrangement>	

传记/历史<bioghist>(administrative information)	
检索控制<controlaccess>	主要适用于网络环境下检索时的控制
数字档案对象<dao>与<daogrp>(digital archival object 与(digital archival object group)	使检索工具可以指向或者关联到已数字化的档案
附注<note>	
其他描述资料<odd>(other description data)	
组织<organization>	
范围与内容<scopecontent>(scope and content)	
附属成分的描述<dec>(description subordinate components)	为档案管理人员针对某一特定馆藏做进一步描述时使用

描述标识符还包括 11 小项：摘要；容器（如箱、盒、案卷、卷等）；数字档案对象；附注；来源，如档案的出处、产生者、收集者、处置者等；物理描述，包括档案的尺寸大小、实体的特点等；物理位置，描述档案在库房中的位置，档案架的位置；档案库；单元日期；单元标识符，有关档案在馆内的检索号，是唯一永久的辨别号；单元题名；

附属描述资料包括：书目、文件计划、索引、附注、其他检索工具、段落、相关资料、个别资料；

管理机构信息包括：访问限制，注明何时档案可以开放供公众利用；自然增加；编纂信息；可选利用形式；鉴定、典藏的历史；附注；段落；首选引用；处理信息；使用限制，与访问限制不同之处在于利用时不可违反与著作权有关的规定，如复印的规定等；

检索控制细分为 10 项：团体名称；家族名称；功能；类型形式；地理名称；名称；职业；个人名称；主题；题名；

附属成分的描述，首先对文件类型属性描述；分析性概述，用于某一文件系列或附属系列的描述；内部构成，用于某一档案装具或案卷，或某些项目的目录；综合，用于某一文件系列的描述，在其后紧接着档案装具或案卷的目录，是上述两项的综合；其他类型，用于描述以上三种属性之外的类型。

4.6 EAD 标签库(EAD Tag Library)

EAD Tag Library 一共定义了 146 个元素和其相应的标签；定义了 3 种类型的属性：General Attributes（元素一般属性），Linking Attributes（链接属性），Tabular Display Attributes（表格或图形显示属性），每种属性都有数十个。

每一个元素的定义由标签名（Tag Name）、元素名（Element Name）、说明（Description）、可能包含（May contain）、可能包含于（May occur within）、属性（Attributes）示例（Examples）7 个部分组成，如下图所示。

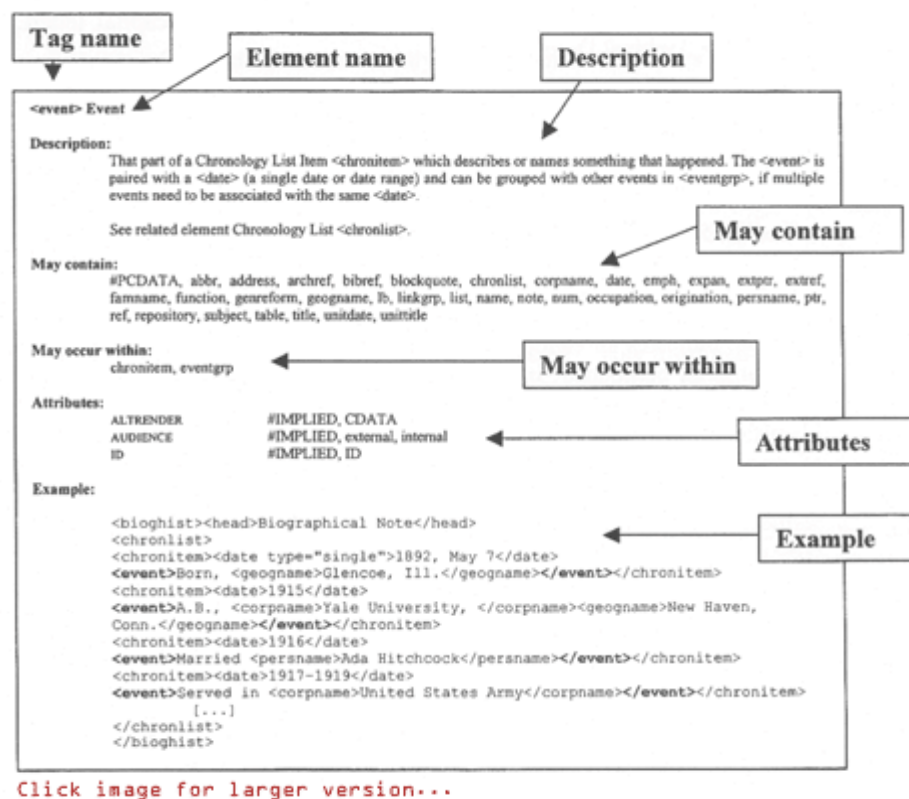


图1 元素定义示意图

- (1) **标签名(Tag Name):** 是元素名的简称，易于记忆，用于机读编码文档。标签名是页面最上方的第一个单词，标签名以小写形式出现在尖括号<>内如：<archdesc>，当出现在“May occur within”和“May contain”的内容中时，标签名的形式不是小写。
- (2) **元素名称(Element Name):** 是“tag name”的扩展形式，更能表达该元素的完整意义。元素的全称通常是一个单词或者短语，表示该元素的作用。在 tag library 中的元素定义文档中，“element name”跟在“tag name”之后，首字母大写，例如<archdesc> Archival Description。
- (3) **描述(Description):** 说明的第一段用来定义元素名所取自的档案术语表，如 ISAD(G), MARC21 等。其它段落用来描述元素的使用方法，相似元素的区分方法，有用的属性，以及一些实例。Stock phrase 意指元素具有争议性；Wrapper element 则表明该元素不能直接有文本描述，而位于下一级的裸元素首先被发布。如描述的唯一标识符<did>就是一个 Wrapper element，在其之下有一系列元素用来描述查档帮助的特定部分。Generic element 是指和其他很多文档类似的一些元素，并不一定仅出现在查档帮助中，如<address>和<date>元素。Formatting element 是用来规避在文本呈现时的一些特殊字符，如 block quotes, chronologies 和 emphasis 等。
- (4) **May contain:** 定义元素可能包含的内容。元素以标签名的字母顺序排列。元素可以是空（如仅含一个属性），或者包含文本（被称为 #PCDATA），或者

包含其它元素，或者既包含文本又包含其它元素。

- (5) **May occur within:** 定义该元素的所有可能的父元素，这些元素以标签名的字母顺序排列。父元素的界定可以提供更多的关于该元素使用的信息。
- (6) **Attributes:** 定义一个元素的所有属性。属性在 XML 编码中以小写字母表示，但是为了与元素区分开来，Online EAD Tag Library 使用大写字母来表示属性。
- (7) **Examples:** 大部分的元素描述包括一个被标记的实例，以表明属性和元素是怎样一起使用的。绝大部分的例子来自于实际的应用，另外一些则专为举例而创建。这些例子说明了元素组合的各种可能，例如<eadheader>标签之下有子元素的情况，再如<archdesc>之下有 level 元素的情况。在另外的情况下，这些例子用来简单地说明什么是可能的。有些示例进行了深度标引，使用了很多属性，而有些则按最低需求进行标引。这两种方式在 EAD 中都是有效的，而且这就把选择权留给存储实体机构，让他们根据自己的资源和指导方针来决定其标记水平。

4.7 EAD 的应用规则

EAD 也是基于 SGML/XML 语法的数据交换格式。在某些时候，通过关系型或面向对象数据库实现档案描述，EAD 此时主要作为数据交换标准；在其它情况下，档案描述可能直接通过基于 SGML/XML 的系统实现。EAD 应满足上述两种情形的需要。

EAD 用于说明档案描述的结构化内容，而不注重其呈现方式。不过，该标准提供了支持不同格式的输出机制，包括传统格式的查档帮助在内。

4.8 EAD 评价

在 EAD 的发展过程中，涌现了大量的实施 EAD 的研究和实验项目。在早期阶段，加利福尼亚大学 Berkeley 图书馆、DUKE 大学和国会图书馆开始用 EAD 著录检索工具，并测试 EAD 的理论和技術基础。加利福尼亚大学和圣地亚哥大学成功地用 EAD 输出了数据库形成的 EAD 检索工具。自 EAD 公布以来，英国伦敦公共档案馆、利物浦、牛津等大学也开始运用 EAD。

总体而言，EAD 在档案领域的应用比较显著，对于其他行业的渗透较少，这与档案的一些特质是相关的。不过，由于其对层级性较强的资源集合的描述及封装甚至包括显示、发布具有较好的支持，对其它类型复合数字对象建设具有一定的参考价值。

5. SCORM——内容可共享的数字对象

SCORM是美国政府在1997年底，由白宫科技办公室与国防部共同推动的高级分布式学习启动计划(Advanced Distributed Learning Initiative)的成果。该计划联合教材开发商与使用者、教学管理系统环球学习联盟(Instructional Management System Global Learning Consortium, IMS)、航空工业基于计算机训练委员会(Aviation Industry CBT Committee, MCC)、电气和电子工程师协会(IEEE)等标准化推动单位，整合美国各界过去在教材标准上的成果，共同研究、制定了一个针对教学课件复合数字对象的技术指南性质的标准，这就是共享型课程对象参考模型(Shareable Course Object Reference Model, SCORM)：SCORM中又主要包括两方面的内容：内容整合模型(Content Aggregation Model, CAM)和运行环境(Run-time Environment, RTE)，本节主要介绍SCORM的内容整合：

5.1 内容模型

SCORM 内容模型提出了用可重用学习资源来创建学习过程的 SCORM 构件，同时也定义了如何集合小粒度的可共享、可重用学习资源而组成大粒度的学习单元。

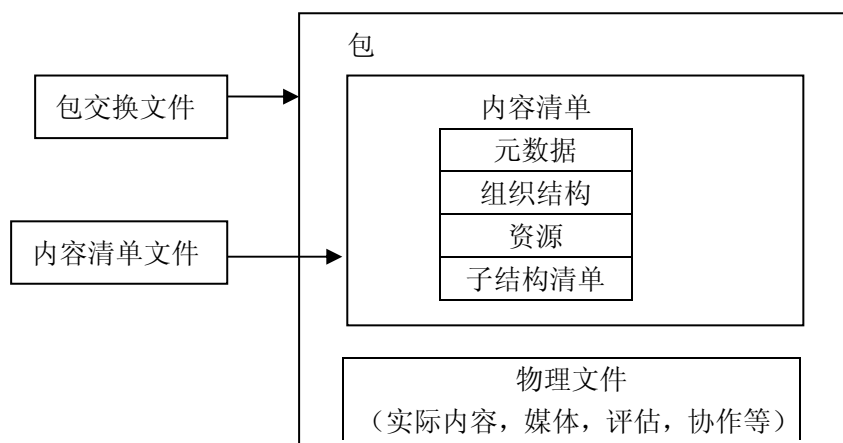
- Asset—这是可以由浏览器读取的文件。比如普通文本文件、HTML、GIF、小程序、基于应用程序的插件等。
- SCO—asset 的集合，实现同学习管理系统接口的预先规定的方法，SCO 是学习课程实现互操作的最低层次——3 任何学习管理系统都能启动和跟踪 SCO。
- Content Aggregation—这是一种类似目录的结构，它组织课程内容的顺序和导航。(注意，内容顺序同导航逻辑是有区别的)。

SCORM 内容模型中 Asset、SCO 和 Content Aggregation 都定义了元数据 XML 绑定，这样使实际物理内容独立于描述信息，可以被其他任何部署了课程包学习管理系统检索和发现，从而促进了共享和重用。

5.2 内容包装

内容包装参考了 IMS 中的内容包装规范(content packaging specification, CP)。内容包装定义了一种能够用来交换学习对象内容的标准数据结构，使之能方便的输入、输出、合并或拆分，其核心思想是将实际物理资源(physical resources)和逻辑组织结构(logical structure)分开。

SCORM 课件包可以是一个 ZIP 压缩文件或 CD 等，每个可供在学习管理系统中交换的学习资源或包，在包的根目录下都有一个内容清单文档(ims-manifest.xml)。内容清单文档中包含了学习对象的元数据信息；资源信息部分(resources 元素)直接引用了物理文件的存放地址；组织结构信息(organizations 元素)描述了这些资源间的组织关系。一个内容清单文件，既可以描述一门课件，又可以描述课件中的一个课时，它描述的粒度可根据实际及用户的需要变化。SCORM 的内容包装信息模型示例如下图所示。



- 元数据：对内容清单做整体描述；
- 组织结构：描述了在内容清单中内容的一个或多个组织结构；
- 资源：包含了对所有实际资源和内容清单中所需媒体元素的引用，还包括描述资源的元数据和对任何外部文件的引用；
- 子内容清单：零个或多个嵌套的内容清单
内容清单的作用范围是可伸缩的。内容清单可以描述独立的学习内容—课程的一部分（教学对象），一门完整的课程或一系列课程的集合。一个包总是只包含一个顶级内容清单，顶级内容清单下可以包含零个或多个子内容清单。顶级内容清单描述整个包，嵌套的子内容清单描述其作用范围内的课程或教学对象。
- 物理文件是指一些时间的媒体文件，文本文件，测评文件和其他以文件形式存在的数据，存放在内容清单中描述的各子目录下。和内容清单一样，它们也是包中不可缺少的部分。

```
<?xml version=" 1.0" >
<manifest identifier=" manifest1 " version=" 1.1" >
<metadata>...</metadata>
<organizations>
  <organizations>
    <item>
      <item identifier=" SC01" Identifierref=" R_S1" />
      <item identifier=" SC02" Identifierref=" R_S2" />
    </item>
  </organizations>
</organizations>
<resource>
  <resource identifier=" R_S1" adlcp:scormtype=" sco"
href=Course/scol.htm" >
    <file href=" Lesson/scol.htm" />
    <file href=" Lesson/scol.htm" />
  </resource>
  <resource identifier=" R_S2" adlcp:scormtype=" sco"
href=Course/scol.htm" >
```

```

<file href=" Lesson/sco2.htm" />
<file href=" Lesson/sco2.htm" />
</resource>
</resource>
</manifest>

```

5.3 元数据

SCORM中的元数据直接参考了IEEE LTSC的学习对象元数据（LOM）标准。SCORM元数据信息模型中定义了元数据的数据元素并把这些数据元素分成9个类别，分别是通用类，生存期类，元数据描述类，技术类，教育类，权利类，关系类，评注类和分类类。

虽然元数据在SCORM 内容包中是可选项，但给内容包添加必要的元数据内容有百益而无一害。无论内容包是聚合重用还是拆分重用，计算机都需要通过元数据这个标签来查询和检索需要的教学资源。因此，除了对内容包整体添加元数据外，还应该对内容包中的学习活动、SCOs及Assets添加元数据。

SCORM 直接复用IEEE LOM元数据集，但该元数据集中元素在应用于数字资源对象、可共享内容对象和课程三个不同层次时，根据情况可能必备(Mandatory)、选用(Optional)或保留(Reserve)。实际上，SCORM根据IEEE LOM 元素适用情况分别组合成：Course Metadata(课程元数据)，指描述整个课程的元数据；Content Metadata(内容元数据)，描述各SCO或由SCO组合成的内容段(Block)；RawMedia Metadata(原始媒体元数据)，描述最底层的媒体文件。

5.4 评价

内容可共享的对象参考模型SCORM (the Sharable Content ObjectReference Model)，它主要解决了信息对象的可存取性、互操作性、持久性、可重用性等问题。一个信息对象类似于一个插件，一次创建后，可以根据需要随时嵌入到不同的学习工具、平台或应用系统中，支持重复使用和永久使用。SCORM的主要优点是：可以避免功能相同的数字对象重复创建与保存；数字对象可以独立加工制作、保存管理，支持统一检索与调用；支持对象的重组、嵌入与重用。

5.5 附录1：IEEE LOM元数据元素

IEEE LOM 基本结构 LOM 将描述学习对象各方面特征的元素分为9个基本类别，每个类别包括若干元素，整个元素体系如下：

- a. General(通用类)，集合了与学习资源总体内容有关的元素，包括：Identifier(标识符)；Title(题名)；CatalogEntry(目录款目，指明著录或标引学习对象的目录及其相应款目，有子元素Catalog和Entry)；Language(语言)；Description(内容说明)；Keywords(关键词)；Coverage(内容覆盖范围)；Structure(结构)；Aggregation Level(集成层次)。
- b. LifeCycle(生命周期类)，集合了与学习对象产生与应用生命周期相关的特征，包括：Version(版次)；Status(版本状态，例如草稿、最终版、修订版等)；Contribute(贡献信息，包括子元素Role，贡献角色，可为责任者、出版者、终审者、编辑者、绘图者等；子元素Entity，与角色对应的具体实体；子元素Date，该实体以该角色做出贡献的日期)。
- c. MetaMetadata(宏元数据类)，对学习对象元数据进行描述的元素集合，包括：Identifier(元数据标识符)；CatalogEntry(元数据目录款目，有子元素Catalog和Entry)；Contribute(元数据贡献信息)；也包括子元素Role(角色)、Entity(实体)、Date(日期)、Metadata Scheme(元数据格式)、Language(语言)。

d. Technical(技术类)，包括学习对象的技术特征元素集合，例如：Format(格式)；Size(数字资源大小)；Location(资源位置)；Requirements(技术系统要求，包括子元素Type，技术系统类型。如操作系统、浏览器；子元素Name，技术系统名称，如操作系统Windows、MacOS等，浏览器如Netscape、IE等；子元素MinimumVersion和Maximum Version，注明所要求的技术系统最低和最高版本)；Installation Remarks(安装说明)；Other Platform Requirements(技术平台的其它要求，如声卡、运行时间等)；Duration(学习对象正常播放的持续时间)。

e Educational(教育类)，包括描述学习对象的教育学和教学学特征的元素集合，例如：Interactivity Level(交互度，对象使用中使用者与内容交互程度的主观测量)；Semantic Density(语义密度，内容含量密度的主观测量)；Intended end user role(最终用户角色，例如教师、学生、管理者)；Learning Context(学习环境，例如小学、高中、大学、研究生、职业培训等)；Typical Age(使用者年龄范围)；Difficulty(难度，使用者完成学习的难度)；Typical Learning Time(通常学习时间)；Description(使用说明)；Language(语言)。

f. Rights(权利类)，包括与学习对象使用有关的元素，如Cost(是否付费)、Copyright and Other Restrictions(是否有版权或其它限制)、Description(使用条件说明)。

g. Relation(关系类)，描述与该学习对象关联的其他资源的元素集合，包括：Kind(关系种类，例如IsPartOf、HasPartOf等)；Resource(关联资源)；子元素Identifier(关联资源标识符)；Description(关联资源说明)；CatalogEntry(关联资源目录款目)。

h. Annotation(注解类)，包括对学习对象教学的注解，例如Person(注解人)、Date(注解日期)、Description(注解内容说明)。

i. Classification(分类类)，包括与对象分类有关的元素，例如Purpose(分类目的，例如学科分类、教育目标分类、教育等级分类等)，Taxon_Path(分类法路径)，Source(分类法来源)，Taxon(分类类目)，Description(分类说明)，Keywords(关键词，按相关度排列)。

LOM 对每个元素定义了其名称、解释、多值性、域、类型、附注和示例，许多元素可自动生成或通过模板生成。

6. METS

元数据编码和传输标准(METS，Metadata Encoding and Transmission Standard)是从1990年以来的美国数字化项目中发展出来的。是在MOA2(Making Of America II)的工作基础之上，提供一个基于XML的为数字图书馆的数字对象进行封装的描述性、管理性和结构性元数据标准。它可用于在不同的存储库内的数字对象交换数据，也可用作对数字对象本地化的输入、检索和显示的封装机制。METS是一个开放的标准。由图书馆员和Interact团体共同建造的。它的主编者是Jerome McDonough，LC是此标准的维护机构，建立了METS的网站(<http://www.loc.gov/standards/mets>)，设计了一个开放的邮件程序来执行和扩展此模式。在METS XML模式发展后一年多，开始了世界范围内的实验。METS是数字化资料仓库的基础，仅在LC就有超过7百万的数字资源随同有关它的信息一同保存。采用多种结构的资料仓库是保存和检索数字化对象的工具。METS数据和其他系统使用的数字化对象可以相互转换。如果一个数字化资源用METS进行描述，它就可以在很多系统中方便地使用。

METS模式的结构高度灵活和相对简单。在METS中，元数据可以被编码到METS文档内，也可以通过链接引用外部文件。METS的标识符扩展系统可以把数字对象的各个部分联系在一起。因此METS可以作为1P(Archival Information Package)用于存储和长期保存，也可以作

为OAIS参考模型中的SIP, AIP和DIP, 提供标准化的传输语法。

它有7个概念化模块, 在这7个模块中, METS标准没有定义数据元素和可用的标识符。它允许使用者选择一个标准“扩展”模式, 进行标识和使用。除了头标区和结构图外, 其余都是可选择的。头标区和结构图对于数字化资源检索是必须的。描述、管理和行为区可以包含在内或文件外。

6.1 METS 对象的组成

METS本身并不提供相关元数据标准, 对相关元数据也不作限制, 可以是任何已知的元数据。但METS推荐使用注册的元数据标准。METS引用元数据的方法比较灵活 通过<mdref>和<mdwrap>两个元素提供指向外部文件元数据和内部封装元数据的机制。一个完整的METS文档由七个部分组成：

- 1) METS文件头(metsHdr);
- 2) 描述元数据部分(dmdSec);
- 3) 管理元数据部分(amdSec);
- 4) 文件部分(fileSec);
- 5) 结构图部分(structMap);
- 6) 结构链接部分(structLink);
- 7) 操作部分(behaviorSec)。

METS文档中除结构图部分是必备项外, 其它六部分均为可选项, 一般一个METS文档对应着一个数字对象 而这个数字对象可以包含有多个文件。所有相关的元数据, 无论是描述元数据 管理元数据还是结构元数据都要编码到一个文档中。各个类型的元数据要放到各个的部分中, 并通过标识符把它们联系到一起。所有的元数据和外部数据既可以通过METS文档引用也可以内嵌到文档中。其中管理元数据部分提供有关数字对象组成文件及数字对象来源文件的管理元数据。在METS文档中的管理元数据有四种:

- 1) 技术元数据(<techMD>)
- 2) 知识产权元数据(<rightMD>);
- 3) 原始资料来源元数据(<sourceMD>),
- 4) 数字对象出处元数据(<digiprovMD>)。

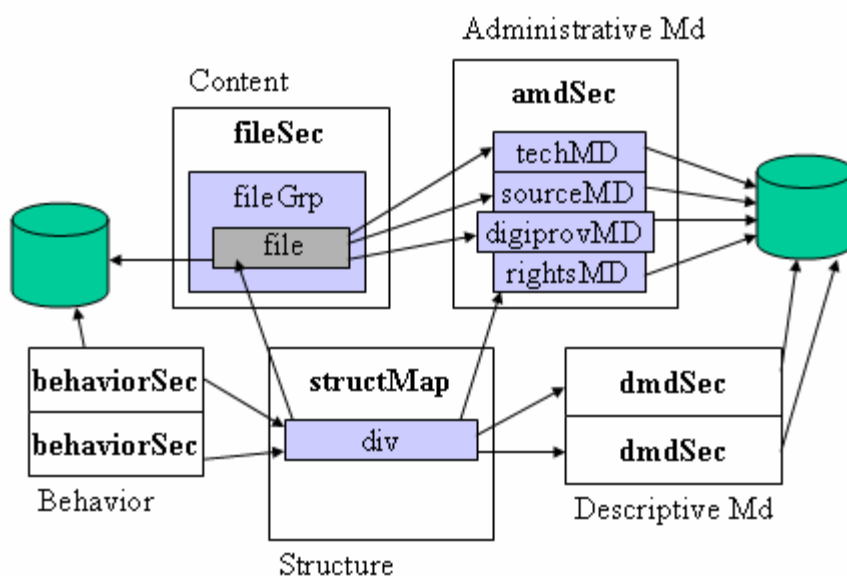
尽管METS在应用时有很大的灵活性, 但总有些基础性的工作是相同的。METS建议创建并注册METS应用profile。应用Profile是对一类METS文档予以详细说明, 以指导文档创建者和程序员创建 处理与具体应用相一致的METS文档。经注册的应用Profile将免费为公众利用。这样, 那些希望根据自己具体需求来创建METS文档的机构, 就可以直接利用这些Profile。目前注册的应用Profile有9个, 如牛津大学数字图书馆METS profile、UCB的图像对象profile等, 应用Profile也是用XML表示, 通常包含有十三部分:

- 1) 唯一URI(uniqueLRI);
- 2) 短标题(short title);
- 3) 摘要(abstract);
- 4) 创建日期和时间(date and time of creation).
- 5) 联系信息(contact information).
- 6) 相关的应用Profile(related Profiles);
- 7) 扩展schema(extension schemas).
- 8) 描述规则(rules of description);
- 9) 控制词表(controlled vocabularies).

- 10) 结构要求 (structural requirement)
- 11) 技术要求 (technical requirement):
- 12) 工具和应用 (tools and application):
- 13) 样式文件 (sample document)

6.2 METS 对象各组成部分之间的关系

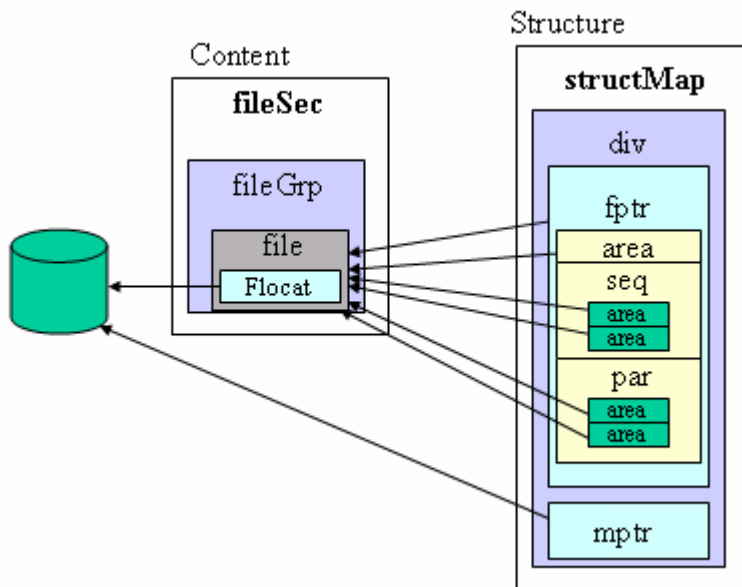
“结构图”是一个METS文件的拔心。它为数字图书馆的数字对象勾画出有关数字对象的层次结构，并且将该结构的各元素连接到与之相应的文件和元数据上。它由若干个DIV构成。它的structMap元素通过一层层的“div”树来表达数字对象的结构（物理结构或逻辑结构）。如图示：



1. 结构元数据与文件之间的链接

“文件组部分”列出组成数字对象的所有的电子版本的所有文件。文件组元素可“嵌套”，用以提供数字对象的版本细分文件。

StructMap通过mptr或fptr元素指向文件组fileSec元素的相应内容，实现结构元数据与全文（Content）之间的链接。



fp_{tr}是一个文件指针，通过它的“FILEID”属性指向该div节点所对应的文件，或者指向该文件的area（一个文件片断）、par（一组并行文件）或者seq（一组顺序文件）区域。mp_{tr}是METS对象指针，指向与“div”子项相关联的另一个独立的METS文档（如上图所示）。

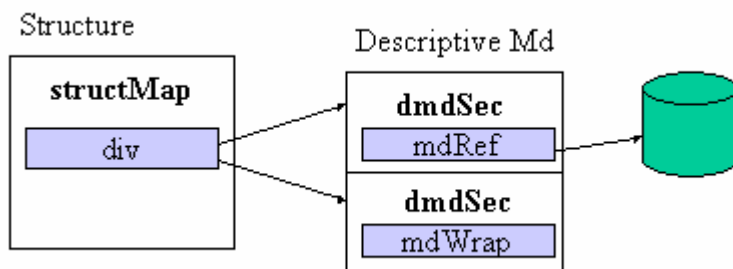
2. 结构元数据与描述性元数据的链接

“描述性元数据部分”记录了METS对象所有子对象的描述性元数据。该元数据可以是指同到METS文件外部的描述性元数据（例如，在OPAC上的一条MPd[~]C记录或在一个万维网服务器上一个EAD 检索工具，用mdRe{表示）。也可以包吉内部嵌入的描述性元数据（用mdWrap表示）。或两者都有。这样可以为每个METS对象的各个独立的子对象记录描述性元数据。

结构元数据与描述性元数据的链接是通过StructMap. div的DMDID属性与描述性元数据dmdSec元素的ID属性的关联，以实现结构中的每一个子项与其描述性元数据的链接。

通过DMDID属性，一个div元素可以指向一条或多条描述元数据。也就是说，一个子项可以对应多个版本的描述性元数据。

每一层的div元素，都可以有自己的描述性元数据。如图示：



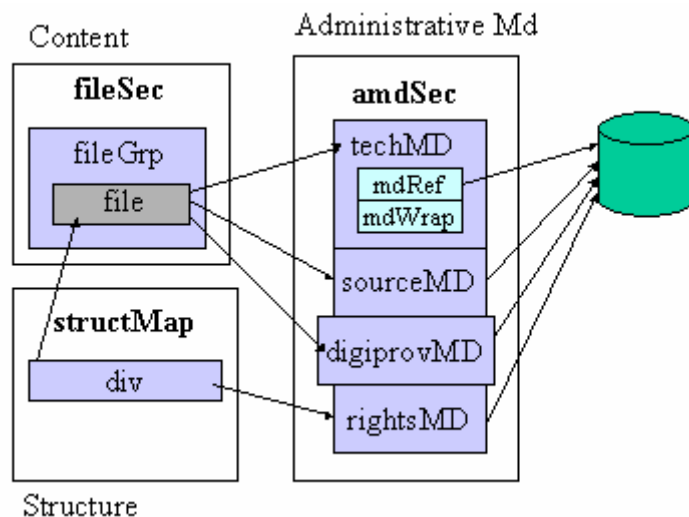
3. 结构元数据与管理元数据及相关文件的链接

“管理性元数据部分”提供关于文件被创造和存储的信息，关于知识产权权利，关于数字对象来源的元数据，W及关于组成数字图书馆对象的文件起源的信息（例如，原生 / 衍生文件关系，迁址 / 转变信息）。与描述性元数据相同，管理性元数据也可 存在于METS文件的外部，或封装在METS文件的内部。

结构元数据与管理元数据之间的链接是通过structMap的div子元素的ADMIN属性。该ADMIN属性值对应amdSec元素中techMD, rightsMD, sourceMD或digiprovMD的ID属性，之间可以是一对多的关系。

不论哪一层的div元素都可以具有其管理元数据。

文件与管理元数据的链接可以通过fileSec区的file元素的ADMIN属性，同样的对应于amdSec元素中techMD, rightsMD, sourceMD或digiprovMD的ID属性，之间可以是一对多的关系。



例如：

```
<METS:mets>
```

```
<METS:structMap TYPE="logical">
```

```
  <METS:div ORDER="1" TYPE="diary" LABEL="Breen Diary"
    DMDID="DM1 DM2" ADMID="RM1">
```

```
    <METS:div ORDER="1" TYPE="entry" LABEL="Nov 20">
```

```
      <METS:fptr FILEID="FID3" />
```

```
      <METS:fptr FILEID="FID35" />
```

```
    <METS:fptr>
```

```
      <METS:area> FILEID="FID1" BETYPE="IDREF"
```

```
        BEGIN="ENTRY1" END="ENTRYEND1" />
```

```
    </METS:fptr>
```

```
  </METS:div>
```

```
  <METS:div ORDER="2" TYPE="entry" LABEL="Nov 21" />
```

```
  ...
```

```
</METS:div>
```

```
</METS:structMap>
```

```
<METS:amdSec>
```

```
  <METS:rightsMD ID="RM1">
```

```
    <METS:mdWrap MDTYPE="OTHER" OTHERMDTYPE="GAMRIGHTS">
```

```
      <METS:xmlData>
```

```
        <gamrights:gamrights>
```

```
          <gamrights:copyRest>Copyright has been assigned to the Bancroft Library.All
```

```
requests...
```

```
        </gamrights:copyRest>
```

```
</gamrights:gamrights>
</METS:xmlData>
</METS:mdWrap>
</METS:rightsMD>
</METS:dmdSec>
<METS:mets>
```

6.3 评价

METs的优点在于它能以一种灵活但有组织、有条理的方式处理大量不同种类的信息资源。它为整合与特定数字对象相关的元数据提供了一种方法，而以XML编码为METS文档的创建、显示、转换、查询带来了极大的便利。

METS允许它的不同模块使用已建立的任何模式。如统一的标准还未建立时，这种灵活性可能会妨碍互操作性。所以METS的使用和发展更有赖于国际化的合作。许多研究机构已经开始了数字化试验并已积累了大量的数据，它们要求更好的组织工具。近期的METS应用主要是按资料的形式和尺寸区分，提供了高质量的信息。LC将METS用于大量的动画和视觉资料及其它混合型媒体。威尔士国家图书馆用METS初始化文本资料。哈佛大学将其用于视觉资料，密歇根州将其用于动画影片。oCLC和RLC都将它用于他们的数字化项目等等。

7. Fedora

7.1 Fedora 简介

William Y. Arms 于 1997 年提出了数字对象框架 DOA (Digital Object Architecture)，提出了数字对象的概念模型。通过该框架，资源类型和格式不同，元数据格式不同，网络分布位置不同，提供服务不同的数字资源都能被统一描述成数字对象，存储管理于仓储之中。这样就很好地解决了统一存储管理复合数字对象的问题。康奈尔大学对此框架进行了原型实现。2000 年弗吉尼亚大学为了自身数字图书馆建设的需要，选择 Fedora (Flexible Extensible Digital Object and Repository Architecture) 框架，并对测试系统进行了开发。2001 年，在梅隆基金会的支持下，弗吉尼亚大学和康奈尔大学合作对 Fedora 进行二次开发，利用 Web 技术开发出一个开放源代码的 Fedora 版本。目前的 Fedora 2.2 为信息的组织管理和表现提供了更丰富的功能。该系统实现了复合数字对象管理框架，可以统一存储管理属性各异的数字资源。

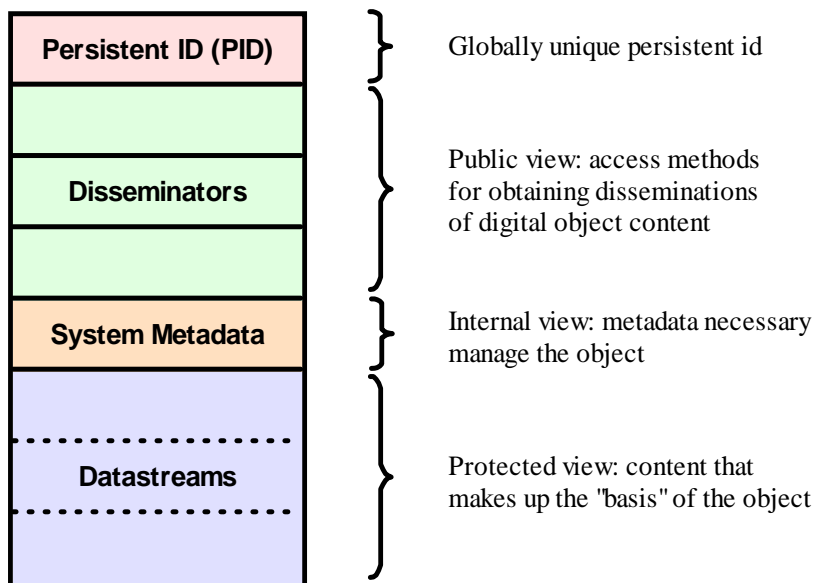
7.2 Fedora 数字对象模型

Fedora 系统由两个实体组成：(1) Fedora 数字对象框架，它实现了 William Y. Arms 最初提出的框架；(2) Fedora 仓储。数字对象是 Fedora 体系结构的核心，提供了一个集成内容（包括数据和元数据）和行为（对内容的操作）的框架。其中内容和行为本身都是可以分布在网络的不同节点上的，通过一个 URI 进行引用。Fedora 仓储提供了对数字对象的存储

管理和访问接口。在本文中，只介绍 Fedora 的数字对象框架。

DOA 框架从面向对象的角度来讲，数字对象（Digital Object）是一个唯一标识的网络实体，它用来封装，描述属性不同的数字资源，并且提供访问数字资源的机制。通常我们会将一条数字资源描述成一个数字对象。一条数字资源可以是一个文本，一张图片，也可以是一本电子书。数字对象由两个部分组成：数据和数据发布者（Disseminator）。数据包含两部分的内容：核心元数据（Key Metadata）和数据流（Datastreams）。核心元数据又包含两部分元素：数字对象唯一标识符（Digital Object Identifier）和系统元数据（System Metadata）。

在 Fedora 系统的实现中，一个数字对象由以下几部分组成：



1. 数字对象唯一标识符用来唯一地标识一个数字对象，以便在命名空间内唯一地引用该数字对象。

2. 数据发布者（Disseminator）是数字对象内部的一种结构，对应着一种发布数字对象内容的方式。一个数字对象有一个或者多个数据发布者。一个数据发布者含有两个元素：数据内容类型（Content Type）和操作的数据元素。数据内容类型对应着一系列的数据内容类型操作接口（Content Type Signature）和数据内容类型操作的实现（Content Type Implementation）。数据内容类型，也就是操作接口和操作的实现本身，是独立于数字对象而存在的。

3. 系统元数据（System Metadata）描述整个数字对象，用来管理该数字对象和建立数字对象的索引。事实上 Fedora 将每个数字对象的系统元数据存储于数据库中，实现了对于数字对象的简单的检索功能。

4. 数据流（Datastreams）是数字对象所包含的数据，它可以是元数据，也可以是数据本身。如果是元数据，可能是各种格式的，比如 Dublin Core, MARC。如果是数据本身，可能是文本，图像，音频，视频数据。数据元素将数据和元数据统一对待，这样就可以统一处理资源类型格式不同，元数据格式不同的数字资源。

这四部分概括为两个层次：数据和数据发布者，数据发布者就是接口层，为数字对象中的数据提供操作服务接口。在整个数字对象的框架中，把数据和数据的服务封装在一起，密封了内部实现上的细节，对外表现出一个一致的形式，用户不需要了解数字对象的任何内部细节，就可以有效地使用这些数字对象。这种结构为数字对象地扩展和重用提供了必要地基础。同时由于整体框架实现中数据和发布者可以相互独立，不同地物理格式可以共用相同地存取方式，物理格式地变化不会对存取方法产生影响。因此，这种分离使得 Fedora 数字对

象框架在理论上可以容纳多种数字资源物理格式。

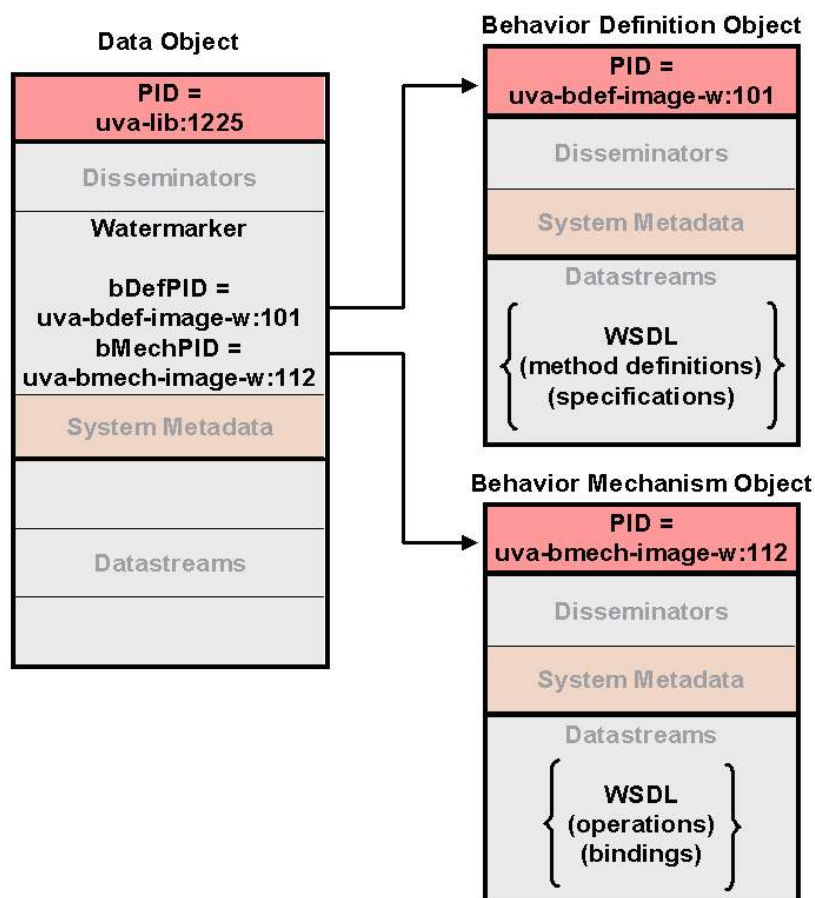
在 Fedora 中，有三种类型的数字对象：（1）数据数字对象，（2）行为定义对象，（3）行为实现对象。Fedora 将数据发布器所包含的行为定义和实现也都做成了数字对象。行为定义对象是一种特殊的数字对象，它将一组抽象的行为定义集成到一起。这些行为定义是一组转换或者展示数字对象数据的方法。行为实现对象也是一种特殊的数字对象，是一组对行为定义对象中方法的实现。行为实现对象包括行为实现和数据元素的绑定关系。抽象的行为定义和实现信息用 WSDL 定义进行表述。

在下图中，一个PID为uva-lib:1225 的数字对象有一个Watermarker数据发布器，可以动态地为一幅图片加水印。数据发布器有两个明显的属性：行为定义标识符和行为实现标识符，这两个标识符实际上是行为定义对象和行为实现对象的PID。行为定义对象和行为实现对象的数据元素就是用WSDL描述的行为定义规范和行为实现绑定。

需要强调的一点是，行为定义数字对象和行为实现数字对象是独立于数据数字对象而存在的。它们之间的绑定是动态执行的。因此该系统可以支持动态的为数字对象增加或者减少行为定义和行为实现。

数字对象最终以 XML 文件的形式存在。编码 Fedora 数字对象的 XML 文件的 Schema 是对 METS (Metadata Transmission and Encoding Standard) 规范进行扩展得到的。METS 是一个用来对元数据进行编码和传输的规范，已经获得了广泛的认可。

图 2 行为定义和行为实现对象



8. MVD——多价文档

8.1 MVD 概述

Multivalent Documents（多价文档，MVD）系统是一种开放式、可扩展、分布式的文件模型。由加州大学伯克利分校开发，主要研制人员为Tom Phelps博士与Robert Wilensky教授。

多价文档，与传统的单一文档结构——由一种单一的标记语言或者文档组成相区别，是由一个具有多层的文档组成，每一层都是同构内容组成，功能和范围相对统一。

8.2 MVD 结构：层和行为

多价文档是由层和行为组成，层用来提供数据，行为用来提供特定的功能，每一个MVD具有不同的层和行为。

MVD文档的每一层都有三个区：第一个区包含文件信息。多数MVD文档都有一个“基本层”，文档的主要内容都从中提取。比如，一个MVD文档可能指定一个读取HTML的行为，伴随着一个HTML层的地址；或一个PDF识别行为，伴随一个PDF文档层。另外两个区分别在层与层之间、用户与计算机之间提供交互界面。大部分的层都有特定格式，如于MVD0版本的图形浏览器的数据格式主要包括：图像、超链接和表格等。

行为是访问和操作数据内容的手段。通过行为实现对各个内容层的操作。行为包括创建、存储、还原、格式化、描绘、选择、检索、打印等等跨媒体类型进行的一般数据操作行为；以及一种行为专门负责识别数据的媒体类型和进行格式转换，称为“媒体适配器”。不同的媒体类型被媒体适配器的整合到多价文档的行为中。

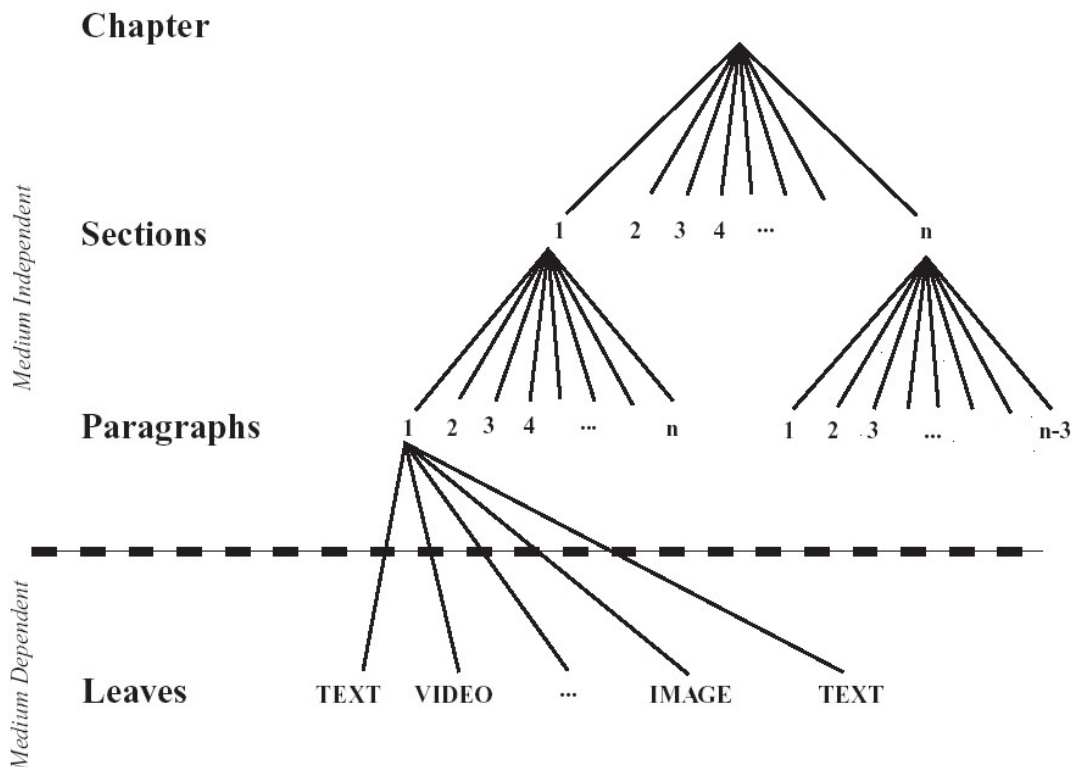
行为提供一个或者多个用户层的功能界面。类似于面向对象程序语言中的类，行为在每一层中进行封装，作为操作私有数据的方法。行为并不直接操作数据，而是通过程序层的界面进行交流。大部分的行为在各种范围的文档中都是可用的。

每一种行为都各有侧重。比如，一些行为是以数据为中心，通过程序层界面提供信息知识库来实现对数据的访问；一些行为是以程序或者功能为中心，如，通常检索行为自身一般不存储任何数据（除了在检索中需要忽略的截词以外），检索功能要求其他的行为为其提供文本以供检索。在用户层，可以通过简单的输入检索条目或者仅仅依赖其他的行为来调用检索；还有一些行为主要是提供用户界面来实现在其他地方对内容的获取和其他的功能。通过多价文档中各种类型的行为，最终实现普通的用户能够达到专家使用的复杂功能和他们之间的互动。

每一个多价文档的实现都提供一个架构，通过该架构，层可以生成不同的行为，追加新的层；同时，行为可以对内容进行操作，或者和其他的层或者行为之间一起实现某种功能实现行为之间的互操作，提供所需要的功能，生成新的文档，在多价文档中加入新的行为或者

重新排列组合行为。¹

8.3MVD 文档树



在多价文档模型中，层用于提供数据，行为用于提供功能，系统是建立在层和行为的异构整合的基础上。文档树是一个树型的数据结构，用于反映文档的逻辑结构，扩展可能的数据类型。在文档树中，行为通过层来操纵各种数据类型。文档树的内部节点反映文档的逻辑层级关系。例如，一本书由不同的章构成，每一章由不同的节构成，每一节由不同的子节（包括图表和段落或者其他类型）构成，图表由不同的单元格构成，段落由文本、图或者其他的类型构成。

文档树的叶节点包括各种元素：单词、图片、架构和图等。²

8.4MVD 批注功能

在层与行为架构的基础上，MVD衍生了独特的批注（notes）功能。批注功能允许使用者在现存内容上附上新增的内容。批注本身即是行为的一种。MVD批注是分布式的，它们可以在被批注的地方分开存贮，但形式上却像是一个文档的内在整体。经过批注的文件可与其他使用者共享，而无需特殊服务器支持。

MVD批注行为可分为几种类型：

¹ <http://elib.cs.berkeley.edu/ib/about.html>

² Thomas A. Phelps. Multivalent Documents: Anytime, Anywhere, Any Type, Every Way User-Improvable Digital Documents and Systems

(1) 跨度 (span)：一般是指文本文档某点与某点之间的内容，比如可将文档的一部分下划线，建立超级链接。

(2) 几何透镜 (lense)：在文档上划一个几何区域的窗口，在窗口中如同透镜一样可以局部改变内容显示结果。如可以在窗口中显示OCR识别结果和文本翻译结果。

(3) 结构 (structure)：在文件树的结构层面上的批注。可以理解为对文档进行整体批注。

(4) 地理批注：可对指定的区域进行GIS查看。

MVD的分布式批注具有一些引人注目的特性：

- (1) 高度的表达性：兼容不同的意义和功能；
- (2) 平台和格式独立性：允许用户使用不同的本地系统；
- (3) 可扩展性：兼容不同用户风格，可无缝集成；
- (4) 分布并开放：凡所见皆可批注，共享结果。

8.5 MVD 类型图

通常，多价文档的层被地域性的分布在各种不同的知识库中。用户可以通过客户端，与不同的服务器之间实现交流，查询首先提交到“句柄服务器”，返回拥有相关信息的服务器列表，客户端查找这个列表，找到需要与文档之间进行最小互动的一些主要的行为，如果需要，同时下载其他的一些行为。在所有的过程中，服务器都返回一系列的类型描述，是对服务器的行为的特定的简单描述，这些信息构成类型图 (the type graph)。

只要客户端有访问权限，服务器都返回一个类型描述，每一个层和行为都被划分为相应的类型，同时构建类型图。当一个层需要一个特定的信息层或者行为，该层将会通过类型图来获取一个对象来满足该层的需求；如果它所需要的行为不能够通过本地获取，层可以通过网络来取得。类型图是管理行为之间的互动的关键。因为类型图是本地管理，所以它可以重新排列和组合，比如可以覆盖或者调整一个行为。

8.6 MVD 评价

总体而言，多价文档并不是一个特定的文档类型，而是通过写入特定的行为来处理某些特定的文档，如，通过行为来处理扫描的图片，ASCII或者HTML。同时，多价文档支持分布式标注，支持协作写作，可以说是MVD在诸多符合数字对象模型中最显著的特色。通过多价文档模型，可以构建具有丰富属性的在线数字文档。

9. 参考文献：

1. 李春旺, 张晓林. 复合数字对象研究. 情报学报, 2004 (4), 444—451
2. 何朝晖, 王波, 朱强. 国外复合数字对象管理研究摘要, 2005 (5), 6-14
3. 马蕾, 元数据及其封装标准 METS 研究. 情报技术, 2002 (2), 56-57)
4. 张铮, 李蓓, 元数据家族中的新成员-MODS 和 METS 2005,18(7), 3
5. 李蓓, 数字化图书馆资源仓库的基础——METS, 2004,22(11),1375-1387
6. 中国高等教育文献保障系统管理中心, 中国高等教育数字图书馆, 2004.
7. Mets, <http://www.loc.gov/standards/mets/>
8. 付琴, 程文清, 杨宗凯, SCORM-可共享对象参考模型的研究. 中国远程教育, 2002(192),60-62
9. 张群,程玉,黄庆炬, SCORM 标准的改进设想和 OCO 的提出. 湖北工业大学学报,2006, 21(4), 134-137
10. 余胜泉, 俞晖, 可共享内容对象参考模型研究. 现代远程教育研究, 2003(64), 47-50)
11. SCORM Specification <http://www.adlnet.org>
12. ADL. Sharable content object reference model, <http://www.adlnet.org/scorm/index.cfm>
13. 王昉, 张晓林, 面向教育资源的元数据, 情报杂志, 2002(7), 37-39
14. Sandra Payette, Carl Lagoze, “Flexible and Extensible Digital Object and Repository Architecture (FEDORA)”, Second European Conference on Research and Advanced Technology for Digital Libraries, September 21-23, 1998
15. Mellon Fedora Technical Specification Version 1.1, <http://www.fedora.info/>, 2002.12
16. METS, <http://www.loc.gov/standards/mets/>
17. Fedora, <http://www.fedora.info/>
18. OEB, <http://www.idpf.org/>
19. EAD<http://www.loc.gov/ead/index.html> [Access 2007-6-23]
20. 候卫真.对档案描述编码格式（EAD）的探讨.北京档案,2003(12):19-21
21. 李建立. 美国档案信息全文著录系统—EAD 的形成与发展. 中国档案. 2001. (10)
22. 蓝天. 档案置标著录(EAD)管窥. 航空档案. 2003 (2) : 21-24
23. Thomas A.Phelps. Multivalent Documents: Anytime, Anywhere, Any Type, Every Way User-Improvable Digital Documents and Systems[D]. Berkeley: Computer Science of University of California, 1998. <http://www.eecs.berkeley.edu/Pubs/TechRpts/1998/CSD-98-1026.pdf>[Access2007-7-15]
24. About MVD Image Browser 0.0. <http://elib.cs.berkeley.edu/ib/about.html> [Access2007-7-15]
25. About MVD. <http://www.eecs.berkeley.edu/Pubs/TechRpts/1998/5332.html>[Access2007-7-15]
26. 何朝晖,王波,朱强.复合数字对象管理技术研究.科技部科技基础条件平台工作重大项目《数字图书馆标准与规范建设》项目资助.