

中华人民共和国国家标准

GB/T ××××—××××

信息技术 学习、教育和培训
学习设计信息模型 V1.0

Information technology - Learning, education and training –
Learning Design Information Model

Version 1.0

(标准送审稿)

(CD2.3)

200X-xx-xx 发布

200X-xx-xx 实施

中华人民共和国国家质量监督检验检疫总局 发布
中国国家标准化管理委员会

前 言	III
引 言	IV
1. 范围	1
2. 规范性引用文件	2
3. 术语和定义	2
4. 缩略语	5
5. 概念模型	6
5.1 学习设计规范的目标	6
5.2 概念模型	6
5.2.1 学习设计中的语义聚合	7
5.2.2 学习设计的概念架构	8
5.2.3 学习单元= 内容包 + 学习设计	10
5.3 学习设计的概念词汇表	11
6. 信息模型	17
6.1 A 层信息模型	18
6.1.1 概念模型	18
6.1.2 学习设计信息表	19
6.1.3 子内容项模型信息表	21
6.1.4 组件信息表	21
6.1.5 角色信息表	22
6.1.6 活动信息表	24
6.1.7 学习活动信息表	26
6.1.8 支持活动信息表	28
6.1.9 活动结构信息表	30
6.1.10 环境信息表	32
6.1.11 服务信息表	35
6.1.12 方法信息表	39
6.1.13 剧信息表	41
6.1.14 幕信息表	43
6.1.15 学习单元表单文件的标准名称	46
6.1.16 学习设计元素的命名空间	46
6.2 B 层信息模型	46
6.2.1 概念模型	47
6.2.1.1 全局属性	48
6.2.2 属性信息表	48
6.2.3 当属性值设定时信息表	53
6.2.4 改变属性值信息表	54
6.2.5 监控信息表	56
6.2.6 电子邮件数据扩展	57
6.2.7 时间限制扩展	57
6.2.8 条件信息表	57
6.2.9 THEN 模块信息表	59
6.2.10 IF 模块信息表	59
6.2.11 表达式模块信息表	61

6.2.12 计算模块信息表.....	63
6.2.13 操作数模块信息表.....	64
6.2.14 当条件为真时信息表.....	64
6.2.15 “显示 & 隐藏”信息表.....	65
6.2.16 全局元素信息表.....	68
6.2.17 全局属性“类”属性.....	71
6.2.18 数据类型.....	71
6.2.19 限制类型.....	72
6.3 C层信息模型.....	73
6.3.1 概念模型.....	73
6.3.2 “通告”信息表.....	74
6.3.3 完成动作模型扩展.....	75
6.3.4 then 模型扩展.....	75
6.3.5 全局元素模型拓展.....	76
7. 行为模型.....	76
7.1 行为模型概述.....	76
7.1.1 实例化.....	76
7.1.1.1 实例化角色.....	77
7.1.1.2 实例化服务.....	77
7.1.2 执行.....	77
7.1.2.1 A层学习设计.....	77
7.1.2.2 B层学习设计.....	78
7.1.2.3 C层学习设计.....	80
7.1.3 控制的层级.....	80
7.2 学习单元实例化.....	84
7.3 实例化建立角色和服务.....	84
7.4 启动过程.....	85
7.5 完成规则.....	86
7.6 完成动作.....	87
7.7 学习成果记录和其 LIP 规范映射.....	87
8. 可扩展性.....	87
附录 A 用户代理标准.....	88
A.1 - A层标准.....	88
A.2 - B层标准.....	90
A.3 - C层标准.....	90

前 言

本标准主要基于IMS全球学习联合公司2003年1月发布的IMS学习设计信息模型V1.0（IMS Learning Design Information Model Version 1.0）制定，同时根据我国现有的教育技术标准体系结构及应用现状加以适当调整。

本标准由教育部提出。

本标准由全国信息技术标准化技术委员会归口。

本标准起草单位：华东师范大学。

本标准主要起草人：吴永和、顾小清、张屹、胡海明、张超、刘名卓、李宝敏。

引 言

学习设计规范应用在e-Learning领域的实际过程中，是设计一个课程框架，以支持课程的多样性及课程创新，并同时促进e-Learning学习材料的可交换与互操作；强调以学习活动为中心，利用教育建模语言描述各种不同的教学法，比如协作学习、混合式学习、基于问题的学习等。

信息技术 学习、教育和培训

学习设计信息模型

1. 范围

本标准给出了学习设计信息模型的指南。该信息模型将荷兰开放大学提交给学习设计工作组（Learning Design working group, LDWG）的教育建模语言（Educational Modelling Language, EML）[1]与现有的规范，特别是内容包装规范[2]进行了整合、扩展，同时也综合考虑了元数据规范[3]及简单编列规范[4]。

学习设计规范定义了实现和一致性的三个层次。本标准分别对此作了介绍。每一层应用对应一个独立的 XML 文档。

第一层学习设计（学习设计 A 层）包含了描述多样教学方法的所有核心词汇。第二层和第三层是在此基础上分别增加了三组概念及其性能，以便对更加复杂的行为加以描述。

第二层学习设计（学习设计 B 层）是在 A 层基础上增加了属性和条件，以便在学习者学习档案的基础上实现更加个别化、精细化的学习过程和学习交互。该层学习设计可以用来引导学习活动、记录学习成果。将属性与条件分离也可使之用于学习设计规范的其他场合，特别是可以用来对简单编列规范加以补充。

第三层学习设计（学习设计 C 层）在 B 层的基础上增加了通告，这对于整个规范来说虽然是很细微的补充，但却具有非常重要的意义，特别是对于规范的应用具有潜在的指导作用。

因此，此规范所采用的方法，并不是定义一个单一而巨大的规则库及其选项，而是定义一个完整的、尽可能简洁的核心（模型），在此基础上定义扩展的属性及行为，以便反映复杂的学习活动。

我们希望遵照规范的应用既严格又有一定的灵活性：A 层是很容易达到的，而灵活性就在于，是否或者何时应用此规范的更高层次。

任何一层的应用系统，都希望是遵照规范的。相对于学习设计，遵照此规范实施的实例文档，不需要实现规范中的每一个要素，因此在内容与支持系统之间，对规范的遵照是有差别的。可选要素适用于文档实例；而不管该实例的选项包括哪些，系统需要在某一层上严格遵照规范的每一个规定，这样才能在该层次上运行所有实例。遵照这一规范的学习设计实例需要利用附加的 XML 解析器进行解析才能生效，但由于并非所有系统对三个层次都提供支持，所以需要指出希望运行系统支持哪一层的应用，这样系统才能确定是否能够运行特定层次的学习设计实例。

本标准是学习设计规范。因此本文档也是以下文档的基础：

- 学习设计 XML 绑定（层次 A、B、C）；
- 学习设计实践范例与实施指南

三份文档合在一起，构成了学习设计规范。

信息模型描述的是学习设计的模型，由以下三个主要部分组成：

概念模型：包括描述概念的词汇表、概念之间的功能关系、与内容包装之间的关联。概念模型是从全局的角度（层次 C）进行描述的。

信息模型：分别从三个层次描述学习设计的要素。同时也对不同层次的特定概念模型进行了描述。

行为模型：描述了传递系统必须实现的运行时行为。

2. 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

GB/T 21365-2008 信息技术 学习、教育和培训 学习对象元数据规范。

CELT9.1 内容包装信息模型规范（最新版本 CD1.6）。

CELT8-8.1 简单课程编列规范：参考模型（CD1.0）。

CELT10-10.1 测试互操作：信息模型标准(CD2.0)。

CELT11-11.1 学习者模型规范：信息模型（CD2.0）。

3. 术语和定义

3.1

通知会议 Announcement Conference

通知是发给用户的消息，告之以新事件或相关信息，在环境、服务、会议对象中声明，其会议类型是**通知**（announcement）。

3.2

异步会议 Asynchronous Conference

能够存储收到消息的群发系统，并且通常按照主题排序。最原始的异步会议系统是INTERNET新闻组。

3.3

属性 Attribute

DTD中所声明元素的参数，通常定义其类型与取值范围（包括默认值）。

3.4

文档 Document

数据流，其结构信息含摄于相关DTD所定义的诸多元素中。

3.5

文档类型定义 DTD

XML声明集合，定义所需的合法的结构、元素、属性。

3.6

元/要素 Element

DTD中声明的文档结构单元，元素的内容模型定义于DTD中，附加的语义定义于元素的详细描述中。

3.7

支持 Facilities

包含元素、属性及其相关语义，实现支持功能在于其能够提供必要的条件。

3.8

实现 Implementation

能够提供诸多便利条件与服务以支持其详细要求的系统。

3.9

解析 Parsing

能够扫描文档，文档之信息被过虑，检查其与元素定义的一致性的行为。

3.10

属性操作 Property Operation

对属性或属性集上的以下运算：设置属性、查看属性、设置属性集、查看属性集、修改属性值。

3.11

渲染 Rendering

展现文档信息，展现之形式最适合于环境的要求（如音频、视频、打印文本）。

3.12

学习单元执行 Run of a Unit of Learning

一些可能事例之集合，这一集合体称为学习单元的一次执行。在执行过程中，每个人都绑定于学习单元中所定义的角色及其相应的启动日程，同样的学习单元、同样的标识可以无数次运行，相同的标识一般需要相同的结构与内容。学习单元标识之差别在于学习设计即内容与结构的版本号。

3.13

执行时 Runtime

用以阐释用户使用学习设计时的具体要求。

3.14

同步会议 Synchronous Conference

能够帮助群体间实时地相互通讯与工作的群集通讯系统，一般需要借助于诸多媒体，最原始的方法是用一种媒体，最复杂的情况是同时用异步与同步会议系统，此时一般也划归同步会议系统。

3.15

统一资源标识符 URI

由IETF与W3C共同制定的标准。其DTD描述时使用了W3C标注的涵义，这种标注并不严格区分URL与URN，每个URI可以是URL或URN，这取决于实现时的特定的情形，URI可

以是相对或绝对地址，前者是全局的，后者是局部的。对于局部URI标识的资源，其意味着资源就在学习单元包中，并且资源就是若干文件。

3.16

用户代理 User Agent

检索、处理学习设计文档的一次实现，其等同于系统的一次执行，至于处理在客户端或服务器端无关紧要。

3.17

有效性 Validation

文档通过相关DTD验证的过程，以确保结构、所使用的元素与属性和DTD中的定义一致。

3.18

WEB 内容 Web Content

资源的一种数据类型，其可以用WEB浏览器访问，如html、xml、flash、applets、text processor或样式单文件等，不要求其定义为良构的WM文档。

3.19

良构 Well-Formed

结构化并且符合XML 1.0 之Section 2.1所推荐的定义规则（参见：<http://www.w3.org/TR/xhtml1/#sec-well-formed>）的文档。该定义主要作用是声明其中的元素，这些元素由开始和结束标签标识，并恰当地相互嵌套。

4. 缩略语

EML: 教育建模语言, Educational Modelling Language

CP: 内容包装标准, Content Packaging Specification

MD: LOM 元数据标准, LOM Meta-Data Specification

QTI: 问题与测试互操作标准, Question and Test Interoperability Specification

LOM: 学习对象元数据, Learning Object Metadata (IEEE 1484.12.1 – 2002)

PCDATA: 字符数据, Character Data

UML: 统一建模语言, Unified Modeling Language

URI: 统一资源标识符, Universal Resource Identifier

W3C: World Wide Web Consortium

XML: 可扩展标记语言, Extensible Mark-up Language

5. 概念模型

5.1 学习设计规范的目标

学习设计规范的目标是为相关要素提供一个封装框架,以便以一种相对正式的方式,对教与学的过程进行描述。具体来说,学习设计规范满足以下要求:

——完整性 (completeness), 即能够完全描述一个学习单元的教学过程, 包括所需的数字化与非数字化学习对象与服务, 具体内容为: 整合学习者与教员的诸多活动、整合学习所需的诸多资源与服务、支持广泛的学习方式、支持混合学习模式、支持[在线学习 \(pure ?\)](#)。

——教学灵活性 (Pedagogical Flexibility), 即能够表述出学习单元内容的不同数据元素所内含的教学涵义与功能, 其灵活性在于描述了各种教学路径, 不预设某种特定的教学路径。

——个性化 (Personalization), 即能够描述学习设计的个性化要求, 学习单元的内容与活动能够适合用户的偏好、学档、预备知识、教育需求、学习情境, 此外必须给出实现灵活性的控制方式, 无论是针对学生、教员、计算机还是设计者。

——规范化 (Formalization), 即能够正式方式描述学习设计中的学习单元, 以保证自动处理。

——再生性 (Reproducibility), 即能够抽象的描述学习设计, 以保证不同用户不同环境的重复执行。

——互操作性 (Interoperability), 即能够支持学习设计的互操作。

——兼容性 (Compatibility), 即尽可能使用现有的标准与规格, 主要是内容包装协议、测试互操作性、LOM元数据标准、简单编列标准。

——重用性 (Reusability), 即能够识别、抽取、解析、交换学习制品, 并且能够在其他上下文中重用。

5.2 概念模型

概念模型可以表述为UML类模型与所用词汇的定义, 应该是全局模型, 不细分为ABC诸级模型, 其中某些元素没有出现于信息模型, 但却是达成更好概念理解所必需。其包括三种基

本模型，即聚合模型、结构模型、表现模型（整合内容包中的信息设计以形成所谓的学习单元）。在词汇定义前，必须提供这些模型。

5.2.1 学习设计中的语义聚合

图1代表学习设计中概念模型的语义聚合(参见 11)，图示标注符合UML1.4版的要求(参见13)，只代表聚合关系(包括组件)与抽象类或类型的特化。



图1 C层学习设计规范的语义聚合层次

这一模型表明，学习设计能够提供资源的语义视图，同时也整合了方法，标示学习设计的动态特征。其标示了语义聚合的三个层次（对应灰显类的三个水平层次）。语义上最高层是学习设计，其聚合了组件、目标或预备知识、方法。聚合的最低级是资源、剧、条件与通告。资源被聚合于组件与目标或预备知识，而剧、条件与通告被聚合于方法。组件可以是七种类型中的某一种：角色、属性、属性集合、活动结构、活动、环境、结果即学习成果。除了学习成果，所有元素都来自学习设计信息模型。角色分二种，学习者与教员；资源分为五种，web内容、1d内容、人员、服务支持、档案，资源可以被某学习设计引用但不是信息模型的显性成分。特定类型的组件绑定于特定类型的资源，资源绑定于学习设计的时机参见下表1。

表1 资源绑定学习设计的时机

组件	绑定资源类型	绑定时刻
角色	人员	实例化或执行时
目标或预备知识	Web内容	设计、实例化或执行时
目标或预备知识	1d内容	设计
属性	档案	实例化

学习对象	Web内容	设计、实例化或执行时
学习对象	ld内容	设计
服务	服务支持	设计、实例化或执行时
活动	Web内容	设计、实例化或执行时
活动	ld内容	设计

示例：学习对象的文本描述可以在设计时完成，在学习设计时可以作为资源使用，不过绝对URL可以提供，以便任何时候编辑文件。设计时没有绑定的资源不作为设计说明书的一部分，在处理人员及其档案时也是如此。但是，学习设计中的人员角色以及放在档案中的属性是学习设计的组成部分。

以上模型表明，组件、目标或预备知识以及资源独立于学习设计，它们可以被许多学习对象引用。方法与学习设计有明确的关联，显然是其不可分割的成分，无法独立存在，也不能为其他学习设计所重用。

5.2.2 学习设计的概念架构

图2是学习设计的另一个概念视图，其强调的是类之间的功能联系。

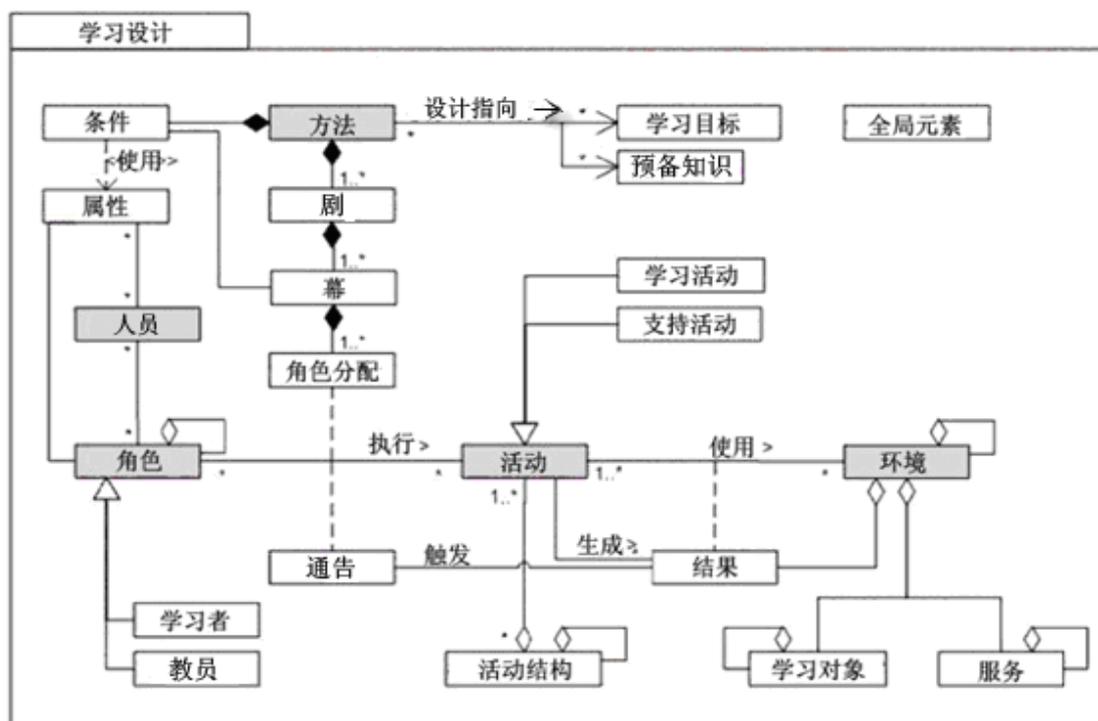


图2 学习设计的全局概念模型

正如图5.2所示，学习设计规范的核心理念是，无论采取何种教学方法，在教学过程中，人员都获得其角色，一般是学习者或教员，其过程都指向一定的学习成果，执行或多或少的结构化学习、支持某环境下的活动。环境包括关联于诸多活动的适当的学习对象与服务。何种角色在何种时刻获得何种活动，取决于方法或通告。注意，以上大部分概念都出现于信息模型，仅有少数只出现于概念模型，如人员、学习成果。

设计方法时,应该满足学习目标(学习者学习成果的描述),并且需要预设某些前提(学习者准入水平的描述)。方法包含若干并发的剧,一个剧包括若干井然有序的幕,一个幕包含若干并发的角色分配,每个角色分配只关联一个角色的一个活动或活动结构。教学过程的模型包含于类似于一次戏剧表演的方法中,一个剧有若干幕,每一幕有若干角色分配,表演时的幕次依次出现(虽然某一幕中角色行为的次序比较复杂)。某一幕的角色分配关联于角色的一个活动,活动描述了角色的所作所为以及角色行为的环境。同理,所分配的活动等价于某幕戏中角色表演时所用的脚本。由于一个幕中有若干角色,它们常常同时上场。

在B层学习设计中,方法常常包含条件(即If-Then-Else规则,其进一步界定活动的可视性以及人员与角色的环境),给出属性的布尔表达式,而属性又可以聚合成属性集合。属性的类型很多,分别代表全局/局部属性以及人员/角色属性。稍后详述。为了让用户设置、查看所呈现内容中的B层属性,模型中设定了所谓的全局元素,设计全局元素时应该运用命名空间囊括所有的内容模式。包含这些全局元素的内容叫做1d内容。通告由学习成果所触发,能够生成由某角色完成的新的活动,得到通告的人不一定是产生学习成果的人。比如,某学生完成某活动(即产生一学习成果),然后另外一个学生或教员得到通告,激活下一个活动。这种机制也可以用于学习设计,此时活动编列的提供取决于先前活动的学习成果的种类(以便灵活地设置任务)。所说的显性角色是学习者与教员角色,它们都可以特化为子角色,但是无词汇指称它们,留待学习设计者来命名、指定其活动。比如,在仿真与游戏中,不同学习者扮演不同角色,在不同环境下完成不同活动。活动可以装配成活动结构。某活动结构聚合了一系列相关活动为一个结构,这种结构又关联于某角色。结构可以将一个活动编列或选择模型化。在一个活动编列中,角色按照给定秩序完成结构中的不同活动。在一个选择中,角色可以从给定角色结构中选择若干数目的活动,此时可以表征学生必须完成二组活动的情形,如此学生可以从活动结构中的(比如说)五组活动自由选择。活动结构还可以引用其他活动结构以及外部的学习单元,以优化此结构。

环境有二种类型:

- 已定位学习对象,通常由包含元数据的URL指定,用户可以进一步将其分类,依据是LOM元数据模型(5.2 学习资源类型)或者所有元素中存在的导出类属性。在EML [1]中,学习对象分为知识对象、工具对象、测试对象。
- 普通服务,服务关联于执行时某具体的服务支持。设计时没有分配URL给服务,但是学习设计执行期间实例化时必须给它一个URL。服务的实例可以是论坛、聊天室、监控工具、搜索引擎等。学习设计时,运行时设置服务的条件要在一抽象水平上指定,比如,在指定聊天室时,必须说明何种学习设计角色有何种访问权限(参与、旁观、修改等)。

注意:倘若学习设计时要用到论坛,并且事先给定其URL,那么包含此学习设计的所有学习单元无论何时何地实例化时都有一个固定的论坛,此时将导致学习设计者或许还不想看到的奇异的学习景观。不过,倘若真有必要,也可以为某正常资源指定固定的URL。

此版规范中,禁止指定这样的访问,一般LMS系统中也很难发现。可以从普通服务中继承,指定新类型来扩展词汇表。由于许多服务都定位于学习设计中某特定情形,需要先给诸多现实人员分配角色,然后再启动服务。由于不同角色有不同服务权限,就需要一种机制,依据具体的服务定义来指定这些权限。EML拥有完整的内容词汇表,基于OASIS文档标准[12]。学习设计中,不能包含任何内容标准,但是允许学习设计标准使用者决定使用哪个标准。为了让终端用户实现执行时交互,全局学习设计元素在B层标准中指定,将其命名于XML内容模

式。建议参考XHTML描述内容，在XHTML中命名全局学习设计元素。

5.2.3 学习单元= 内容包 + 学习设计

学习设计的主要目的是学习单元模型化，将学习设计纳入内容包装，最好（不是必须）纳入内容包装。在本规范中，我们假定需要学习设计与内容包装一起来实现学习单元模型化，如何实现之在这一节介绍。

内容包用XML文档描述其内容，此文档称为“内容清单”（manifest）。该清单可以囊括包中资源的结构视图；每个视图是子内容项的层次结构，该视图称为“组织结构（organizations）”。每个子内容项对应一个资源，或许就是包中的一个物理文件，当然它还可以指向一外部资源。图 3描述了整个内容包装的概念模型。

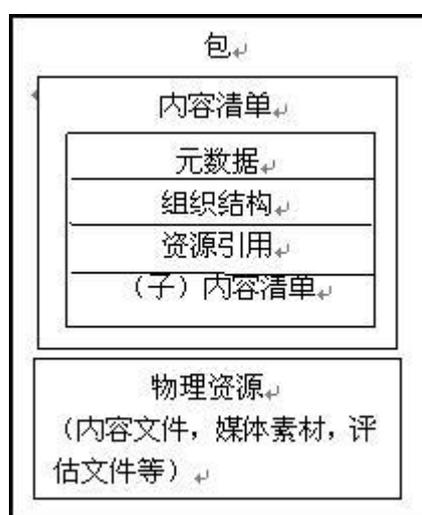


图3 内容包装的结构

包装清单是定义于内容包装规范中的一种信息结构，其内含的包是一个XML文档，有固定的预定义名称即celtsmanifest.xml。这样能够在包内诸多内容文件中很容易找到它。学习设计整合于内容包装结构的情形可参看图4。

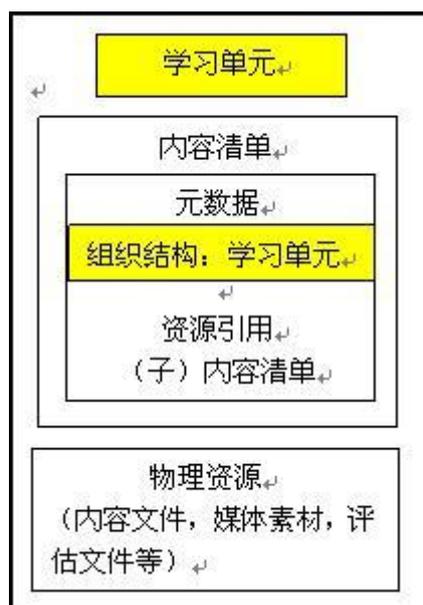


图4 学习单元结构，学习设计整合于内容包装的组织结构

创建学习单元时，学习设计整合于内容包装的方法是，将学习设计元素作为标记<organizations>中的一种组织结构，学习设计元素采用标准命名空间。倘若标准命名空间是“[standard-namespace-for-learning-design]”，此时学习设计元素可以表示如下（忽略无关元素及属性）：

```
<manifest>
  <metadata/>
  <organizations>
    <learning-design xmlns="[standard-namespace-for-learning-design]">
      [add learning design elements here]
    </learning-design>
  </organizations>
  <resources/>
</manifest>
```

其中斜体部分可添入相应的命名空间与元素。在包含学习设计元素的包中，忽略了组织结构的嵌套情况。这种机制符合内容包装的扩展机制。倘若某组织元素包含了学习设计元素，那么同一组织元素中的任何组织元素将被忽略，执行系统只读取学习设计元素。倘若确实需要其他的内容组织元素，可将其囊括于子清单中，同样子包也可以聚合于正常内容包中。

5.3 学习设计的概念词汇表

这一部分概要说明了学习设计规范中的基本概念术语。

学习单元 (Unit of Learning)

学习设计是任何学习单元不可分割的组成部分，学习单元是一抽象的术语，意旨教育或培训中任何有限部分，比如一门课程、一个单元模块、一节课等。值得注意的是，学习单元不仅代表有序学习资源的集合，同时囊括各种预设的活动（比如问题解决、搜索、讨论、相

互评价等活动)、测评、服务以及教师、训练者与其他人员提供的支持条件。采用哪个活动、哪个资源、哪个流程取决于学习单元的学习设计元素。学习单元模块化是内容包装整合学习设计于包内的过程。内容包装之所有叫做学习单元仅仅在于，其包装清单的组织结构囊括了有效的学习设计元素。学习单元通常包括清单、学习设计、物理文件。

学习设计 (Learning Design)

学习设计是这样一种描述方法，其能够帮助学习者在某种学习环境下通过某种编列的学习活动获得学习目标。学习设计是基于设计者的教育原则以及具体的领域与环境变量（比如数学教学的设计不同于语言教学，远距离教学设计不同于面授教学）。文献中有数百种设计，其教学过程的假设各不相同[5]。在日常实践中，大多数教师与培训者有自身的教学原则，因此对相同领域内容有无数的可能设计方案。为了有效的将这些情形各异的设计囊括于e-learning模块，采用了元语言方法，旨在描述所有学习设计，同时又不强求特定方案。

学习设计元素是学习设计规范的根元素，其包括核心元素集合，由学习设计规范添加至现有的内容包装规范，其能够提供资源的语义结构视图以及学习过程信息。以下术语是所添加的关键子内容项。

学习目标 (Learning Objectives)

学习目标是学习者欲完成的学习单元的总学习目标，具体制定时有若干级别描述方法。在学习设计中，设计者可以在二种水平上指定学习目标，各有其优缺点。其一，可以将学习设计定义在全局学习单元水平上；其二，在学习设计时可以为学习目标指定详细的活动内容。设计者可以采用以下几个方法：

- 学习目标只定义在学习单元总体水平上，不定义各个学习活动的子目标或者其对总目标的补充。
- 学习目标只定义在各个学习活动上，不针对学习单元总体。学习单元的学习目标仅仅是不同学习活动的学习目标列表。
- 学习目标定义在二个水平上，即学习单元的学习目标在活动层次上可以描述得更抽象。

预备知识 (Prerequisites)

预备知识指定学习者完成学习单元的总准入条件。和学习目标一样，预备知识可以是学习单元水平上，也可以是各个学习活动水平上。学习目标与预备知识之描述，可以采用可重用能力与教育目标定义格式 (RDCEO)，同时也可以引用简单资源（如文本）来描述学习目标。

组件 (Components)

其定义了学习设计方法部分的“组块”的种种情形。在A层学习设计中，包括角色、活动、环境；在B或C层学习设计中，包括角色、属性、活动、环境。组件应该与方法分开声明，以免在多次使用同一组件时方法的重定义。组件与方法可以类比于菜谱，组件是些调味品，方法是烹调的程序。

角色 (Roles)

角色是学习单元所指定的参与者类型，有二种角色类型，学习者、教员；它们可以进

一步分类，以便学习者扮演不同的角色以适应不同的学习活动，比如基于任务的学习、角色游戏、仿真等。教员也可以进一步分类，指定更具体的角色，比如教师、辅导员、导师等。在多用户学习模型中，角色起到了根本性的作用。角色之命名取决于教学过程与应用背景，有时学习者叫做学生，有时叫做参与者。教员角色的名称就更加多元化了，比如助学者、导师、评价员。各个角色用自身的“头衔”来命名角色，在执行时可以分配多个用户同样的角色，但是应该限制每个角色的最大最小人数。从这种意义上说，角色可用于群体学习。

属性 (Properties)

属性只存在于学习设计规范的B与C层中，基于属性才能建立用户与角色的档案及学档。属性是进行监控、个性化、评价以及用户交互的必备成分。学习设计支持五种属性，即局部属性、局部个人属性、局部角色属性、全局个人属性、全局属性。由此，属性还可以成集合，比如组成窗体。局部属性声明于学习设计，全局属性要由外部声明，但是有一种机制，可以在需要时声明新的全局属性。

全局元素 (Global Elements)

在B、C层学习设计中，为了让用户能够设置、查看教学过程中的属性，提供全局元素作为学习设计规范的独立部分，有四种全局元素：属性设置、属性查看、属性集合设置、属性集合查看。属性设置元素让用户扩展WEB接口，改变某属性的当前值。属性查看元素给出作为学习内容的所需属性值。属性集合设置与属性集合查看元素操作的对象是属性集合。全局元素不是学习设计树的成分，它是单独提供的，一般设计为XML文档，应用XML命名空间（包括XHTML）。只有通过这些元素才能访问相关属性。使用全局元素的内容必须赋予特定的资源类型（即内容包装中的资源元素属性），需要使用“ldcontent”而不是“webcontent”。在将来的规范中，全局元素集合会囊括于内容模式中。

活动 (Activities)

活动是学习设计的学习流程中的核心结构元素。借助于活动，学习环境中的角色、学习对象与服务联系在一起，它们描述了在学习对象与服务组成的特定学习环境中角色必须完成的活动，它们也指定了终端的条件以及终端的工作。活动有二大类：学习活动、支持活动。学习活动旨在为每个用户获得学习目标，各个用户只执行学习活动一次。支持活动是帮助角色完成若干学习活动。执行时若干人员分配于一个角色。实际上，这就意味着支持活动必须多次执行，因为有许多用户需要支持。活动还可以聚合成活动结构，这种机制提供结构化活动，将学习单元变成一个编列或者用户的选择过程。活动引用了其得到执行的环境，任何活动执行过程中，用户至少需要活动的描述，有时需要了解完成活动所需的学习对象与服务组成的学习环境。

学习活动 (Learning Activity)

一个学习活动包括唯一的活动描述及其若干要素，活动描述是用户描述所要完成的活动的实际线索（由用户界面处理），多数情形下活动描述是文本形式（web内容方式）。某些情况下，由音频文件、视频文件等形式描述。无论如何，活动描述是用内容包装中的<item>元素引用的，即向内容包装中引入资源元素。除了环境元素，其他可选元素包括：标题、元数据、学习对象、学习目标、预备知识以及二个新元素，其一完成活动（complete-activity），用于标记何时完成活动，对于A层学习设计，由用户选择或时间限制标识，在B层学习设计中，由属性值设置来触发，其二完成动作（on-completion）设置（指定活动结束后要执行的活动）。在A层学习设计中，“完成动作”只有一个元素即反馈描述，引用用户完成时要显示

的内容。在B层学习设计中，由改变属性值元素进一步扩展；在C层学习设计中，由通告元素进一步扩展。

支持活动 (Support Activity)

支持活动中的大部分元素与学习活动相同，不包含其中的学习目标、预备知识，但另外添加了角色引用元素。角色引用元素指出该活动支持何人。被支持的角色可以有若干人，换言之，所支持的活动对分配此角色的每个用户都要做一遍，如此才算完成。这是其与学习活动的关键区别，后者只执行一遍。比如，教员角色的支持活动是为学习者角色比如学生的报告记分，每个学生都有自己的报告，教员要为每一份报告记分（即重复“给报告记分”这一支持活动）。

活动结构 (Activity-Structure)

活动结构包含以下若干引用：

- 学习活动
- 支持活动
- 子活动结构
- 另一独立学习单元

在学习单元中，引用了HREF作为学习单元的URI。此URI可以是全世界任何唯一标识符，包括URL（W3C命名空间以此来识别唯一命名）。使用内容包装时，这就意味着其引用了内容清单的“标识符”属性，其必须是某种文档规范的世界范围内的唯一标识。如同单一活动，活动结构可能引用若干环境。这就允许同一环境下执行不同的活动编列。当某活动结构引用若干环境时，这些环境便会支配引用活动的环境。在活动结构的层次结构之间，环境不允许继承，相反允许环境的忽略。结果，对于活动结构的各个层次，对环境的相应引用必须指定，必要时还要重复指定。结构包含学习，其提供CP组织结构/子内容项，由此获得了链接诸多资源的结构，这些资源又包含此活动结构的更多信息。

环境 (Environment)

学习活动发生于所谓的“环境”中，它是学习对象、服务、子环境的结构化的集合体。活动与环境的关系可以用活动的语言学描述来揭示。活动中的名词意味着环境中学习对象的存在。对其他人的引用意味着通信服务的存在，某些动词意味着支持服务或工具的存在，比如，活动“阅读此问题并且和同事讨论其解决方案”，它所引用的环境成员是“问题”与“同事”，前者指存在这东西可供阅读，后者指存在可通信的对象（包括通信手段）。

学习对象 (Learning Object)

此处定义的学习对象，指能够完成学习活动或支持活动的、任何可再生可访问的数字或非数字资源。在内容包装中，它们表示为“资源”元素，比如网页、教科书、制作工具（文本处理器、编辑器、计算器等）、测试子内容项，仪器（如显微镜）。学习对象的分类在 LOM 规范中可以找到（在5.2中，学习资源类型的区分如下：练习、模拟、问卷调查、表格、图示、叙述文本、索引、幻灯、考试、实验、问题陈述、自测、讲演）。学习对象可以是其任何一种。它假定执行时系统能够处理这些子内容项。

服务 (Service)

除了需要设计时定义的资源，教学时还有许多所谓的“服务支持”，比如论坛以及其他通讯设施。某些设施无法在设计时指定URL，其必须在本地执行服务实例化分配，因为倘若服务支持在设计时绑定，那么此服务就要被学习设计中所有相关用户访问。倘若需要学习设计与相关用户的唯一的执行时服务(比如聊天室分配给某组学习者，支持具体学习设计的相关教员)，那么就必须创立唯一的执行时服务，并且在学习设计实例以及相关的学习者与教员已经建立之后为其分配一局部URL。为此，需要定义良好的服务类型集合，比如聊天室、论坛、通告频道等，这在普通学习管理系统中都能发现。在学习设计中，使用、设置这样的服务需要在抽象水平上声明，如此声明执行时设施才能根据需求设置必要的设施。在学习设计规范中，服务支持的抽象声明叫做“service”，其实例化叫做“service facility”。现在服务类型包括：发送电子邮件(send-mail)，会议(conference)，监控(monitor, level B)，索引搜索(index search)。选择哪些服务需要学习共同体决定。我们只是给出在线学习环境中最常用的服务。

发送电子邮件服务 (Send-Mail Service)

任何在线学习设施中都要有邮件收发功能，由e-mail客户端完成。不过，学习系统需要了解其师生的所有e-mail地址，以便实现群发。这一信息存在于系统执行时。为了帮助用户在执行学习单元的同时将邮件发给其他人，有必要在服务端声明邮件发送设施。倘若邮件发送服务包含于用户环境，系统执行时必须提供某设施来编辑邮件信息及其附件，并通过e-mail将其发送给学习单元运行时的相关用户。这些用户要么是某角色的所有用户，要么是某角色的特定人选。用户接收邮件是一般使用其e-mail客户端的常规收件箱。

会议服务 (Conference Service)

一种常见的通讯服务是会议。对于会议服务，除了标题与元数据，还需要指定4种会议系统角色，即参与者、观察者、会议组织者、会议主持人，在学习设计中需要指定相关角色。一旦学习设计中的角色分配了相关会议角色，这些信息就可以用来自动设定相应的会议空间。本学习设计规范没有定义会议角色的权限，**而是留给系统实现者**。此会议角色拥有通常意义上权限，学习设计者期望系统实现者也如此理解。会议服务可分为三种，同步会议(比如聊天、视频会议)、异步会议(比如新闻组、聊天)、通告会议(一对多异步会议)。

监控服务 (Monitor Service)

监控服务提供设施以使用户看到自己或其他人的结构化显示的属性。创作者定义LD内容(比如XHTML表以及全局属性元素)来观察相关属性。这种LD内容引用于“item”元素中，一旦要创建监控对象，创作者就要选择是让用户看到其自身档案的属性还是看到有关角色中的所有用户的属性。基于监控对象，人们可以看到自身档案的属性或者某角色的所有用户档案的属性。一旦选定“自身”，各个属性都有一个值。一旦角色选定，就可以看到特定角色所有用户的相关档案属性。此时，学习设计者必须十分小心，虽然只要一个查看属性被指定，其效果是对某角色的所有用户的重复作用。显然，此时在解析学习内容时用户界面的相关目录必须自动更新：

- 倘若查看属性是一文本行，此行中没有其他查看属性，那么就要创建属性值列表，每个属性值单独一行。
- 倘若一文本行有若干查看属性，那么就要创建属性值列表，每个属性值单独一行并且依据查看属性的分组特性将各行分组。
- 倘若查看属性是一表格，那么对角色的各个用户在表中都要创立一新行。

索引搜索服务 (Index-Search Service)

索引搜索服务是帮助学习单元索引化,从而能够搜索学习单元。除了标题、元数据元素,还包含索引元素、搜索元素。索引元素包装了索引方面的操作,用于建立搜索服务,索引在后台确定(用户不可见性)。可见性的是搜索元素。索引的功能取决于搜索元素:

- 倘若是自由文本搜索,那么被索引的资源指向了该索引(比如html中的下划线)。
- 倘若是索引搜索,无论有无引用目标,索引的是具有相同类的元素,包括下划线子内容项,其拥有内容表的形式。

搜索元素指定用户如何访问被索引的实体,此时有三种可能:

- 用户获得自由文本搜索对话框,此时可以用自由文本形式搜索索引(即要为自由文本检索建立索引)。自由文本检索的语法取决于实现方式(比如 Google or Altavista的格式)。
- 用户面临的是文本索引(内容表格),带有超链接(或者其他形式,比如页码),指向相应的资源。
- 用户面临的是文本索引(内容表格),没有超链接,指向相应的资源,比如提供学习单元结构的信息。

方法 (Method)

方法包含学习数据规范的二个核心部分:剧(play)、条件(condition),以及相关完成活动语句与完成动作语句。

剧 (Play)

学习设计的核心内容展现于剧。剧指定了真实的学习设计与教学过程,引用以上声明的组件。在剧中,需要指定何种角色以何种顺序完成何种活动。在阅读学习设计时,主要是阅读剧。人是如此,机器亦如此。剧中未引用的组件不会出现于系统执行时,剧的模型化类似于戏剧之剧乃至若干幕与角色分配。通常剧有若干活动组成的序列,对各个活动,不同角色有不同活动并且并发完成。活动之转换取决于满足了学习设计的完成语句需求。

条件 (Conditions)

条件只存在于学习设计的B层与C层,其使用的目的在于进一步优化属性,为学习设计添加个性化特色。条件的基本形式是:IF [表达式] THEN [显示、隐藏、改变、通告]。

表达式多数以学习者档案属性来定义(比如IF pre-knowledge-english='4')。条件之效果因人而异,即便用户分配了相同的角色。条件的作用域是当前活动的上下文。实际上,条件在选择类型的活动结构中是最有用的。

通告 (Notification)

通告只出现于学习设计C层规范,借助于通告,可以给角色发消息,给角色分配新的学习活动或支持活动,倘若发生了某事件。事件类型如下:

- 某活动完成动作
- 某幕完成动作

- 某剧完成动作
- 某学习单元完成动作
- 某条件中表达式为真
- 某属性值发生改变

子内容项 (Item)

正如某组件、学习目标、预备知识需要某资源，“item”元素在内容包装的组织结构中也起相似的作用。学习设计为这些子内容项提供了语义背景，系统执行时可以了解如何处置该资源。比如，

```
<learning-objectives><item identifierref="o123"/></learning-objectives>
```

显然其中有标识符‘o123’的资源是学习目标。又如：

```
<activity-description><item identifierref="o345"/></activity-description>
```

此时子内容项是活动，系统执行时将学习目标定位于用户界面某个地方。这与活动描述不同，活动描述之处理不同于其他学习内容（依据具体系统实现，即信息模型的行为描述之边界）。

6. 信息模型

三个层次的信息模型将以下列格式分别加以描述：

- 根据图 5.2 得到的 UML 概念模型，在不同层次只体现相应要素，无关要素将被删除。
- 树状图描述学习设计中相互联系的部分及全局图。
- 信息表描述信息结构。图表可相互阐释并陆续展开。

注释：出于表格描述的需要，同一个要素在不同的表中可能描述多次。不同表中对同一个要素的描述保持一致，若同一个表中出现多次，该信息不再重复，而是标明“同上”。

图的描述采用以下格式：

- 只显示要素（不显示属性）。
- 以树状结构表示，从左至右浏览。左边的要素包含右边要素。最左边的要素是顶部节点。
- 整个文档只有两个树状图：顶层为“学习设计”的树状图，以及稍小些的顶层为“全局要素”的树状图。根据描述的需要，“学习设计”树状图将按照从左至右、从上到下的顺序逐渐展开。完全没有展开的要素在后续的图中不会展开。
- 图中以“<”表示“OR”的关系。
- 图中以“[”表示“AND”的关系。
- “*”表示要素在容器中发生零或多次。
- “+”表示要素在容器中发生一次或多次。
- “?”表示要素可选。
- 如果要素名字之前没有任何符号，则表示出现正好一次。

表格是对图中涉及要素及其属性的描述，采用以下的格式：

No: 表示层级图中要素序号。

Name: 要素名称或属性，属性以斜体表示。

无注释要素属于 A 层学习设计。

(*) 要素属于 B 层学习设计 (B 包含 A)。

(**) 要素属于 C 层学习设计 (C 包含 A 和 B)。

(cp) 要素属于内容包装规范。

Explanation: 解释，指出要素的含义及功能。

Reqd: 先决条件，指出要素或属性是强制的 (M) 还是可选的 (O)。

Mult: 重数，指出要素或属性的重复次数。

1, 出现 1 次。

0..1, 可选，出现 0 次或 1 次。

0..n, 出现 0 次或多次。

1..*, 出现 1 次或多次。

-, 重数未确定。对于顶层要素，其重数在使用环境才会确定。

Type: 指出要素的属性类型。

Container: 容器，将同一类的要素进行包装。

Choice: 选择，从多种要素中进行选择并包装。

Sequence: 序列，将多种要素按顺序进行包装。

Group: 集合，一种占位符，是将多次复用的要素包装为谱系结构，这一谱系会在独立的表中展开。

String: 字符串，一种占位符，提供给字符数据。

Any: 任意，一种占位符，提供给任何其他结构。

Empty: 空，结束标记，不含字符数据。

在信息模型中，对属性进行命名的通用规则如下所示。

为了区别学习设计模型内部的引用与内容包装规范的引用，采用以下规则：

1. 命名为“ref”(IDREF)的属性指称学习设计内部的定位引用，例如：`<act-ref ref=""/>`，指的是在学习设计内部对一个行为的引用。

2. 附带“identifierref”属性的要素，指称内容包装规范中的资源引用。例如：`<item identifierref="...">`，引用的是一个资源。名为“uri”的属性指称 URI，也就是全球统一定位符，属性“href”用来对 URI 进行引用。

6.1 A 层信息模型

6.1.1 概念模型

A 层概念模型的 UML 图如图 5 所示。

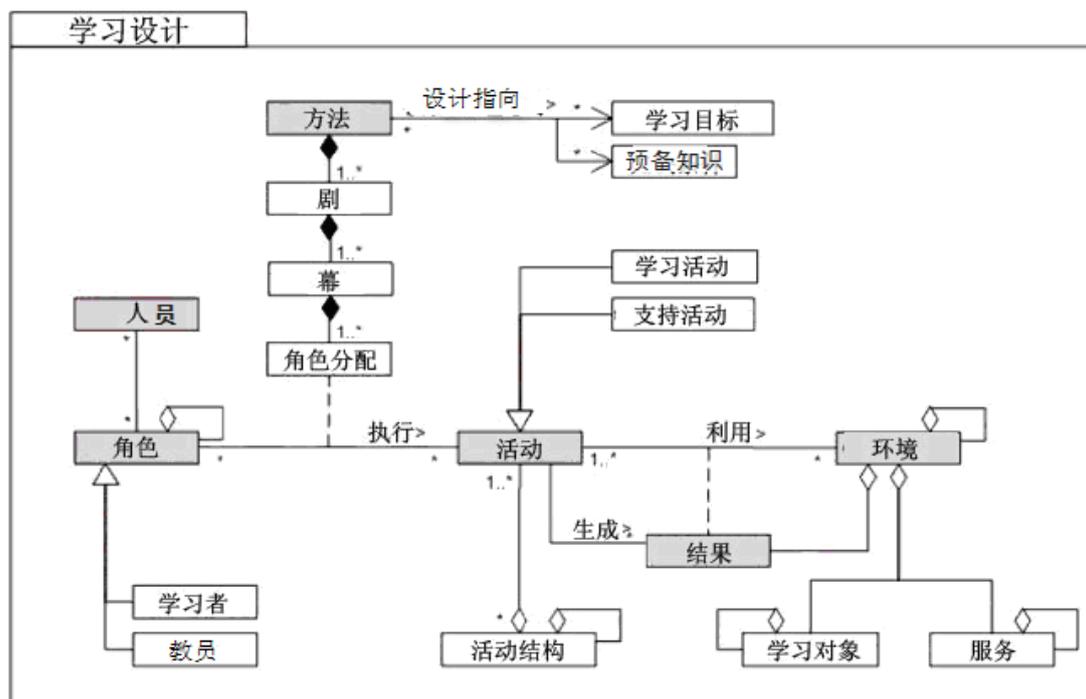


图 5 A 层概念模型

6.1.2 学习设计信息表

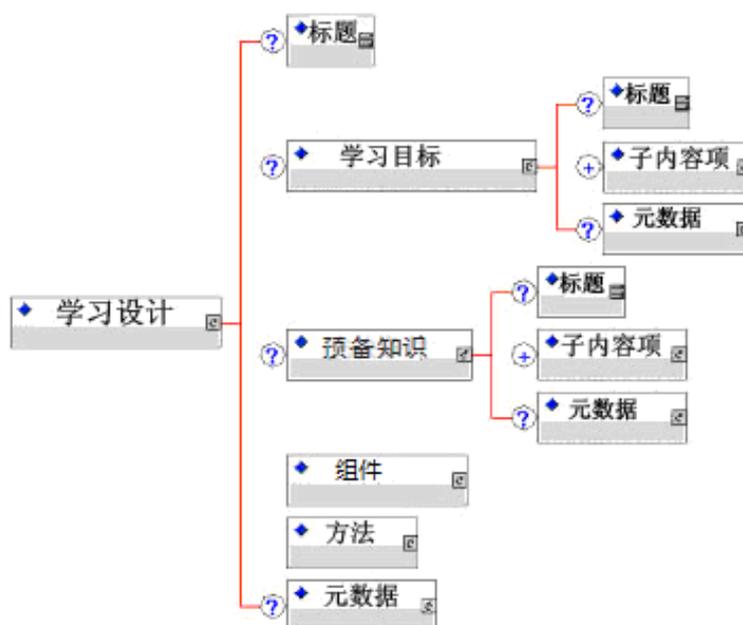


图 6 学习设计信息图

表2 学习设计信息表

学习设计						
编号	名称	英文名	解释	先决条件	重数	类型
0	学习设计	learning-design	该要素对学习设计进行规定	-	-	序列
0.1	标识	identifier	学习设计中文件的唯一标识符 (ID)	M	1	标识符
0.2	版本	version	版本号	O	1	字符串
0.3	uri	uri	指定一个 URI	M	1	任何 URI
0.4	级别	level	指定文档实例适合的最低学习设计级别, 用下列字符指定级别: A, B, C, a, b,c。 可能的值: A, B, C, a, b, c	M	1	标记
0.5	使用编列	sequence-used	布尔值, 当设置为“true”时, 文档实例的适当位置包含了简单编列规范。缺省为“false”。 可能的值: true, false 缺省值: false	O	1	布尔值
0.6	标题	title	适合用户界面呈现的资源名称	O	0..1	字符串
0.7	学习目标	learning-objectives	学习目标描述预期的学习成果。学习目标和学习条件包含一集合标准化的子内容项, 指向资源或具体子内容项。资源类型与学习目标相连接, 可以是 webcontent, ldcontent, 或指向一个 RDECO 大纲。可以在两个地方指定学习目标与学习条件: 学习设计层 (在学习设计的根部), 或者在学习活动层 (在学习活动内部)。前者可以是一般性的描述, 后者更为具体。学习目标有两类: 1.可读的人类语言描述 (该类条目指向文本资源), 或者2.可被机器理解的描述。这些通过href属性指向的资源进行处理。学习目标可以是用户定义的或预设的。对于后一种情况, 会有指向的文本链接 (通过href)。	O	0..1	序列
0.7.1	子内容项模型	{itemmodel}	大纲集合	M	1	集合
0.8	预备知识	prerequisites	学生进入学习的预备知识, 例如所需的准备知识。该类子内容项的形式见“学习目标”项的描述。	O	0..1	序列
0.8.1	子内容项模型	{itemmodel}	同上 (0.7.1)	M	1	集合
0.9	组件	components	描述方法部分所用到的组件。	M	1	序列
0.10	方法	method	方法包含了按一定顺序组成的要素, 以反映学习过程的动态性。由一个或多个操作 (可由学习单元的运行脚本解释) 以及学习单元的结束声明组成。	M	1	序列
0.11	元数据	metadata	元数据占位符, 包括元数据, 使用其名字空间。	O	0..1	序列

6.1.3 子内容项模型信息表

参见前图，此信息表来自内容包装规范。

表 3 子内容项模型信息表

子内容项模型						
编号	名称	英文名	解释	先决条件	重数	类型
0.1	标题	title	适合用户界面呈现的资源名称。	O	0.1	字符串
0.2	子内容项	item	结构中的一个节点，指向某一资源。	M	1..*	序列
0.2.1	标识	identifier	学习设计文件中的唯一标识符(ID)。	O	1	标识符
0.2.2	标识引用	identifierref	指向内容包装中的某一资源标示符(学习设计外部)。	O	1	标识符引用
0.2.3	可见性	isvisible	初始可见性属性； 可能的值：true（缺省），或false 缺省值：true	O	1	布尔值
0.2.4	参数	parameters	在运行时可传递的参数。	O	1	字符串
0.2.5	标题	title	同上（0.1）	O	0.1	字符串
0.2.6	子内容项	item	同上（0.2）	O	0..*	序列
0.2.7	元数据	metadata	元数据占位符，包括元数据，使用其名字空间。	O	0.1	序列
0.3	元数据	metadata	元数据占位符，包括元数据，使用其名字空间。	O	0.1	序列

6.1.4 组件信息表

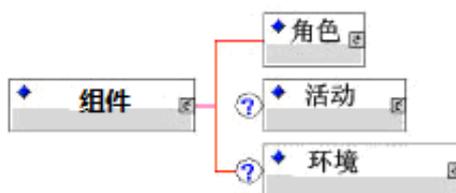


图 7 组件信息图

表 4 组件信息表

组件						
编号	名称	英文名	解释	先决条件	重数	类型
0	组件	components	规定在方法部分所使用的构成要素。	-	-	序列
0.1	角色	roles	该要素规定在学习设计中的特定角色。由两类角	M	1	序列

组件						
编号	名称	英文名	解释	先决条件	重数	类型
			色集合成了角色序列：学习者和教员。以href指向一全局角色（例如某机构指定的），当指定全局角色并连接其属性时这是必需的。使用href属性规定全局角色。其他申报是局部的（如信息）。全局角色只能由机构指定，并在学习设计中提供URI，不可自行通告。URI不必指向某一资源地址，只用来作为全球唯一标识。角色的“标识”属性可用来表示一组同类角色（学习者或教员）。学习设计中，至少规定一个学习者角色。在机构的实施中，角色名称是固定的。例如在大多数大学，学习者的角色标识是“学生”。			
0.1.1	标识	identifier	学习设计中文件的唯一标识符(ID)。	O	1	标识符
0.2	活动	activities	该要素包含各类活动的选项，包括“活动结构”。	O	0..1	选择
0.3	环境	environments	是环境要素的容器。	O	0..1	容器

6.1.5 角色信息表

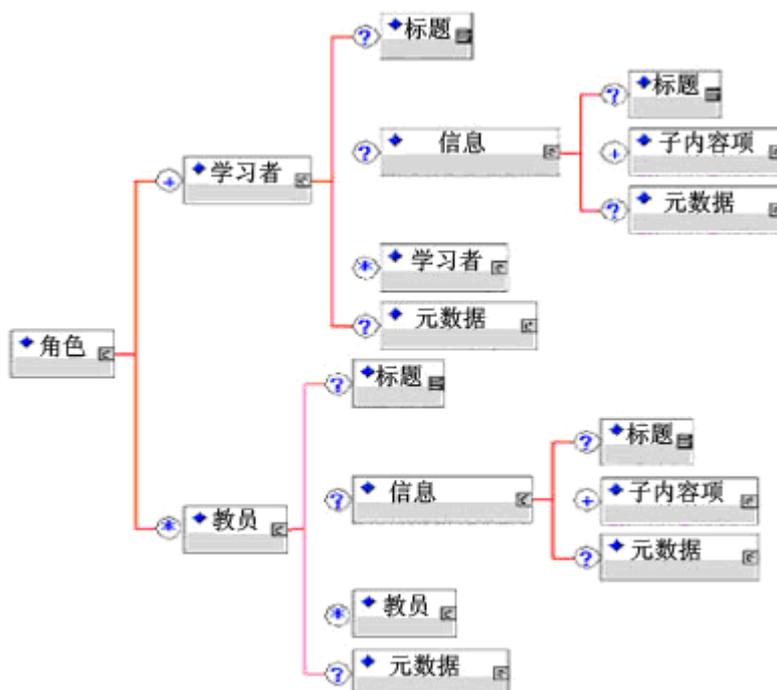


图 8 角色信息图

表 5 角色信息表

角色						
编号	名称	英文名	解释	先决条件	重数	类型
0	角色	Role	该要素规定在学习设计中的特定角色。由两类角色集合成了角色序列：学习者和教员。以href指向一全局角色（例如某机构指定的），当指定全局角色并连接其属性时这是必需的。使用href属性规定全局角色。其他申报是局部的（如信息）。全局角色只能由机构指定，并在学习设计中提供URI，不可自行通告。URI不必指向某一资源地址，只用来作为全球唯一标识。角色的“标识”属性可用来表示一组同类角色（学习者或教员）。学习设计中，至少规定一个学习者角色。在机构的实施中，角色名称是固定的。例如在大多数大学，学习者的角色标识是“学生”。	-	-	序列
0.1	标识	identifier	学习设计中文件的唯一标识符(ID)。	O	1	标识符
0.2	学习者	learner	每个学习设计至少有一个学习者。学习者可以是“嵌套”的，也就是该角色可分为“子角色”。学习者模型的名称用来为该角色命名。例如在一个教育游戏中，你可以将子学习者分别命名为“首领”和“参与者”。	M	1..*	序列
0.2.1	新建	create-new	这一属性指出该角色在运行时是否可以创建新的实例。若值为“not-allowed”，那么该角色只有一个实例；若值为“allowed”（缺省），那么运行系统需要提供创建新实例的机制。随着角色新实例的创建，其子角色也随之创建。 可能的值： allowed, not-allowed 缺省值： allowed	O	1	标记
0.2.2	超链接	href	指向一个URI	O	1	任何URI
0.2.3	标识	identifier	同上	M	1	标识符
0.2.4	匹配者	match-persons	该属性适用于多子角色（例如主席、秘书、成员）。匹配可以是唯一的，即具有了主席角色后就不能同时担任其他角色。若匹配不是唯一的，则可同时任多个子角色（缺省设置）。 可能的值： exclusively-in-roles, not-exclusively	O	1	标记
0.2.5	最大人数	max-persons	规定在开始运行的时候与某角色绑定的最大人数。若最大人数与最小人数为空，则表示没有限制。使用时，遵循以下规则：	O	1	非负整数

角色						
编号	名称	英文名	解释	先决条件	重数	类型
			$0 \leq \text{min-persons} \leq \text{max-persons}$			
0.2.6	最小人数	min-persons	规定在开始运行的时候与某角色绑定的最小人数。若最大人数与最小人数为空,则表示没有限制。使用时,遵循以下规则: $0 \leq \text{min-persons} \leq \text{max-persons}$	O	1	非负整数
0.2.7	标题	title	适合用户界面呈现的资源名称	O	0..1	字符串
0.3	教员	staff	教员可以是“嵌套”的,也就是该角色可分为“子角色”。例如教育机构可能将教员分为导师、评估者、教师等。	O	0..*	序列
0.3.1	新建	create-new	同上	O	1	标记
0.3.2	超链接	href	同上	O	1	任何URI
0.3.3	标识	identifier	同上	M	1	标识符
0.3.4	匹配者	match-persons	同上	O	1	标记
0.3.5	最大人数	max-persons	同上	O	1	非负整数
0.3.6	最小人数	min-persons	同上	O	1	非负整数
0.3.7	标题	title	同上	O	0..1	字符串
0.3.8	信息	information	同上	O	0..1	序列
0.3.8.1	子内容项模型	{itemmodel}	同上	M	1	集合
0.3.9	教员	staff	同上	O	0..*	序列
0.3.10	元数据	metadata	元数据占位符,包括元数据,使用其名字空间。	O	0..1	序列

6.1.6 活动信息表

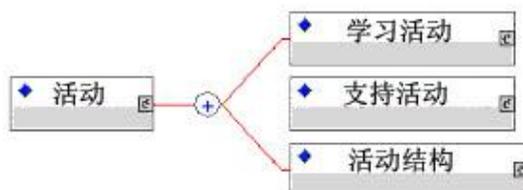


图9 活动信息图

表6 活动信息表

活动						
编号	名称	英文名	解释	先决条件	重数	类型
0	活动	activities	该要素包含各类活动的选项，包括“活动结构”（activity-structure）。	-	-	选择
0.1	学习活动	learning-activity	该要素包含一系列学习活动。	M	1	序列
0.1.1	标识	identifier	学习设计中文件的唯一标识符(ID)。	M	1	标识符
0.1.2	可见性	isvisible	初始可见性属性。 可能的值：true（缺省），或false 缺省值：true	O	1	布尔值
0.1.3	参数	parameters	在运行时可传递的参数	O	1	字符串
0.2	支持活动	support-activity	该要素包含一系列支持活动。支持活动可选择性地与某一角色绑定。也就是说该活动可为其所支持的角色重复使用（学习者/教员）。支持活动一般是教员为了对学习者提供支持而执行的（例如指导者）。但是在某些教学模式下，学习者也能够相互提供支持（同伴教学）。教员也可以为教员提供支持。如果设置了角色引用选项，那么支持活动将对所支持角色的每一个用户执行。也就是说，为受支持的角色中的每一个实例重复该活动。若角色引用未设置，则表明支持活动只是一个独立活动（同学习活动）。	M	1	序列
0.2.1	标识	identifier	同上	M	1	标识符
0.2.2	可见性	isvisible	同上	O	1	布尔值
0.2.3	参数	parameters	同上	O	1	字符串
0.3	活动结构	activity-structure	活动结构是按一定顺序排列或选择的学习活动的组合。树的操作深度优先（而不是宽度）。	M	1	序列
0.3.1	标识	identifier	同上	M	1	标识符
0.3.2	重复数	number-to-select	若设置了重复数选项，当完成活动数等于设置值时，活动结构结束。重复数必须小于或等于活动数（包括学习单元）。若重复数选项未设置，当结构上所有活动结束时活动结构终止。	O	1	非负整数
0.3.3	排序	sort	排序选项确定可见性的顺序。缺省的可见性顺序下即为活动结构规定的顺序。 可能的值：as-is, visibility-order 缺省值as-is	O	1	标记
0.3.4	结构类型	structure-type	指出活动结构是顺序型的还是选择型的。 可能的值：sequence, selection	O	1	标记

6.1.7 学习活动信息表

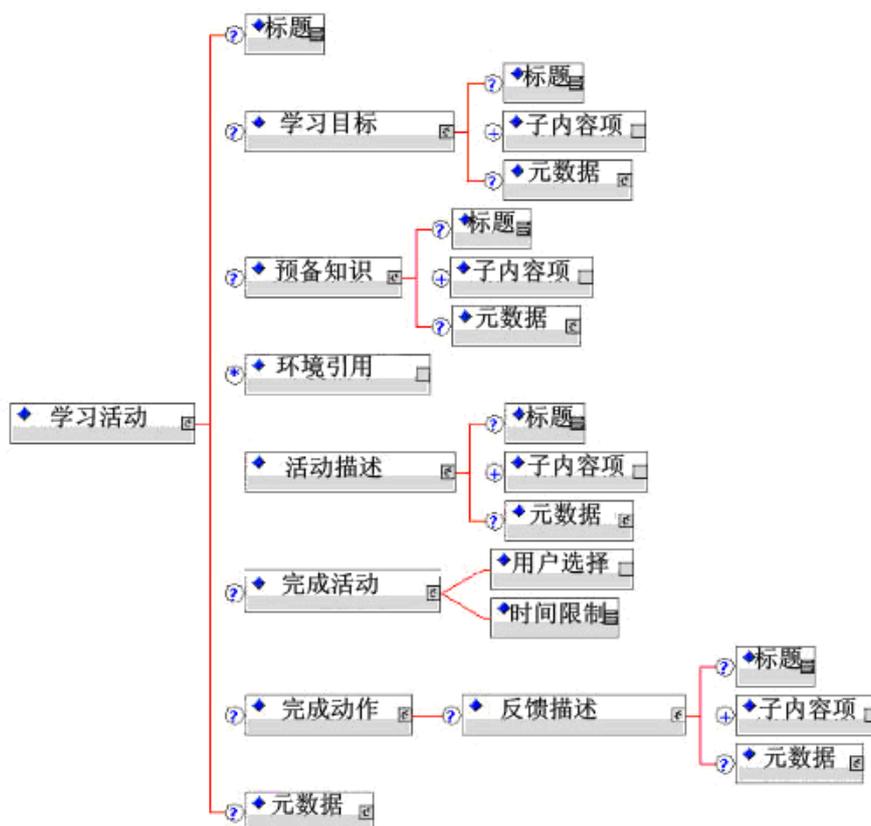


图 10 学习活动信息图

表7 学习活动信息表

学习活动						
编号	名称	英文名	解释	先决条件	重数	类型
0	学习活动	learning-activity	该要素包含一系列学习活动。	-	-	序列
0.1	标识	identifier	学习设计中文件的唯一标识符(ID)	M	1	标识符
0.2	可见性	isvisible	初始可见性属性。 可能的值: true (缺省), 或false 缺省值: true	O	1	布尔值
0.3	参数	parameters	在运行时可传递的参数。	O	1	字符串
0.4	标题	title	适合用户界面呈现的资源名称。	O	0..1	字符串
0.5	学习目标	learning-objectives	学习目标描述预期的学习成果。学习目标和学习条件包含一组标准化的子内容项, 指向资源或具体子内容项。资源类型与学习目标相连接, 可以是 webcontent, ldcontent, 或指向一个 RDECO 大纲。可以在两个地方指定学习目标与学习条件: 学习设计层 (在学习设计的根部), 或者在学习活动层 (在学习活动内部)。前者可以是一般性的描述, 后者更为具体。学习目标有两类: 1.可读的人类语言描述 (该	O	0..1	序列

学习活动						
编号	名称	英文名	解释	先决条件	重数	类型
			类条目指向文本资源), 或者2.可被机器理解的描述。这些通过href属性指向的资源进行处理。学习目标可以是用户定义的或预设的。对于后一种情况, 会有指向的文本链接(通过href)。			
0.5.1	子内容项模式	{itemmodel}	大纲集合	M	1	集合
0.6	预备知识	prerequisites	学生进入学习的预备知识, 例如所需的准备知识。该类子内容项的形式见“学习目标”项的描述。	O	0..1	序列
0.6.1	子内容项模式	{itemmodel}	同上	M	1	集合
0.7	环境引用	environment-ref	指向当前包所在环境。	O	0..*	空
0.7.1	引用	ref	指向当前学习设计内的一个标识。	M	1	标识符引用
0.8	活动描述	activity-description	别名: 任务。活动描述为用户提供所要完成任务的提示(在用户界面中呈现)。多数情况下, 以文本描述活动(Web内容的形式或Id内容的形式)。有些情况下也可以是音频文件(Web内容), 视频文件或其他形式。活动描述也对活动环境进行定义。描述中的每一个名词指向环境中的资源。设计者可以决定对环境中的名词进行精确的表示抑或开放的表示。	M	1	序列
0.8.1	子内容项模型	{itemmodel}	同上	M	1	集合
0.9	完成活动	complete-activity	包含表明活动是否完成的选项。若此要素不出现, 活动设为“completed”。	O	0..1	选择
0.9.1	用户选项	user-choice	此要素适用于活动的完成设置, 规定用户可以选择何时结束活动。也就是说, 在用户界面上, 活动状态是可控的, 用户可以选择“completed”。用户可以设置一次(不能反悔)。一旦设为完成, 表明此活动在运行系统中结束。	M	1	空
0.9.2	时间限制	time-limit	时间限制规定当前学习单元启动以后至活动结束的时间。时间表示格式见期限“duration”格式(信息模式中描述)。从学习单元启动开始计时(见“time-unit-of-learning-started”), 设计者需要注意的是, 为角色、幕和剧设置的时间限制是逻辑时间, 运行时, 剧的时间限制接管对角色、幕的时间设置。在层次B和C, 时间限制可在一个属性(属性集合属性, 局部属性类型, 数据类型=字符串, 可被创造者声明)中指定。在这种情况下, 设计者可以控制(设置属性)是否可由用户控制属性的值。当对属性集合引用作了规定后, 属性值接管此项内容。	M	1	字符串
0.10	完成动作	on-completion	当一个活动、幕、戏剧或学习单元结束, 该要素所包含的选项被执行。在A层仅包含一个要素选项。此包	O	0..1	容器

学习活动						
编号	名称	英文名	解释	先决条件	重数	类型
			对B和C层可扩展。			
0.10.1	反馈描述	feedback-description	指向一个描述反馈信息的资源（Web内容或Id内容的形式）。完成后此文本为可见性。	O	0..1	序列
0.10.1.1	子内容项模型	{itemmodel}	同上	M	1	集合
0.11	元数据	metadata	元数据占位符，包括元数据，使用其名字空间。	O	0..1	序列

6.1.8 支持活动信息表

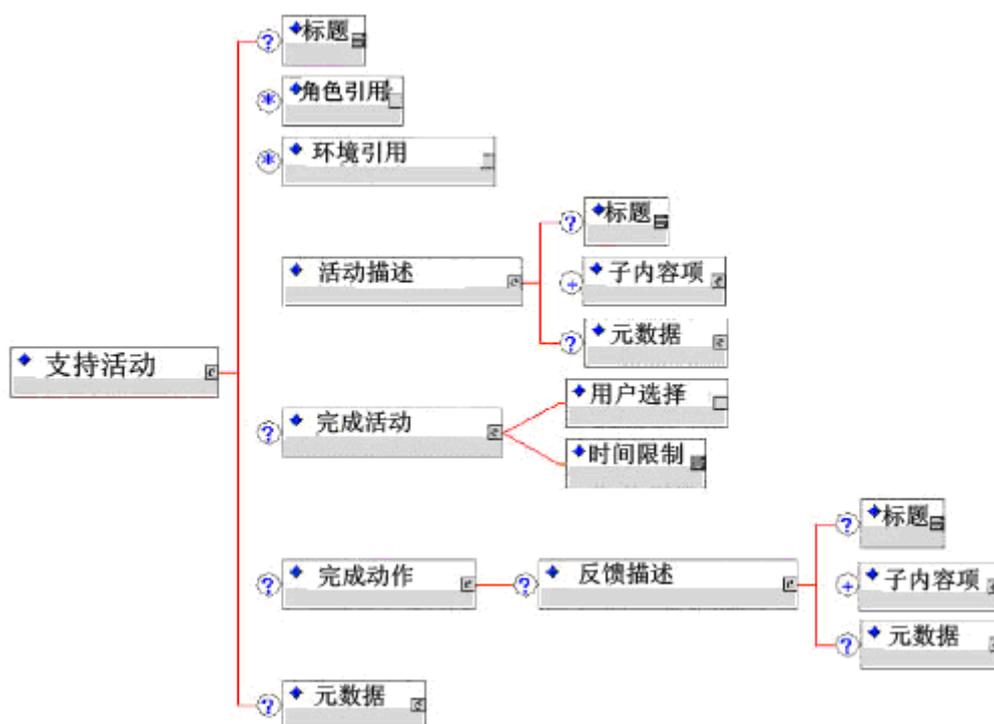


图 11 支持活动信息图

表8 支持活动信息表

支持活动						
名称	名字	英文名	解释	先决条件	重数	类型
0	支持活动	support-activity	该要素包含一系列支持活动。支持活动可选择性地与某一角色绑定。也就是说该活动可为其所支持的角色重复使用（学习者/教员）。支持活动一	-	-	序列

支持活动						
名称	名字	英文名	解释	先决条件	重数	类型
			一般是教员为了对学习者的支持而执行的（例如指导者）。但是在某些教学模式下，学习者也能够相互提供支持（同伴教学）。教员也可以为教员提供支持。如果设置了角色引用选项，那么支持活动将对所支持角色的每一个用户执行。也就是说，为受支持的角色中的每一个实例重复该活动。若角色引用未设置，则表明支持活动只是一个独立活动（同学习活动）。			
0.1	标识	identifier	学习设计中文件的唯一标识符(ID)。	M	1	标识符
0.2	可见性	isvisible	初始可见性属性。 可能的值: true (缺省), 或false 缺省值: true	O	1	布尔值
0.3	参数	parameters	在运行时可传递的参数。	O	1	字符串
0.4	标题	title	适合用户界面呈现的资源名称。	O	0..1	字符串
0.5	角色引用	role-ref	指向角色资源的标识。该要素可用作表达式的操作数。	O	0..*	空
0.5.1	引用	ref	指向学习设计内部的一个标识。	M	1	标识符引用
0.6	环境引用	environment-ref	指向当前包所在环境。	O	0..*	空
0.6.1	引用	ref	同上	M	1	标识符引用
0.7	活动描述	activity-description	别名: 任务。活动描述为用户提供所要完成任务的提示（在用户界面中呈现）。多数情况下，以文本描述活动（Web内容的形式或Id内容的形式）。有些情况下也可以是音频文件（Web内容），视频文件或其他形式。活动描述也对活动环境进行定义。描述中的每一个名词指向环境中的资源。设计者可以决定对环境中的名词进行精确的表示抑或开放的表示。	M	1	序列
0.7.1	子内容项模型	{itemmodel}	大纲集合	M	1	集合
0.8	完成活动	complete-activity	包含表明活动是否完成的选项。若此要素不出现，活动设为“completed”。	O	0..1	选择
0.8.1	用户选项	user-choice	此要素适用于活动的完成设置，规定用户可以选择何时结束活动。也就是说，在用户界面上，活动状态是可控的，用户可以选择“completed”。用户可以设置一次（不能反悔）。一旦设为完成，表明此活动在运行系统中结束。	M	1	空

支持活动						
名称	名字	英文名	解释	先决条件	重数	类型
0.8.2	时间限制	time-limit	时间限制规定当前学习单元启动以后至活动结束的时间。时间表示格式见期限“duration”格式（信息模式中描述）。从学习单元启动开始计时（见“time-unit-of-learning-started”），设计者需要注意的是，为角色、幕和剧设置的时间限制是逻辑时间，运行时，剧的时间限制接管对角色、幕的时间设置。在层次B和层次C，时间限制可在一个属性（属性集合属性，局部属性类型，数据类型=字符串，可被创造者声明）中指定。在此案例中，创造者可为用户控制此属性设置本属性的控制（设置属性）。当一个属性集合引用被指定，在元素中的内容被忽略。	M	1	字符串
0.9	完成动作	on-completion	当一个活动、幕、剧或学习单元结束时，此要素所包含的行为被引发。在A层中仅包含一个要素。此包对B和C层可扩展。	O	0..1	容器
0.9.1	反馈描述	feedback-description	指向一个描述反馈信息的资源（Web内容或Id内容的形式）。完成后此文本为可见性。	O	0..1	序列
0.9.1.1	子内容项模型	{itemmodel}	同上	M	1	集合
0.10	元数据	metadata	元数据占位符，包括元数据，使用其名字空间。	O	0..1	序列

6.1.9 活动结构信息表

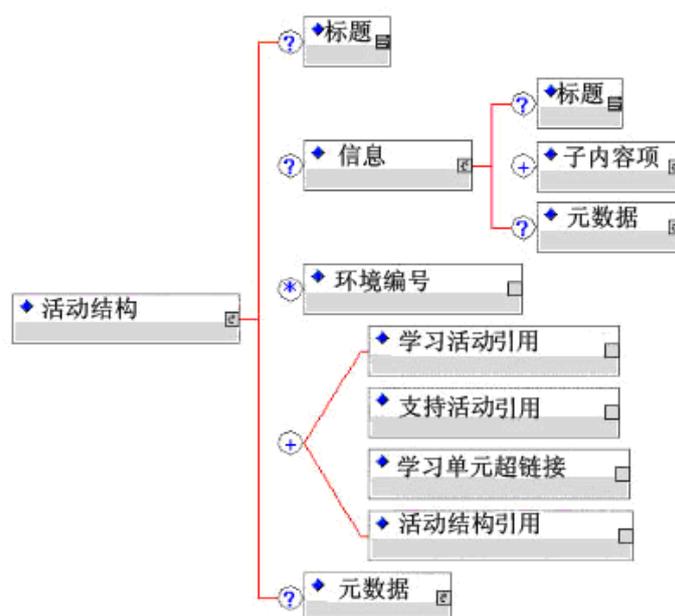


图 12 活动结构信息图

表9 活动结构信息表

活动结构						
编号	名称	英文名	解释	先决条件	重数	类型
0	活动结构	activity-structure	活动结构是按一定顺序排列或选择的学习活动的组合。结构树按照深度优先处理（而非广度优先）。	-	-	序列
0.1	标识	identifier	学习设计中文件的唯一标识符(ID)。	M	1	标识符
0.2	重复数	number-to-select	若设置了重复数选项,当完成活动数等于设置值时,活动结构结束。重复数必须小于或等于活动数(包括学习单元)。若重复数选项未设置,当结构上所有活动结束时活动结构终止。	O	1	非负整数
0.3	排序	sort	排序选项确定可见性的顺序。缺省的可见性顺序下即为活动结构规定的顺序。 可能的值: as-is, visibility-order 缺省值as-is	O	1	标记
0.4	结构类型	structure-type	指出活动结构是顺序型的还是选择型的。 可能的值: sequence, selection	O	1	标记
0.5	标题	title	适合用户界面呈现的资源名称。	O	0..1	字符串
0.6	信息	information	信息要素用来为活动结构提供附加的信息,指定资源所包含的子内容项。	O	0..1	序列
0.6.1	子内容项类型	{itemmodel}	大纲集合	M	1	集合
0.7	环境引用	environment-ref	指向当前包所在环境。	O	0..*	空
0.7.1	引用	ref	指向当前学习设计内的一个标识。	M	1	标识符引用
0.8	选择		Choice	M	1..*	选择
0.8.1	学习活动引用	learning-activity-ref	指向一个学习活动。该要素可用作表达式或算式的操作数。	M	1	空
0.8.1.1	引用	ref	同上	M	1	标识符引用
0.8.2	支持活动引用	support-activity-ref	指向一个支持活动。该要素可用作表达式或算式的操作数。	M	1	空
0.8.2.1	引用	ref	同上	M	1	标识符引用
0.8.3	学习单元超链接	unit-of-learning-href	该要素可用作表达式或算式的操作数。该元素用来指向一个外部学习单元的要素 (unit-of-learning (uol))。该指向可以是同一个包内的(使用相对URI),或者包外的(使用绝对URI)。需要将文件ID添加到链接。其用法与	M	1	空

活动结构						
编号	名称	英文名	解释	先决条件	重数	类型
			标识符引用在XML文件中的引用相同，指向一个活动结构、学习活动、支持活动或环境因素的ID。注释：这等价于一个简单或“空名”的XPointer，这有一个文档：URI#ID并是HTML片段标识符的XML等价物。在XML模式中任何同一资源标识符结构都支持此形式。			
0.8.3.1	超链接	href	指向一个URI。	M	1	任何URI
0.8.4	活动结构引用	activity-structure-ref	一个活动结构的引用。	M	1	空
0.8.4.1	引用	ref	同上	M	1	标识符引用
0.9	元数据	metadata	元数据占位符，包括元数据，使用其名字空间。	O	0..1	序列

6.1.10 环境信息表

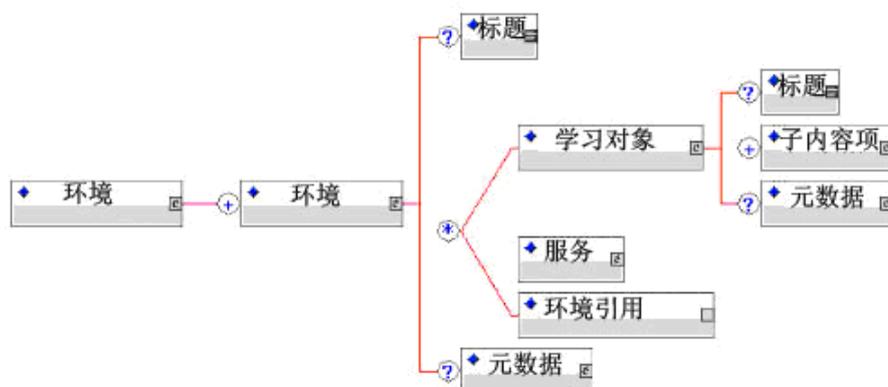


图 13 环境信息图

表10 环境信息表

环境						
名称	名字	英文名	解释	先决条件	重数	类型
0	环境	environments	是环境要素的容器。	—	—	容器
0.1	环境	environment	包含一系列能构成环境的要素。简单编列要素的名字空间可应用于环境，以支持环境要素的编列。若未提供编列信息，环境中所有信息将按既定顺序显示。	M	1..*	序列
0.1.1	标识	identifier	学习设计中文件的唯一标识符(ID)。	M	1	标识符

环境						
名称	名字	英文名	解释	先决条件	重数	类型
0.1.2	标题	title	适合用户界面呈现的资源名称。	O	0..1	字符串
0.1.3	选择		Choice	O	0..*	选择
0.1.3.1	学习对象	learning-object	通过引入大纲 (如 QTI) 或引用子内容项要素资源的方式整合学习对象。简单编列要素名字空间可应用于学习对象, 以支持子内容项的编列。若未提供编列信息, 所有学习对象将按既定顺序显示。	M	1	选择
0.1.3.1.1	类	class	指学习设计或内容要素的类属性的值, 包含一个CDATA字符串。和HTML一样, 一个CDATA字符串可指定多个类, 以空格分开, 其排列优先顺序同CSS的规定 (见 http://www.w3.org/style/css)。原则上每一个要素都具有类属性。在HTML 4.0和XHTML[14]中, “Class”是W3C定义的全局属性。该属性为一个类分配一个或一套类名。许多要素可能被分配相同的类名。多个类名以空间分开。可以类对要素进行语义上的分组, 也可结合LD条件及样式表进行操作。当向Web客户传递一个学习对象时, 类属性及其值也包含其中。	O	1	字符串
0.1.3.1.2	标识	identifier	同上	M	1	标识符
0.1.3.1.3	可见性	isvisible	初始可见性属性。 可能的值: true (缺省), 或false 缺省值: true	O	1	布尔值
0.1.3.1.4	参数	parameters	在运行时可传递的参数。	O	1	字符串
0.1.3.1.5	类型	type	学习对象的类型 (例如知识对象、工具对象、测试对象等)。所用词汇可能是学习资源类型 (“learning resource type”, IEEE LTSC LOM)。	O	1	字符串
0.1.3.1.6	编列	Sequence	Sequence	M	1	序列
0.1.3.1.6.1	标题	title	同上	O	0..1	字符串
0.1.3.1.6.2	子内容项	item	结构中的一个节点, 指向一个资源。	M	1..*	序列
0.1.3.1.6.2.1	标识符	identifier	同上	O	1	标识符
0.1.3.1.6.2.2	标识符引用	identifierref	指向内容包中的一个标识 (学习设计之外)。	O	1	标识符引用
0.1.3.1.6.2.3	可见性	isvisible	同上	O	1	布尔值
0.1.3.1.6.2.4	参数	parameters	同上	O	1	字符串
0.1.3.1.6.2.5	标题	title	同上	O	0..1	字符串
0.1.3.1.6.2.6	子内容项	item	同上	O	0..*	序列
0.1.3.1.6.2.7	元数据	metadata	元数据占位符, 包括元数据, 使用其名字空间。	O	0..1	序列
0.1.3.1.6.3	元数据	metadata	元数据占位符, 包括元数据, 使用其名字空间。	O	0..1	序列
0.1.3.1.7	编列		编列	M	1	序列

环境						
名称	名字	英文名	解释	先决条件	重数	类型
0.1.3.1.7.1	文档规范	schema	标示将使用的文档规范。	O	0..1	字符串
0.1.3.1.7.2	文档规范版本	schemaversion	标示将使用的文档规范版本。	O	0..1	字符串
0.1.3.1.8	子内容项模型	{itemmodel}	一组文档规范。	M	1	集合
0.1.3.2	服务	service	与学习单元运行绑定的服务通告。为了使服务设置过程自动化,需要将学习设计的运行数据翻译为会议系统的配置格式,以便会议系统自动启动。这是一个执行中问题。也可由系统管理器读取这些信息,并手工启动会议系统,但最好是自动启动以减轻系统负荷。服务是可扩展的,当一个服务实现时,运行系统将记录一个“context”句柄,内容涉及服务与谁绑定,服务为谁提供。子内容项要素的标识符引用属性会引用服务,子内容项是环境内部的。环境与活动关联,或可能直接与角色关联。活动或角色组成服务的使用情境,担当角色的用户可访问这些服务。	M	1	选择
0.1.3.2.1	类	class	同上	O	1	字符串
0.1.3.2.2	标识	Identifier	同上	M	1	标识符
0.1.3.2.3	可见性	isvisible	同上	O	1	布尔值
0.1.3.2.4	参数	parameters	同上	O	1	S字符串
0.1.3.3	环境	environment-ref	指向当前包所在环境。	M	1	空
0.1.3.3.1	引用	ref	指向当前学习设计内的一个标识。	M	1	标识符引用
0.1.4	元数据	metadata	元数据占位符,包括元数据,使用其名字空间。	O	0..1	序列

6.1.11 服务信息表

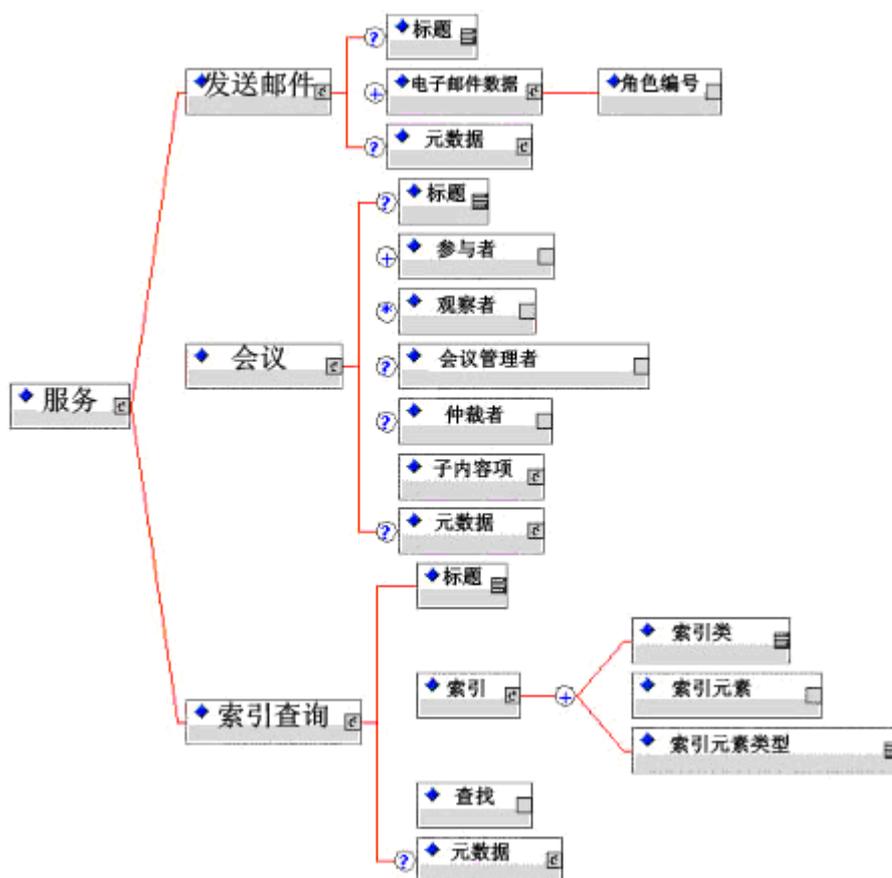


图 14 服务信息图

表11 服务信息表

服务						
编号	名称	英文名	解释	先决条件	重数	类型
0	服务	service	与学习单元运行绑定的服务通告。为了使服务设置过程自动化，需要将学习设计的运行数据翻译为会议系统的配置格式，以便会议系统自动启动。这是一个执行中问题。也可由系统管理器读取这些信息，并手工启动会议系统，但最好是自动启动以减轻系统负荷。服务是可扩展的，当一个服务实现时，运行系统将记录一个“context”句柄，内容涉及服务与谁绑定，服务为谁提供。子内容项要素的标识符引用属性会引用服务，子内容项是环境内部的。环境与活动关联，或可能直接与角色关联。活动或角色组成服务的使用情境，担当角色的用户可访问这些服务。	-	-	选择
0.1	类	class	指学习设计或内容要素的类属性的值，包含一个CDATA字符串。和HTML一样，一个CDATA字符串	0	1	字符串

服务						
编号	名称	英文名	解释	先决条件	重数	类型
			可指定多个类，以空格分开，其排列优先顺序同CSS的规定（见 http://www.w3.org/style/css ）。原则上每一个要素都具有类属性。在HTML 4.0和XHTML[14]中，“Class”是W3C定义的全局属性。该属性为一个类分配一个或一套类名。许多要素可能被分配相同的类名。多个类名以空间分开。可以类对要素进行语义上的分组，也可结合LD条件及样式表进行操作。当向Web客户传递一个学习对象时，类属性及其值也包含其中。			
0.2	标识	identifier	学习设计中文件的唯一标识符(ID)。	M	1	标识符
0.3	可见性	isvisible	初始可见性属性。 可能的值: true (缺省), 或false 缺省值: true	O	1	布尔值
0.4	参数	parameters	在运行时可传递的参数。	O	1	字符串
0.5	发送邮件	send-mail	该服务用来向用户角色发送邮件，B与C层的属性中有邮件地址。	M	1	序列
0.5.1	选择	select	固定选项：“all-persons-in-role”或“persons-in-role”。对于第一个选项，信息只发送给角色，所有该决色的用户都能收到信息。对于第二个选项，可选择对该角色的某些用户发送信息。 可能的值: <i>all-persons-in-role</i> , <i>persons-in-role</i>	M	1	标记
0.5.2	标题	title	适合用户界面呈现的资源名称。	O	0..1	字符串
0.5.3	邮件数据	email-data	为发送邮件用（作为环境中的一个服务或通告）。在B层中，该要素的属性指向为所关联角色准备的邮件数据。在A层中，为指定数据来源，由执行者决定所需数据。分配为该角色的所有用户相关属性（email、用户名）应可用。	M	1..*	容器
0.5.3.1	角色引用	role-ref	指向角色资源的标识。该要素可用作表达式的操作数。	M	1	空
0.5.3.1.1	引用	ref	指向学习设计内部的一个标识。	M	1	标识符引用
0.5.4	元数据	metadata	元数据占位符，包括元数据，使用其名字空间。	O	0..1	序列
0.6	会议	conference	会议的用户权限设置包括参与者、观察者、会议管理者、仲裁者，他们都包含一个role-ref属性，与学习设计中的某一角色关联。若需为会议角色分配多个角色（例如有多个LD角色参与），则需多个会议角色实例，分别对应每个LD角色。这取决于实施时会议如何设置与管理：1.如果会议系统是运行系统的综合组成部分，则最好自动设置启动；2.若会议系统是外部的，可由会议管理器手工设置用户权限。会议管理器要从运行系统获得会议类型、参与用户、用户权限等	M	1	序列

服务						
编号	名称	英文名	解释	先决条件	重数	类型
			信息，以便作相关设置；3. 也可利用该会议要素的数据，生成脚本、配置文件或界面以便进行设置。不管哪种情况，运行系统提供的信息都必须是结构化的。子内容项要素指向会议系统资源，外部会议系统可以是任何通过因特网可访问的类型（资源类型是Web内容），例如：网络会议系统，地方硬件（同步的），第一-班级——first-class，莲公司的便签，新闻组（异步的）。公告对象（announcement object）设置这样的权限：公告的创造者= 参与者。公告的阅读器=观察者。			
0.6.1	会议类型	conference-type	固定选项，规定运行时会议的类型：同步、异步或公告。 可能的值：asynchronous, announcement	M	1	标记
0.6.2	标题	title	同上	O	0..1	字符串
0.6.3	参与者	participant	规定会议的参与者。参与者在会议中能读（听/看）、发信息。该要素影响用户的会议权限设置。会议必须至少有一个角色作为参与者。	M	1..*	空
0.6.3.1	角色引用	role-ref	指向一个角色标识。	M	1	标识符引用
0.6.4	观察者	observer	规定会议的观察者。观察者在会议中只有读的权限，没有写的权限。该要素影响用户的会议权限设置。	O	0..*	空
0.6.4.1	角色引用	role-ref	同上	M	1	标识符引用
0.6.5	会议管理者	conference-manager	会议管理者可创建、删除新的分议题。新会议是现有会议的子会议。会议管理者无权删除现有基点会议，而只能由系统管理员在学习单元结束后删除。会议管理者具有会议观察者、参与者的所有权限。	O	0..1	空
0.6.5.1	角色引用	role-ref	同上	M	1	标识符引用
0.6.6	仲裁者	moderator	规定会议的仲裁者。仲裁者有权对参与者所发布的管理（在其公布之前）。一旦指定了仲裁者角色，也就意味着参与者的消息不能直接发布，而必须经由仲裁者审核。仲裁者可拒绝、改写或接受参与者的消息。不管哪种情况下，发布者都能得到仲裁者对其消息的处理结果。若多个用户具有仲裁者权限，他们具有相同的权限，第一个处理者优先。该要素对用户的会议权限设置有影响。	O	0..1	空
0.6.6.1	角色引用	role-ref	同上	M	1	标识符

服务						
编号	名称	英文名	解释	先决条件	重数	类型
						引用
0.6.7	子内容项	item	结构中的一个节点，指向一个资源。	M	1	序列
0.6.7.1	标识	identifier	同上	O	1	标识符
0.6.7.2	标识符引用	identifierref	指向内容包中的一个资源标识（在学习设计之外）。	O	1	标识符引用
0.6.7.3	可见性	isvisible	同上	O	1	布尔值
0.6.7.4	参数	parameters	同上	O	1	字符串
0.6.7.5	标题	title	同上	O	0..1	字符串
0.6.7.6	子内容项	item	同上	O	0..*	序列
0.6.7.7	元数据	metadata	元数据占位符，包括元数据，使用其名字空间。	O	0..1	序列
0.6.8	元数据	metadata	元数据占位符，包括元数据，使用其名字空间。	O	0..1	序列
0.7	索引搜索	index-search	包含要素编列的索引和/或搜索服务。	M	1	序列
0.7.1	标题	title	同上	O	0..1	字符串
0.7.2	索引	index	形成索引的要素清单，用以搜索服务。索引是后台操作的（对用户不可见性）。可见性的是搜索要素。索引的功能依赖于搜索要素：若是全文搜索，则对资源作索引（即html文本）；若是索引检索，则只建立要素索引，以内容表格的形式呈现。	M	1	选择
0.7.2.1	类索引	index-class	为指定变址特征的一个元素的选择，用于设置一个查询服务。索引是在后台形成的（对用户来说是不可见的）。索引的功能决定于查询元素：-当是一个免费的文本查询时，索引是指向索引的资源形成的（如带有下划线的HTML文本文件）。-当是一个有/没有会议索引的查询时，仅由在同一类中共享的元素组成，包含带有下划线的项。它有一个内容表的形式。	M	1	字符串
0.7.2.2	要素索引	index-element	对要素作索引（每个要素索引只有一个指向reference）。只有当要素具有结构，或采用全文检索时，该索引才有意义。	M	1	空
0.7.2.2.1	索引	index	索引所指向的要素。	M	1	标识符引用
0.7.2.3	要素类型索引	index-type-of-element	索引要素的类型。每个index-type-element每次只有一个要素名。要素名称需符合LD 要素名称，例如： <index-type-of-element>learning-activity</index-type-of-element>	M	1	字符串
0.7.3	搜索	search	该要素规定用户如何访问索引实体，有三种可能性： 1. 用户获得全文检索对话框，可输入文本进行检索（这也意味着需要以全文检索建立索引），检索文本	M	1	空

服务						
编号	名称	英文名	解释	先决条件	重数	类型
			语法由执行确定，例如可以类似Google或Yahoo搜索引擎语法；2. 用户获得一个文本索引（文本表格），可超链到所需资源；3. 用户获得一个文本索引（文本表格），但没有超链，而是提供学习单元的结构信息。			
0.7.3.1	搜索类型	search-type	固定选项，指定期望的运行时搜索类型：全文检索、参照索引、无参照索引。全文检索采用全文检索机制，无参照索引只提供资源列表，无资源页码或超链。参照索引则提供资源页码或超链（根据出版媒体而定）可能的值： <i>free-text-search</i> , <i>index-with-reference</i> , <i>index-without-reference</i>	M	1	标记
0.7.4	元数据	metadata	元数据占位符，包括元数据，使用其名字空间	O	0..1	序列

6.1.12 方法信息表

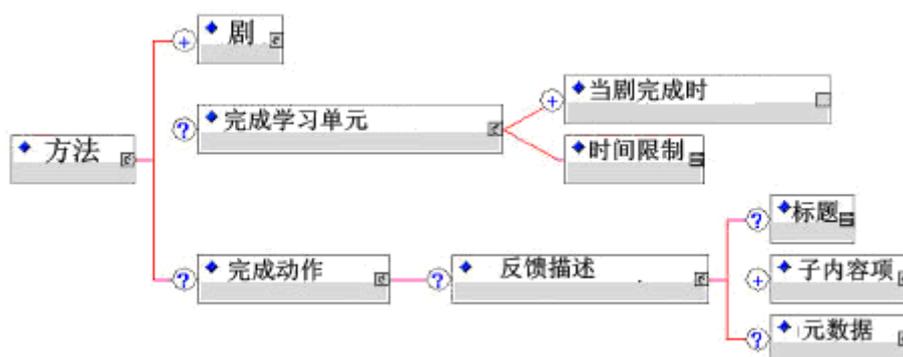


图 15 方法信息图

表12 方法信息表

方法						
编号	名称	英文名	解释	先决条件	重数	类型
0	方法	method	方法包含一系列组成动态学习过程的要素，由一个或多个剧（可看成学习单元的运行脚本）及一个学习单元完成通告组成。	-	-	序列
0.1	剧	play	是解释学习设计的根要素，描绘学习过程中活动的流向（“学习流”）。一个剧由一系列幕组成，每一幕又由一系列角色出演。一个学习设计（每个学习单元）至少有一个剧。在运行时刻，剧呈	M	1..*	序列

方法						
编号	名称	英文名	解释	先决条件	重数	类型
			现为活动、（其他的）学习单元、环境与资源。如果同时有不止一个剧，他们同时发生并相互影响。同一用户在同一用户界面可同时观看不止一个剧的演出。实践经验表明，很多设计同时使用多个剧来表示每个角色的活动流，例如，学习者的剧和教师的剧。但是，只有活动是相互依赖的情况下这（多剧）才有效。			
0.1.1	标识	identifier	学习设计中文件的唯一标识符(ID)。	O	1	标识符
0.1.2	可见性	isvisible	初始可见性属性。 可能的值: true (缺省), 或false 缺省值: true	O	1	布尔值
0.2	完成学习单元	complete-unit-of-learning	规定学习单元何时结束的一个选项。若此要素未发生, 则完成状态设为“unlimited”。	O	0..1	选择
0.2.1	当剧完成时	when-play-completed	规定当指向的剧结束时学习单元随之结束。指向的剧可以不止一个, 这就意味着所有指向的剧必须全都结束后学习单元才会结束。在运行时刻, 当学习单元结束时, 系统管理员应获得结束信息。	M	1..*	空
0.2.1.1	引用	ref	指向学习设计内部的一个标识。	M	1	标识符引用
0.2.2	时间限制	time-limit	时间限制规定当前学习单元启动以后至活动结束的时间。时间表示格式见期限“duration”格式(信息模式中描述)。从学习单元启动开始计时(见“time-unit-of-learning-started”), 设计者需要注意的是, 为角色、幕和剧设置的时间限制是逻辑时间, 运行时, 剧的时间限制接管对角色、幕的时间设置。在层次B和层次C, 时间限制可在一个属性(属性集合属性, 局部属性类型, 数据类型=字符串, 可被创造者声明)中指定。在此案例中, 创造者可为用户控制此属性设置本属性的控制(设置属性)。当一个属性集合引用被指定, 在元素中的内容被忽略。	M	1	字符串
0.3	完成动作	on-completion	当一个活动、幕、剧或学习单元结束时, 此要素所包含的行为被引发。在A层中仅包含一个要素。此包对B和C层可扩展。	O	0..1	容器
0.3.1	反馈描述	feedback-description	指向一个描述反馈信息的资源(Web内容或Id内容的形式)。完成后此文本为可见性	O	0..1	序列
0.3.1.1	子内容项模型	{itemmodel}	大纲集合	M	1	集合

6.1.13 剧信息表

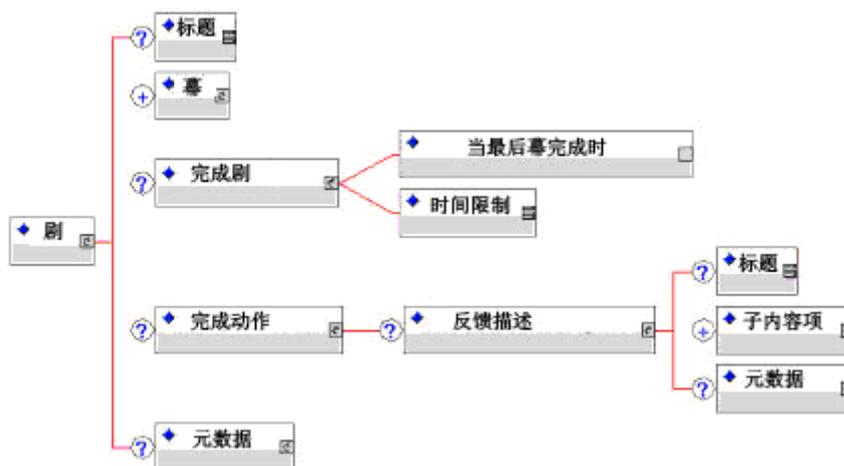


图 16 剧信息图

表13 剧信息表

剧						
编号	名称	英文名	解释	先决条件	重数	类型
0	剧	play	是解释学习设计的根要素，描绘学习过程中活动的流向（“学习流”）。一个剧由一系列幕组成，每一幕又由一系列角色出演。一个学习设计（每个学习单元）至少有一个剧。在运行时刻，剧呈现为活动、（其他的）学习单元、环境与资源。如果同时有不止一个剧，他们同时发生并相互影响。同一用户在同一用户界面可同时观看不止一个剧的演出。实践经验表明，很多设计同时使用多个剧来表示每个角色的活动流，例如，学习者的剧和教员的剧。但是，只有活动是相互依赖的情况下这（多剧）才有效。	-	-	序列
0.1	标识符	identifier	学习设计中文件的唯一标识符(ID)。	O	1	标识符
0.2	可见性	isvisible	初始可见性属性。 可能的值： true（缺省），或false 缺省值： true	O	1	布尔值
0.3	标题	title	适合用户界面呈现的资源名称。	O	0..1	字符串
0.4	幕	act	一个剧由一系列幕组成，每一幕又由一系列角色出演。一个幕表现同时发生的角色分配。一个剧至少有一个幕。若一个剧包含多个幕，则其按顺序呈现。在一个剧的任何时间仅有一个活动幕，活动幕按顺序从第一个开始，后续启动的剧目依次称为活动幕。未启动的非活动幕是不可见性的。条件变量不	M	1..*	序列

剧						
编号	名称	英文名	解释	先决条件	重数	类型
			能控制支配权，也就是说幕比条件具有优先权。不过，完成的幕仍是可见性的及可访问的，但其界面与当前幕具有明显区别，表明完成幕仅作学习历史的参考作用。			
0.4.1	标识	identifier	同上	O	1	标识符
0.5	完成剧	complete-play	规定一个剧何时结束的一个选项。若此要素未发生，则完成状态设为“unlimited”。	O	0..1	选择
0.5.1	当最后幕完成时	when-last-act-completed	规定当最后一个幕结束时剧随之结束。	M	1	空
0.5.2	时间限制	time-limit	时间限制规定当前学习单元启动以后至活动结束的时间。时间表示格式见期限“duration”格式（信息模式中描述）。从学习单元启动开始计时（见“time-unit-of-learning-started”），设计者需要注意的是，为角色、幕和剧设置的时间限制是逻辑时间，运行时，剧的时间限制接管对角色、幕的时间设置。在B和C层，时间限制也可能在属性里设置（property-ref attribute, of type loc-property, datatype=string, to be declared by the author）。在这种情况下，设计者可以控制（设置属性）是否可由用户控制属性的值。当对property-ref作了规定后，属性值接管此项内容。	M	1	字符串
0.6	完成动作	on-completion	当一个活动、幕、剧或学习单元结束时，此要素所包含的行为被引发。在A层中仅包含一个要素。此包对B和C层可扩展。	O	0..1	容器
0.6.1	反馈描述	feedback-description	指向一个描述反馈信息的资源（Web内容或ld内容的形式）。完成后此文本为可见性。	O	0..1	序列
0.6.1.1	子内容项模型	{itemmodel}	一组文档规范。	M	1	集合
0.7	元数据	metadata	元数据占位符，包括元数据，使用其名字空间。	O	0..1	序列

6.1.14 幕信息表

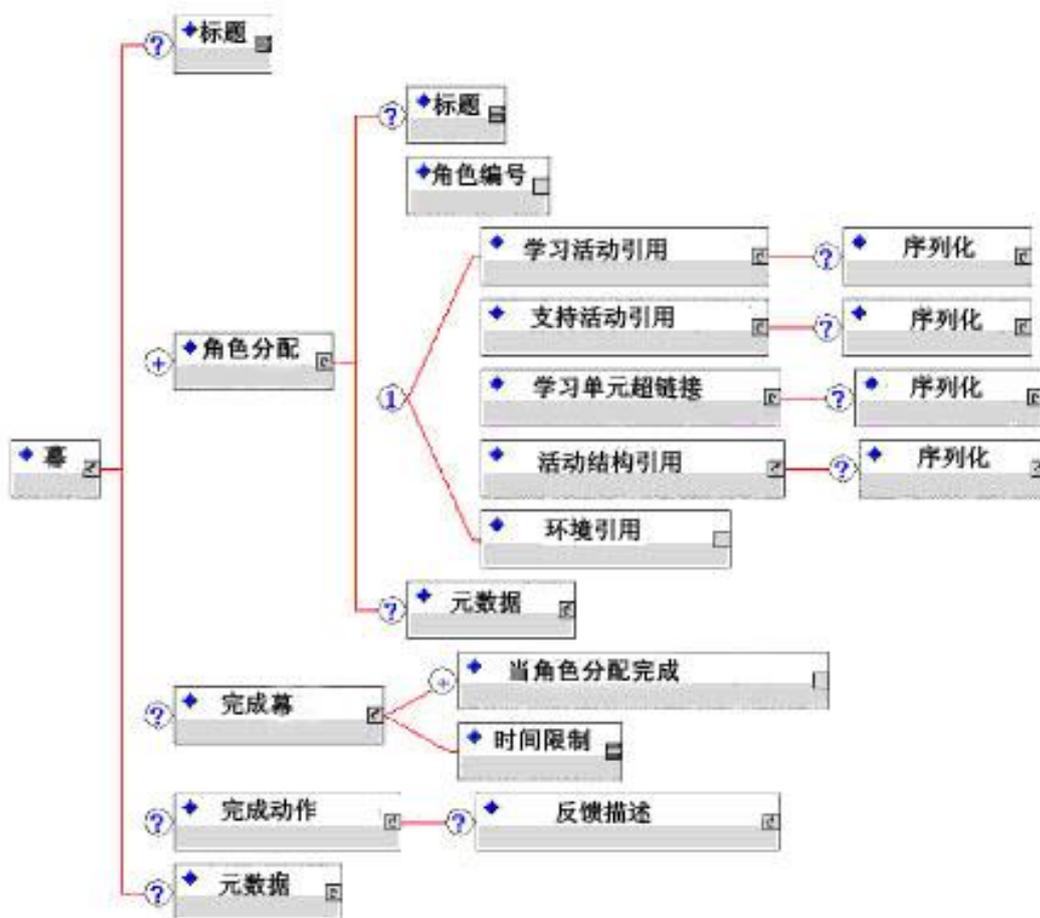


图 17 幕信息图

表14 幕信息表

幕						
编号	名称	英文名	解释	先决条件	重数	类型
0	幕	act	一个剧由一系列幕组成，每一幕又由一系列角色出演。一个幕表现同时发生的角色分配。一个剧至少一个幕。若一个剧包含多个幕，则其按顺序呈现。在一个剧的任何时间仅有一个活动幕，活动幕按顺序从第一个开始，后续启动的剧目依次称为活动幕。未启动的非活动幕是不可见性的。条件变量不能控制支配权，也就是说幕比条件具有优先权。不过，完成的幕仍是可见性的及可访问的，但其界面与当前幕具有明显区别，表明完成幕仅作学习历史的参考作用。	-	-	序列

幕						
编号	名称	英文名	解释	先决条件	重数	类型
0.1	标识	identifier	学习设计中文件的唯一标识符(ID)。	O	1	标识符
0.2	标题	title	适合用户界面呈现的资源名称。	O	0..1	字符串
0.3	角色分配	role-part	一个剧包含一系列的幕, 而一个幕又包含一系列的角色扮演。一个角色分配将角色与活动精确地进行一一对应(包含另外一些学习单元和活动结构的执行)。包含在一个扮演中的角色分配是并行发生的。当一个活动或项的可见性属性设置为“假”, 当角色分配将活动设置为角色时, 在活动树中的链接可设置为可见的, 但内容是不可存取的。	M	1..*	序列
0.3.1	标识	identifier	同上	O	1	标识符
0.3.2	标题	title	同上	O	0..1	字符串
0.3.3	角色引用	role-ref	指向一个角色标识, 该要素可用作表达式的操作数。	M	1	空
0.3.3.1	引用	ref	指向学习设计内部的一个标识。	M	1	标识符引用
0.3.4	选择		Choice	M	1	选择
0.3.4.1	学习活动引用	learning-activity-ref	指向一个学习活动。该要素可用作表达式或算式的操作数。	M	1	空
0.3.4.1.1	引用	ref	同上	M	1	标识符引用
0.3.4.2	支持活动引用	support-activity-ref	指向一个支持活动。该要素可用作表达式或算式的操作数。	M	1	空
0.3.4.2.1	引用	ref	同上	M	1	标识符引用
0.3.4.3	学习单元超链接	unit-of-learning-href	该要素可用作表达式或算式的操作数。该元素用来指向一个外部学习单元的要素。该指向可以是同一个包内的(使用相对URI), 或者包外的(使用绝对URI)。需要将文件ID添加到链接。其用法与标识符引用在XML文件中的引用相同, 指向一个活动结构、学习活动、支持活动或环境因素的ID。注释: 这等价于一个简单的或“空名”X指针(XPointer), 它有个形式: URI#ID和XML等价于一个HTML片段标识符。在XML模式中此形式通过URI结构支持。	M	1	空
0.3.4.3.1	超链接	href	指向一个URI	M	1	任何URI

幕						
编号	名称	英文名	解释	先决条件	重数	类型
0.3.4.4	活动结构引用	activity-structure-ref	指向一个活动结构。	M	1	空
0.3.4.4.1	引用	ref	同上	M	1	标识符引用
0.3.4.5	环境引用	environment-ref	指向当前包所在环境。	M	1	空
0.3.4.5.1	引用	ref	同上	M	1	标识符引用
0.3.5	元数据	metadata	元数据占位符, 包括元数据, 使用其名字空间。	O	0..1	序列
0.4	完成幕	complete-act	规定一个幕何时结束的一个选项。若此要素未发生, 则完成状态设为“unlimited”。	O	0..1	选择
0.4.1	当角色分配完成	when-role-part-completed	规定当指向的角色分配结束时幕随之结束。指向的角色分配可以不止一个, 这就意味着所有指向的角色分配必须全都结束后学习单元才会结束。注意: 所有角色分配引用必须在当前幕中指定。	M	1..*	空
0.4.1.1	引用	ref	同上	M	1	标识符引用
0.4.2	时间限制	time-limit	时间限制规定当前学习单元启动以后至活动结束的时间。时间表示格式见期限“duration”格式(信息模式中描述)。从学习单元启动开始计时(见“time-unit-of-learning-started”), 设计者需要注意的是, 为角色、幕和剧设置的时间限制是逻辑时间, 运行时, 剧的时间限制接管对角色、幕的时间设置。在B和C层, 时间限制也可能在属性里设置。在这种情况下, 设计者可以控制(设置属性)是否可由用户控制属性的值。当对property-ref作了规定后, 属性值接管此项内容。	M	1	字符串
0.5	完成动作	on-completion	当一个活动、幕、剧或学习单元结束时, 此要素所包含的行为被引发。在A层中仅包含一个要素。此包对B和C层可扩展。	O	0..1	容器
0.5.1	反馈描述	feedback-description	指向一个描述反馈信息的资源(Web内容或Id内容的形式)。完成后此文本为可见性。	O	0..1	序列
0.5.1.1	子内容项模型	{itemmodel}	大纲集合	M	1	集合
0.6	元数据	metadata	元数据占位符, 包括元数据, 使用其名字空间。	O	0..1	序列

6.1.15 学习单元表单文件的标准名称

如任何的内容包装，学习单元的内容清单文件有一个预定义的名称和位置。命名为“manifest.xml”的文件放置在互相交换文件包或任何其他图像包（如CD-ROM）的根部。所有的名称都用小写字母表示。它可能使运行时刻系统知道什么规范可以在内容包装（QTI, LD, SS等）中展现。现在，还没有剖面机制（profiling mechanism）展现用来在读取文件前通知解析器或运行系统必须用到什么工具。可能在未来的规范中展现。

6.1.16 学习设计元素的命名空间

学习设计模式元素可用到的命名空间是 http://www.imsglobal.org/xsd/imslid_v1_p0

6.2 B层信息模型

B层信息模型添加的元素，极大扩展了信息设计的能力，使之能够控制学习单元中的学习流程，主要添加的元素是：

- 属性
- 条件判断

其影响表现在以下几个方面：

- 1) 属性元素引入了组件模型，属性需在组件中声明。
- 2) 引入完成活动（complete-activity）、完成幕（complete-act）、完成剧（complete-play）、完成学习单元（complete-unit-of-learning）模型，以满足当属性值设定时（*when-property-value-is-set*）元素。
- 3) 引入完成动作模型，以满足改变属性值（*change-property-value*）元素。
- 4) 引入服务模型，以满足监控元素。
- 5) 引入电子邮件数据（email-data）模型及其二属性（电子邮件属性引用 *email-property-ref* 与用户名属性引用 *username-property-ref*），以引用数据的全局属性。
- 6) 引入时间限制模型的扩展属性即属性引用（*property-ref*），以指称数据的属性。
- 7) 引入方法模型，以满足条件判断元素。
- 8) 引入完成幕模型，以满足当条件判断为真元素。
- 9) 引入一组独立的全局元素，以便读取、设置一切XML文档内容大纲（如XHTML）中的属性。
- 10) 应用由W3C全局属性class组成，以便显示或隐藏一切XML文档内容大纲（如XHTML）中内容元素之条件。

6.2.1 概念模型

B层信息模型的UML概念模型如图6.2，灰显部分是相对于A层信息模型所添加的部分。

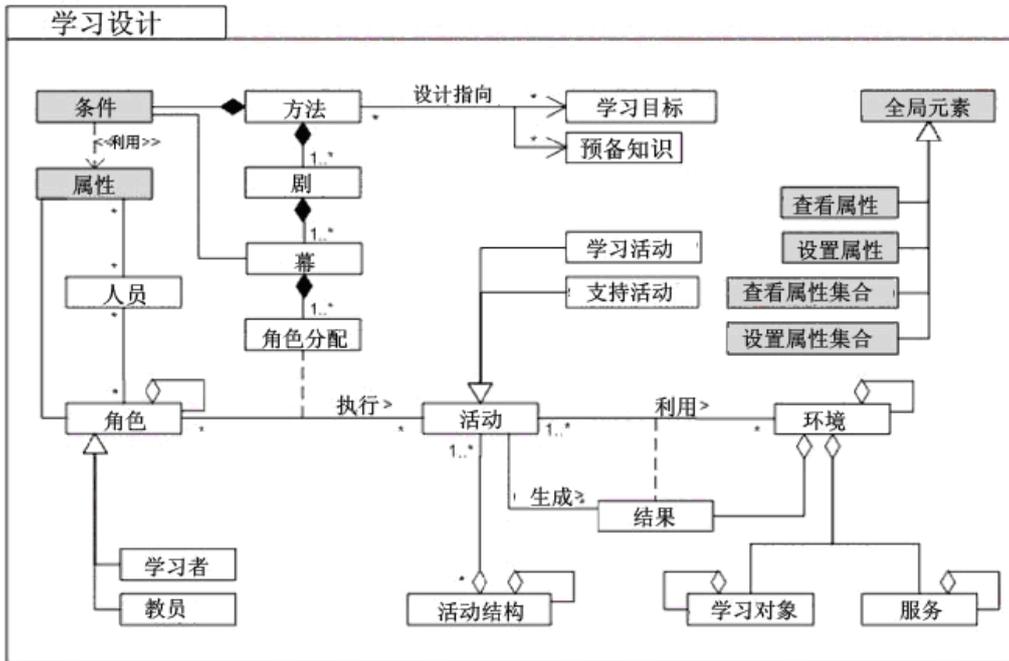


图 18 B 层信息模型的概念模型

运用执行时系统或用户代理来记录用户与角色的属性值与属性定义，保存在所谓的“档案”（dossier）中。在信息设计、组件、属性中定义或声明属性（针对已定义的全局属性），并且由属性运算元素（如查看属性、设置属性、条件判断、改变属性值等）运算。

有以下几种属性：

1) 局部属性(元素名称：*loc-property*)，其存储的是学习单元本地执行的数据，在学习单元中定义使用。这个属性的值对各个用户都一样，但出现在不同的执行时。

2) 全局属性(元素名称：*glob-property*)，要在学习单元之外才能访问（比如其他学习单元），其定义在某学习单元中但是在其他学习单元使用。在LD中可以定义全局属性，执行时控制所定义的全局属性URI是否存在。全局属性一旦定义就永远不能改变其定义，因此倘若属性已经存在，就忽略其定义。

3) 个人属性(元素名称：*locpers-property*、*globpers-property*)，由个人拥有（局部或全局），用于实现个性化。比如，统合若干学习单元的学档，可以通过*globpers*属性得到其模型。个人属性存储于个人可移动的“档案”中。

4) 局部角色属性(元素名称：*locrole-property*)，由角色拥有，永远是局部的。每个有相应角色的用户都可以访问此属性，在同一个学习单元执行时有相同的值。用户代理对属性的存在，要求是安全的，并且有最大操作数目(由系统实现者细化)。

6.2.1.1 全局属性

全局属性是为了管理持久存储。这种控制持久存储的组织机制有效决定了全局属性的范围，允许或拒绝对存储区的访问。一般说来，执行时系统能够访问持久存储，由于可能有许多执行时系统访问相同的存储区，全局属性的作用域扩展到所有执行时系统。因此，有必要区分全局个人属性与普通全局属性。

普通全局属性一般受提供学习的组织控制，即学习提供者决定其作用域，将来某一天，有可能实现世界范围内对学习进度文件的访问，由此来维护学习活动时产生的数据，此时全局个人实现的作用域就真的是全球性（global）的了，假如系统执行时学习者并发使用，同时访问同一个持久单元数据。

比如，某人是个职员，正在参加工作训练课程，其个人时间注册为一个全球性大学的业余远程学习者，此时架构、安全、所属权、控制权等都相应构造出来，并且达成共识。这是LIP规范实现终生学习时所面临的一个大问题。

由此，在近期乃至中期的未来，学习者信息还是由各个机构单独维护（虽然对终生学习会有问题）。所以，就目前而言，学习提供者就决定了全局个人属性的作用域。其他一些获得广泛认同的大问题是全局学习者的命名、类型、词汇表等，以便属性跨系统应用。

6.2.2 属性信息表

组件元素模型中添加的是属性元素。

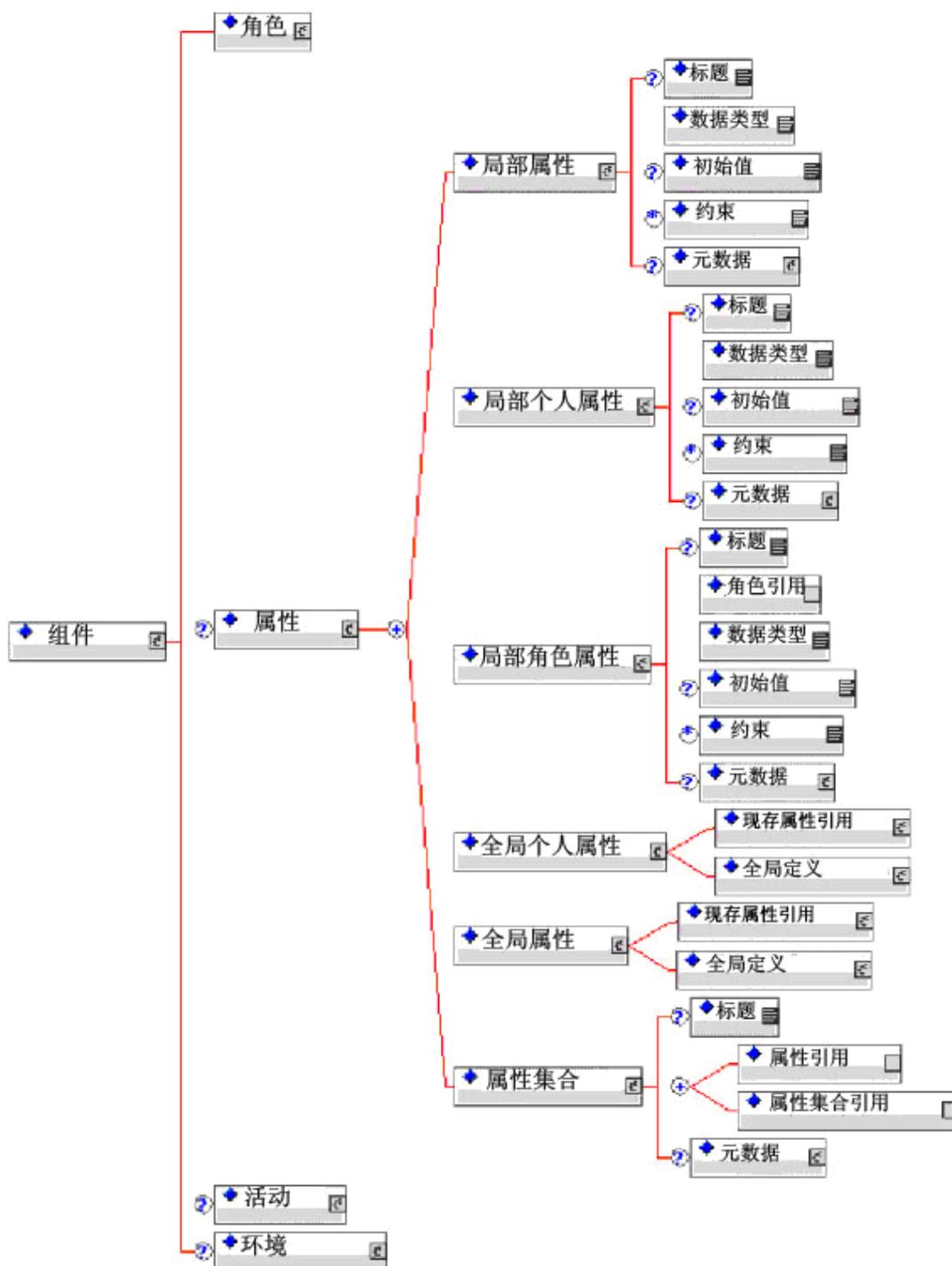


图 19 属性信息图

表15 属性信息表

属性元素						
编号	名称	英文名	解释	先决条件	重数	类型
0	属性(*)	properties(*)	定义新的或声明已存在属性，学习设计所引用的所有属性都必须在此处声明，包括对定义在别处的全局属	—	—	选择

属性元素						
编号	名称	英文名	解释	先决条件	重数	类型
			性的引用。在学习设计中未声明、未引用的全局属性仍然能够被1内容类型的资源中的全局元素所设置、查看。所有元素都能够施行属性运算（属性引用、查看属性、查看属性组等）。			
0.1	局部属性(*)	loc-property(*)	局部属性，别名执行时属性。此属性在执行时各个用户都拥有相同的值，此属性的拥有者是学习单元的执行者，标识符可用于引用此学习单元包中的属性，此标识符的值也能被施行属性运算。	M	1	序列
0.1.1	标识	identifier	某标识符在此学习设计文件(ID)中是唯一的。	M	1	标识符
0.1.2	标题	title	资源的简短名称，用于处理用户代理。	O	0..1	字符串
0.1.3	数据类型(*)	datatype(*)	属性值包含预定义的固定值集，属性数据类型也是预定义的，扩展定义时，使用其他值指定这一元素的内容。	M	1	字符串
0.1.3.1	数据类型	datatype	供选择的固定的数据类型集合，包括以下数据类型：boolean, integer, real, string, datetime, duration, text, file, uri and other. 它们的数据格式在该信息模型中另外的部分描述，可能包括有：string, boolean, integer, uri, datetime, file, real, text, duration等。	M	1	标记
0.1.4	初始值(*)	initial-value(*)	属性的初始值是引用此元素时设定的，此值未设定时初始值为“<no value>”。	O	0..1	字符串
0.1.5	约束(*)	restriction(*)	若干不同类型限定于属性值上，意味着属性值的有效性取决于所指定的数据类型与值是否符合指定的限制规则。若干限定能设定的属性是“restriction-type”(其形式与W3C XML schema 1.0中的规定相同)。属性不包含数据的阵列(列表)，只能包含单个值，由此只能限定于单个值(即“whitespace”限定在LD中不支持。信息模型的其他地方指定所限定的类型)。	O	0..*	字符串
0.1.5.1	约束类型	restriction-type	所支持的限定集合为：minExclusive, minInclusive, maxExclusive, maxInclusive, totalDigits, fractionDigits, length, minLength, maxLength, enumeration, whiteSpace, pattern (解释参见信息模型其他地方)。可能取值:minExclusive, minInclusive, maxExclusive, maxInclusive, totalDigits, fractionDigits, length, minLength, maxLength, enumeration, whiteSpace, pattern.	O	1	标记
0.1.6	元数据	metadata	元数据标记，应用命名空间，标识元数据	O	0..1	序列
0.2	局部个人属性(*)	locpers-property(*)	局部个人属性，在学习单元的一个执行时，其对于每个角色的各个用户的值都不同，即为每个用户指定一	M	1	序列

属性元素						
编号	名称	英文名	解释	先决条件	重数	类型
			个值。此标识符用于引用本学习单元包内的属性，对其中的值也能作属性运算。			
0.2.1	标识	identifier	同上	M	1	标识符
0.2.2	标题	title	同上	O	0..1	字符串
0.2.3	数据类型(*)	datatype (*)	同上	M	1	字符串
0.2.3.1	数据类型	datatype	同上	M	1	标记
0.2.4	初始值(*)	initial-value (*)	同上	O	0..1	字符串
0.2.5	约束(*)	restriction (*)	同上	O	0..*	字符串
0.2.5.1	约束类型	restriction-type	同上	O	1	标记
0.2.6	元数据	metadata	元数据标记，应用命名空间，标识元数据。	O	0..1	序列
0.3	局部角色属性(*)	locrole-property (*)	别名group-property，即局部角色属性，此属性对相应角色的各个用户，在一个学习单元的执行时其值相同，此属性的拥有者是学习单元的参与者，此标识符用于引用此学习单元包中的属性，对其中的值也能作属性运算。	M	1	序列
0.3.1	标识	identifier	同上	M	1	标识符
0.3.2	标题	title	同上	O	0..1	字符串
0.3.3	角色引用	role-ref	引用角色资源的标识符，在表达式中可以作为操作数。	M	1	空
0.3.3.1	引用	ref	引用学习设计中的标识符。	M	1	标识符引用
0.3.4	数据类型(*)	datatype (*)	同上	M	1	字符串
0.3.4.1	数据类型	datatype	同上	M	1	标记
0.3.5	初始值(*)	initial-value (*)	同上	O	0..1	字符串
0.3.6	约束(*)	restriction (*)	同上	O	0..*	字符串
0.3.6.1	约束类型	restriction-type	同上	O	1	标记
0.3.7	元数据	metadata	元数据标记，应用命名空间，标识元数据	O	0..1	序列
0.4	全局个人属性(*)	globpers-property(*)	全局个人属性，别名学档属性(portfolio-property)，此属性对各个用户有不同值，不依赖于学习单元的不同执行时(指用户的学档)。人员拥有此属性，此标识符用于引用此学习单元包中的属性，对其中的值也能作属性运算。	M	1	选择
0.4.1	标识	identifier	同上	M	1	标识符
0.4.2	现存属性引用(*)	existing (*)	引用已声明的属性(比如其他学习单元或全局档案之属性，参见“global-definition”之情形，倘若创作者定义了实际上已经存在的新的全局属性)。此属性用 href引用，指定其绝对URI。注意，当验证此学习单元时，URI不必一定存在，URI之声明是由一外部学习单元在任何时候声明，因此其只能由创作者控制。	M	1	空

属性元素						
编号	名称	英文名	解释	先决条件	重数	类型
0.4.2.1	Href	href	引用URI.	M	1	任何URI
0.4.3	全局定义(*)	<i>global-definition</i> (*)	全局定义用于声明、定义全局属性。全局定义只能被定义一次，在学习设计的上下文中的再次声明也无法改变其值（只存在于存储属性的数据库中）。全局属性最好由外部机制定义。为了保证一致性，应遵循以下规则：一旦定义了全局属性，无论在何处，不允许改变！在学习单元的重新发布时也是如此。定义之使用只能是此URI (href) 尚未存在，倘若存在也被忽略。此URI必须是能够唯一标识全局属性的标识符，并且必须是绝对URI，倘若URI就是URL，此URI不必指向属性的位置，只是能够解释为标识符。	M	1	序列
0.4.3.1	Uri	uri	指定URI.	M	1	任何URI
0.4.3.2	标题	title	同上	O	0..1	字符串
0.4.3.3	数据类型(*)	<i>datatype</i> (*)	同上	M	1	字符串
0.4.3.3.1	数据类型	<i>datatype</i>	同上	M	1	标记
0.4.3.4	初始值(*)	<i>initial-value</i> (*)	同上	O	0..1	字符串
0.4.3.5	约束(*)	<i>restriction</i> (*)	同上	O	0..*	字符串
0.4.3.5.1	约束类型	<i>restriction-type</i>	同上	O	1	标记
0.4.3.6	元数据	<i>metadata</i>	元数据标记，应用命名空间，标识元数据。	O	0..1	序列
0.5	全局属性(*)	<i>glob-property</i> (*)	全局属性是全局唯一的属性，其存储的值独立于用户、学习单元与角色。此标识符用于引用此学习单元包中的属性，对其中的值也能作属性运算。	M	1	选择
0.5.1	标识	<i>identifier</i>	同上	M	1	标识符
0.5.2	现存属性引用(*)	<i>existing</i> (*)	同上	M	1	空
0.5.2.1	Href	<i>href</i>	同上	M	1	任何URI
0.5.3	全局定义(*)	<i>global-definition</i> (*)	同上	M	1	序列
0.6	属性集合(*)	<i>property-group</i> (*)	其定义了属于整体的属性集合(可以在表格中编辑)。其只能包含相同类型的属性，此标识符用于引用此学习单元包中的属性，对其中的值也能作属性运算。	M	1	序列
0.6.1	标识	<i>identifier</i>	同上	M	1	标识符
0.6.2	标题	<i>title</i>	同上	O	0..*	字符串
0.6.3	选择		选择	M	1..*	选择
0.6.3.1	属性引用(*)	<i>property-ref</i> (*)	引用属性，可以引用任何类型的属性，比如：local property, globalproperty, local personal property, local role property, global personal property, localrole property。引用属性指向学习	M	1	空

属性元素						
编号	名称	英文名	解释	先决条件	重数	类型
			设计中的属性声明，此元素可以作为计算或表达式中的操作数。			
0.6.3.1.1	引用	ref	同上	M	1	标识符引用
0.6.3.2	属性集合引用(*)	property-group-ref(*)	引用属性组	M	1	空
0.6.3.2.1	引用	ref	同上	M	1	标识符引用
0.6.4	元数据	metadata	元数据标记，应用命名空间，标识元数据	O	0..1	序列

6.2.3 当属性值设定时信息表

元素“当属性值设定时”添加于A层学习设计内容模型如下元素中，

- 完成活动
- 完成幕
- 完成剧
- 完成学习单元

在以上四种情形下，此元素作为最后一个元素添加进来。

比如如下的完成活动模型：

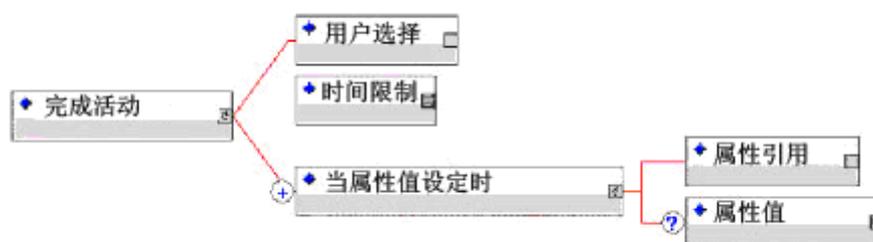


图20 完成活动模型图

“当属性值设定时”模型如下：

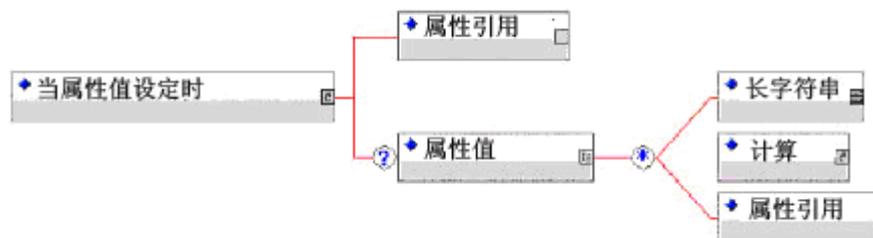


图21 “当属性值设定时”模型图

表16 “当属性值设定时”信息表

当属性值设定时						
编号	名称	英文名	解释	先决条件	重数	类型
0	当属性值设定时(*)	when-property-value-is-set (*)	简单表达式, 包含二个子元素, 属性引用及其属性值(可选)。其取真值条件为: 1) 设置属性为一指定值; 2) 属性非空, 属性值忽略。	—	—	序列
0.1	属性引用(*)	property-ref (*)	引用属性, 可以引用任何属性: local property, global property, local personal property, local role property, global personal property, local role property. 属性引用指向学习设计中的属性声明, 此元素可以作为计算或表达式中的操作数。	M	1	空
0.1.1	引用	ref	引用学习设计中任意标识符。	M	1	标识符引用
0.2	属性值(*)	property-value (*)	此元素可以作为计算或表达式中的操作数, 其为属性指定值或比较值。具体情况取决于上下文。比如, 在if语句中, 属性是比较值, 在改变属性值中, 属性被设置此值。依据属性类型, 此值是 PCDATA 或 langstring 类型。属性值可以由其他属性值计算, 也可能是直接占有其他属性的属性值(如 property-ref)。	O	0..1	字符串
0.2.1	长字符串	langstring	其等同于XHTML <p>元素, 绑定来自元数据。依据W3C规范, 属性attribute xml:lang可以被添加至所有元素, 是此元素的特别需要。	M	1	字符串
0.2.2	计算(*)	calculate (*)	此容器是元素执行计算所需, 也用于表达式中。	M	1	选择
0.2.2.1	表达式(*)	{expression} (*)	大纲集合	M	1	集合
0.2.3	属性引用(*)	property-ref (*)	同上	M	1	空
0.2.3.1	引用	ref	同上	M	1	标识符引用

6.2.4 改变属性值信息表

改变属性值元素添加于A层内容模型的元素on-completion(完成动作)中, 亦添加于B层内容模型的元素then之中。

对A层设计的扩展:

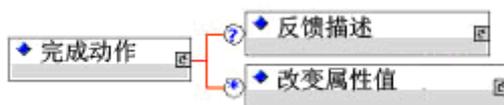


图22 A层设计的扩展图

改变属性值模型如下：

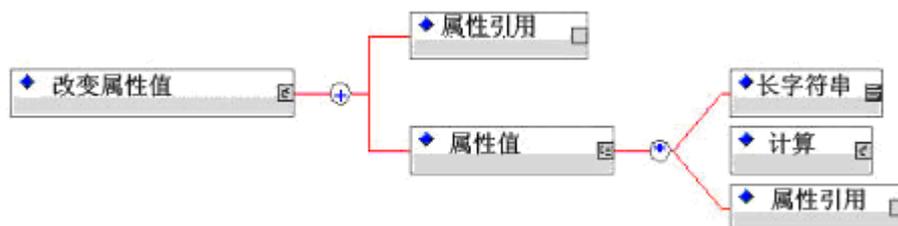


图23 “改变属性值”模型图

表17 “改变属性值模型”信息表

改变属性值						
编号	名称	英文名	解释	先决条件	重数	类型
0	改变属性值(*)	change-property-value (*)	此元素用于改变属性值，由事件触发（比如完成什么任务），如完成活动时，要改变属性值以反映此事实。档案中记载完成活动时自动保存的记录。因此不必要记录此完成事件，可用于记录（或修改）其他问题。	—	—	序列
0.1	属性引用(*)	property-ref (*)	引用属性，可以引用任何属性：local property, global property, local personal property, local role property, global personal property, local role property. 属性引用指向学习设计中的属性声明，此元素可以作为计算或表达式中的操作数。	M	1	空
0.1.1	引用	ref	引用学习设计中任意标识符。	M	1	标识符引用
0.2	属性值(*)	property-value (*)	此元素可以作为计算或表达式中的操作数，其为属性指定值或比较值。具体情况取决于上下文。比如，在if语句中，属性是比较值，在改变属性值中，属性被设置此值。依据属性类型，此值是 PCDATA 或 langstring类型。属性值可以由其他属性值计算，也可能是直接占有其他属性的属性值（如 property-ref）。	M	1	字符串
0.2.1	长字符串	langstring	其等同于XHTML <p>元素，绑定来自元数据。依据W3C规范，属性attribute xml:lang可以被添加至所有元素，是此元素的特别需要。	M	1	字符串

0.2.2	计算(*)	calculate (*)	此容器是元素执行计算所需，也用于表达式中。	M	1	选择
0.2.2.1	表达式(*)	expression} (*)	大纲集合	M	1	集合
0.2.3	属性引用 (*)	property-ref (*)	同上	M	1	空
0.2.3.1	引用	ref	同上	M	1	标识符 引用

6.2.5 监控信息表

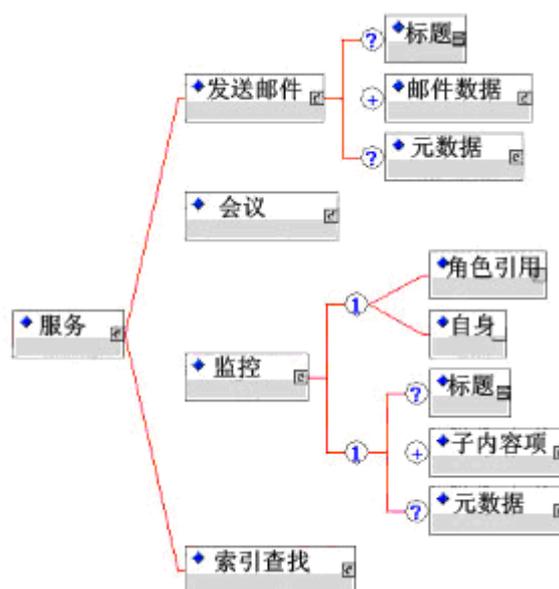


图24 监控信息图

表18 监控信息表

监控						
编号	名称	英文名	解释	先决条件	重数	类型
0	监控(*)	monitor (*)	监控服务便于用户查看自身或他人的属性的结构化显示，其使用1d内容类型的资源中的全局属性，来查看自身或计算中所有用户的属性。	—	—	序列
0.1	选择		选择	M	1	选择
0.1.1	角色引用	role-ref	引用角色资源的标识符，可作为表达式中的操作数。	M	1	空
0.1.1.1	引用	ref	引用学习设计中的标识符。	M	1	标识符 引用
0.1.2	自身(*)	self (*)	引用人员自身的属性，不是其角色的属性。	M	1	空
0.2	子内容项 模型	{itemmodel}	大纲集合	M	1	集合

6.2.6 电子邮件数据扩展

元素电子邮件数据 (*email-data*) 在B层设计中添加二个属性，在A层设计无属性。

表19 电子邮件数据扩展表

编号	名称	英文名	解释	先决条件	重数	类型
.1	电子邮件属性引用	<i>email-property-ref</i>	此属性引用了用户收到的email地址属性。	M	1	任何URI
.2	用户名属性引用	<i>username-property-ref</i>	此属性引用了用户收到的用户名属性。	O	1	任何URI

6.2.7 时间限制扩展

元素时间限制 (*time-limit*) 在B层设计中添加一个属性，在A层设计无属性。

表20 时间限制扩展表

编号	名称	英文名	解释	先决条件	重数	类型
.1	属性引用	<i>property-ref</i>	引用属性标识符，在B、C层模型中，属性(属性引用属性，属于局部属性，类型为string，由用户声明)中还要指定时间限制。此时用户控制该属性，以便用户控制属性值，指定属性时，元素的内容将被忽略。	O	1	标识符引用

6.2.8 条件信息表

元素条件添加于A层模型的方法元素的模型中。

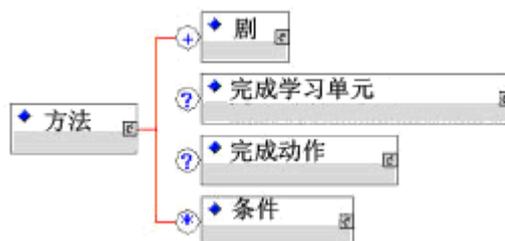


图25 方法元素图

条件模型如下：

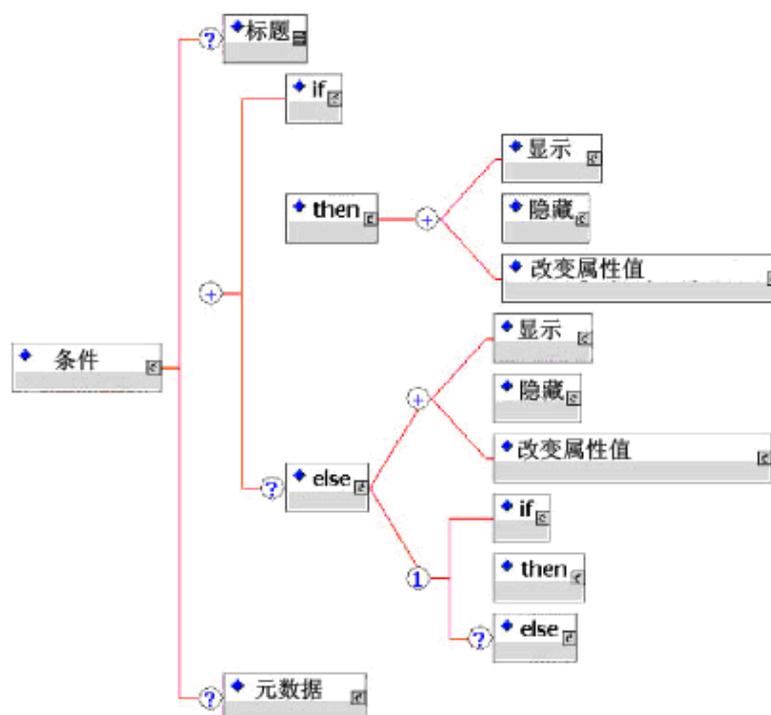


图26 条件模型图

表21 条件信息表

条件						
编号	名称	英文名	解释	先决条件	重数	类型
0	条件(*)	conditions (*)	所有条件是预定义的，必须求值，每当学习设计开启新的会话或者改变了属性值，并且只适合以下属性： a) 用户访问学习单元中的属性 b) 学习单元表达式中要计算的属性，此属性在表达式中但不能自动计算（比如 time-unit-of-learning-started），事件启动的条件是条件的真值，事件包括显示、隐藏、改变属性值（level B）、通知某角色（level C）。显示、隐藏事件设置对象的可见属性，被设置的对象有活动、环境、姓名、剧、活动结构、学习单元、各种类的对象。	—	—	序列
0.1	标题	title	资源的简称，便于处理用户代理。	O	0..1	字符串
0.2	序列		序列	M	1..*	序列
0.2.1	if(*)	if(*)	if的表达式模式，其值为真假，可用各种表达式模式，但最好用LD提供的表达式模式，倘若使用独立命名空间，则可以扩展表达式元素（比如随机数函数）。表达式为真，则启动“then”规则，表达式为假，则启动“else”规则，否则什么也不做。	M	1	选择

0.2.1.1	表达式(*)	{expression} (*)	大纲集合	M	1	集合
0.2.2	then (*)	then (*)	if中表达式为真, 执行then子句。	M	1	选择
0.2.2.1	then模型(*)	{thenmodel} (*)	大纲集合	M	1	集合
0.2.3	else (*)	else (*)	if中表达式为假, 执行该子句。	O	0..1	选择
0.2.3.1	then模型(*)	{thenmodel} (*)	同上	M	1	集合
0.2.3.2	序列		序列	M	1	序列
0.2.3.2.1	if (*)	if (*)	同上	M	1	选择
0.2.3.2.1.1	表达式(*)	{expression} (*)	同上	M	1	集合
0.2.3.2.2	then (*)	then (*)	同上	M	1	选择
0.2.3.2.2.1	then模型(*)	{thenmodel} (*)	同上	M	1	集合
0.2.3.2.3	else (*)	else (*)	同上	O	0..1	选择
0.3	元数据	metadata	元数据标记, 应用命名空间, 标识元数据	O	0..1	序列

6.2.9 THEN 模块信息表

参见上图, then模型 {thenmodel} 是一个大纲集合。

表22 THEN模块信息表

then模型						
编号	名称	英文名	解释	先决条件	重数	类型
0.1	显示(*)	show (*)	选择元素, 指定条件为真时显示哪个元素, 其影响相应实体的可见性。	M	1	选择
0.1.1	显示—隐藏	{show-hide}	大纲集合	M	1	集合
0.2	隐藏(*)	hide (*)	选择元素, 指定条件为真时隐藏哪个元素, 其影响相应实体的可见性。	M	1	选择
0.2.1	显示—隐藏	{show-hide}	同上	M	1	集合
0.3	改变属性值(*)	change-property-value (*)	此元素用于改变属性值, 由事件触发(比如完成什么任务), 如, 完成活动时, 要改变属性值以反映此事实。档案中记载完成活动时自动保存的记录。因此不必要记录此完成事件, 可用于记录(或修改)其他问题。	M	1	序列

6.2.10 IF 模块信息表

if模块有二组实体, 即表达式与计算, 表达式实体还可以作为元素存在于元素 *when-condition-true* 与 *users-in-role* 中(参见下文)。计算实体也可以作为元素存在于元素 *property-value* 中(参见 *change-property-value*)。为避免相同内容重复若干次, 该模型只述于此处。

If						
编号	名称	英文名	解释	先决条件	重数	类型
			间, 则可以扩展表达式元素(比如随机数函数)。表达式为真, 则启动“then”规则, 表达式为假, 则启动“else”规则, 否则什么也不做。			
0.1	表达式(*)	{expression} (*)	大纲集合	M	1	集合

6.2.11 表达式模块信息表

参见上图, 表达式{expression}是一个大纲集合。

表24 表达式模块信息表

表达式						
编号	名称	英文名	解释	先决条件	重数	类型
0.1	是角色成员(*)	is-member-of-role(*)	倘若此人是“ref”所引用的角色成员则为真。	M	1	空
0.1.1	引用	ref	引用学习设计中的标识符。	M	1	标识符引用
0.2	是(*)	is (*)	固定局部运算符, 表达式为真时则为真。	M	1	序列
0.2.1	计算(*)	{calculate} (*)	大纲集合	M	1	集合
0.3	不是(*)	is-not (*)	固定局部运算符, 表达式为假时则为真。	M	1	序列
0.3.1	计算(*)	{calculate} (*)	同上	M	1	集合
0.4	并且(*)	And (*)	固定局部运算符, 表达式皆为真时则为真。	M	1	序列
0.4.1	表达式(*)	{expression} (*)	同上	M	1	集合
0.4.2	表达式(*)	{expression} (*)	同上	M	1..*	集合
0.5	或者(*)	or (*)	固定局部运算符, 表达式至少一个为真时则为真。	M	1	序列
0.5.1	表达式(*)	{expression} (*)	同上	M	1	集合
0.5.2	表达式(*)	{expression} (*)	同上	M	1..*	集合
0.6	加(*)	sum (*)	固定局部运算符, 返回相应值的和。	M	1	容器
0.6.1	计算(*)	{calculate} (*)	同上	M	1..*	集合
0.7	减(*)	subtract(*)	固定局部运算符, 返回前值减后值的差。	M	1	序列
0.7.1	计算(*)	{calculate} (*)	同上	M	1	集合
0.8	乘(*)	multiply (*)	固定局部运算符, 返回相应值的积。	M	1	序列
0.8.1	计算(*)	{calculate} (*)	同上	M	1	集合
0.9	除(*)	divide (*)	固定局部运算符, 返回前值除以后值的商。	M	1	序列
0.9.1	计算(*)	{calculate} (*)	同上	M	1	集合
0.10	大于(*)	greater-than (*)	固定局部运算符, 前值比后值大时则为真。	M	1	序列
0.10.1	计算(*)	{calculate} (*)	同上	M	1	集合
0.11	小于(*)	less-than (*)	固定局部运算符, 前值比后值小时则为真。	M	1	序列

表达式						
编号	名称	英文名	解释	先决条件	重数	类型
0.11.1	计算(*)	{calculate} (*)	同上	M	1	集合
0.12	用户角色(*)	users-in-role (*)	包含系列元素，此表达式将所有人员指定某角色（不指向角色本身）。	M	1	序列
0.12.1	角色引用	role-ref	引用角色资源的标识符，可作为表达式中操作数。	M	1	空
0.12.1.1	引用	ref	同上	M	1	标识符引用
0.12.2	表达式(*)	expression (*)	包含系列表达式元素。	M	1	选择
0.12.2.1	表达式(*)	{expression} (*)	同上	M	1	集合
0.13	无值(*)	no-value (*)	属性为空是则为真。	M	1	容器
0.13.1	属性引用(*)	property-ref (*)	引用属性，可以引用任何属性：local property, global property, local personal property, local role property, global personal property, local role property. 引用属性指向学习设计中的属性声明，此元素可以作为计算或表达式中的操作数。	M	1	空
0.13.1.1	引用	ref	同上	M	1	标识符引用
0.14	学习单元开始时间(*)	time-unit-of-learning-started (*)	引用已经启动的学习单元的执行时间，此为固定时间，存储于学习设计的实例中，为datetime格式（参见datatype）。	M	1	空
0.14.1	学习单元URI	unit-of-learning-uri		M	1	任何URI
0.15	活动开始日期时间(*)	datetime-activity-started (*)	属性首次被用户访问的日期时间，为datetime格式（参见datatype）。	M	1	空
0.15.1	引用	ref	同上	M	1	标识符引用
0.16	当前日期时间(*)	current-datetime (*)	目前日期时间，为datetime格式（参见datatype）。	M	1	空
0.17	完成(*)	complete (*)	相应元素（如活动）已经完成则为真。	M	1	选择
0.17.1	学习活动引用	learning-activity-ref	引用学习活动，此元素可以作为计算或表达式中的操作数。	M	1	空
0.17.1.1	引用	ref	同上	M	1	标识符引用
0.17.2	支持活动引用	support-activity-ref	引用支持活动，此元素可以作为计算或表达式中的操作数。	M	1	空
0.17.2.1	引用	ref	同上	M	1	标识符引用
0.17.3	学习单元超链接	unit-of-learning-href	此元素可以作为计算或表达式中的操作数，用于引用外部学习单元的相应元素，或许在同一包内（因此href是相对URI），或许指向包外的学习单	M	1	空

表达式						
编号	名称	英文名	解释	先决条件	重数	类型
			元（因此href是绝对URI），需要在文件引用时添加分片标识符（#ID），其用法与XML文档内部使用标识符引用的方法一样，指向包含此外部学习单元的活动结构、学习活动、支持活动或环境元素的ID。注意，其等同于简单XPointer，格式是URI#ID，是XML中类似于HTML分片标识符的标记。在XML大纲中，任何URI结构支持该格式。			
0.17.3.1	href	href	引用URI。	M	1	任何URI
0.17.4	活动结构引用	activity-structure-ref	引用活动结构。	M	1	空
0.17.4.1	引用	ref	同上	M	1	标识符引用
0.17.5	角色分配引用(*)	role-part-ref (*)	引用角色分配。	M	1	空
0.17.5.1	引用	ref	同上	M	1	标识符引用
0.17.6	幕引用(*)	act-ref (*)	引用活动序列(如方法、剧、活动序列)。	M	1	空
0.17.6.1	引用	ref	同上	M	1	标识符引用
0.17.7	剧引用(*)	play-ref (*)	引用剧。	M	1	空
0.17.7.1	引用	ref	同上	M	1	标识符引用
0.18	不是(*)	Not (*)	固定逻辑表达式。	M	1	选择
0.18.1	表达式(*)	{expression} (*)	同上	M	1	集合

6.2.12 计算模块信息表

参见上图，计算{calculate}是一个大纲集合。

表25 计算模块信息表

计算						
编号	名称	英文名	解释	先决条件	重数	类型
0.1	操作数(*)	{operand} (*)	大纲集合	M	1	集合
0.2	操作数(*)	{operand} (*)	同上	M	1	集合

6.2.13 操作数模块信息表

参见上图，操作数{operand}是一个大纲集合。

表26 操作数模块信息表

操作数						
编号	名称	英文名	解释	先决条件	重数	类型
0.1	属性引用 (*)	property-ref (*)	引用属性，可以引用任何属性：local property, global property, local personal property, local role property, global personal property, local role property. 引用属性指向学习设计中的属性声明，此元素可以作为计算或表达式中的操作数。	M	1	空
0.1.1	引用	ref	引用学习设计中的标识符。	M	1	标识符引用
0.2	属性值 (*)	property-value (*)	此元素可以作为计算或表达式中的操作数，其为属性指定值或比较值。具体情况取决于上下文。比如，在if语句中，属性是比较值，在改变属性值中，属性被设置此值。依据属性类型，此值是PCDATA 或langstring类型。属性值可以由其他属性值计算，也可能是直接占有其他属性的属性值（如property-ref）。	M	1	字符串
0.2.1	长字符串	langstring	其等同于XHTML <p>元素，绑定来自元数据。依据W3C规范，属性attribute xml:lang可以被添加至所有元素，是此元素的特别需要。	M	1	字符串
0.2.2	计算(*)	calculate (*)	此容器是元素执行计算所需，也用于表达式中	M	1	选择
0.2.2.1	表达式 (*)	{expression} (*)	大纲集合	M	1	集合
0.2.3	属性引用 (*)	property-ref (*)	同上	M	1	空
0.2.3.1	引用	ref	同上	M	1	标识符引用
0.3	表达式 (*)	{expression} (*)	同上	M	1	集合

6.2.14 当条件为真时信息表

该元素添加于“完成幕”信息模型中，元素“当属性值已经设置”也是其扩展元素。

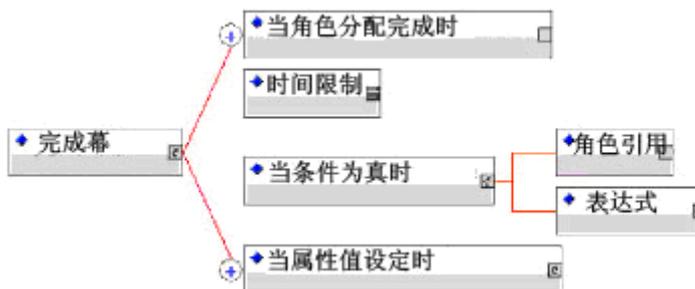


图28 当条件为真时信息图

表27 当条件为真时信息表

当条件为真时						
编号	名称	英文名	解释	先决条件	重数	类型
0	当条件为真时(*)	when-condition-true(*)	条件简单表达式，用于角色引用中所有人，表达式对指定角色所有用户为真时，此条件为真。	—	—	序列
0.1	角色引用	role-ref	引用角色资源标识符，可作为表达式中操作数。	M	1	空
0.1.1	引用	ref	引用学习设计中的标识符。	M	1	标识符引用
0.2	表达式(*)	expression (*)	不同表达式元素之选择。	M	1	选择
0.2.1	表达式(*)	{expression} (*)	大纲集合	M	1	集合

6.2.15 “显示 & 隐藏”信息表

元素“显示”与“隐藏”具有相同的内容模式。



图29 “显示”信息图

表28 “显示”信息表

显示						
编号	名称	英文名	解释	先决条件	重数	类型
0	显示(*)	show (*)	选择元素，指定条件为真时显示相应元素，其影响相应实体的可见性。	—	—	选择
0.1	显示—隐藏	{show-hide}	大纲集合	M	1	集合



图30 “隐藏”信息图

表29 “隐藏”信息表

隐藏						
编号	名称	英文名	解释	先决条件	重数	类型
0	隐藏(*)	hide (*)	选择元素，指定条件为真时隐藏相应元素，其影响相应实体的可见性。	—	—	选择
0.1	显示—隐藏	{show-hide}	大纲集合	M	1	集合

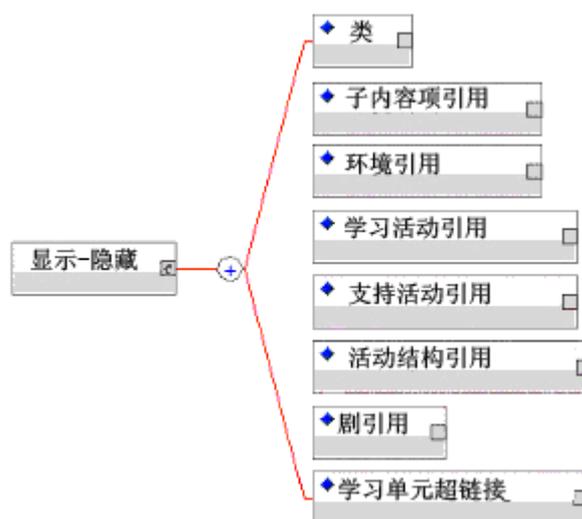


图31 “显示 & 隐藏”信息图

表30 “显示 & 隐藏”信息表

显示—隐藏						
编号	名称	英文名	解释	先决条件	重数	类型
0.1	类(*)	class (*)	指定拥有某类属性值的元素之显示或隐藏，到底如何取决于该元素的上下文。类属性是全局属性，其设置于ldcontent类型的资源，存在于学习设计的环境元素中，注意类可用于类似样式单的功能（如设置可见性）中，同时也有其语义分类目的（如同HTML），这是样式单或其他自动处理功能所缺乏的。此时用于识别通用对象的类，以便立刻处理。	M	1	空
0.1.1	类	class	引用学习设计或内容元素中的类属性的值，包含一个CDATA串，如同HTML，串中可指定若干类，相互之间以空格分开。类的优先级次序与CSS规范一样（参见 http://www.w3.org/style/css ）。原则上，各个元素都有类属性。“Class”是HTML	O	1	字符串

显示—隐藏						
编号	名称	英文名	解释	先决条件	重数	类型
			4.0与XHTML [14]的全局W3C属性，其将一个类名称或类名称集合赋予某元素，同一类名称可赋予任意多元素，类名称之间以空格分开。类元素可用于元素的语义组，且由LD的条件与样式单处理，倘若向WEB客户端发送学习对象及其类属性与值。			
0.1.2	标题	<i>title</i>	当内容折叠（参见“with-control”），必须指定标题，指定时需在类元素的“title”属性中指明。	O	1	字符串
0.1.3	受控	<i>with-control</i>	布尔值，若真隐藏元素，在用户接口中必须提供折叠与展开控制(如同浏览器中的[+]控制符)。由此用户可以自己决定隐藏或显示元素中的内容。	O	1	布尔
0.2	子内容项引用(*)	<i>item-ref</i> (*)	引用设计内容中的标识符。	M	1	空
0.2.1	引用	<i>ref</i>	引用学习设计中的标识符。	M	1	标识符引用
0.3	环境引用	<i>environment-ref</i>	引用包中的某环境。	M	1	空
0.3.1	引用	<i>ref</i>	同上	M	1	标识符引用
0.4	学习活动引用	<i>learning-activity-ref</i>	引用学习活动，用于计算或表达式中的操作数。	M	1	空
0.4.1	引用	<i>ref</i>	同上	M	1	标识符引用
0.5	支持活动引用	<i>support-activity-ref</i>	引用支持活动，用于计算或表达式中的操作数。	M	1	空
0.5.1	引用	<i>ref</i>	同上	M	1	标识符引用
0.6	活动结构引用	<i>activity-structure-ref</i>	引用活动结构。	M	1	空
0.6.1	引用	<i>ref</i>	同上	M	1	标识符引用
0.7	剧引用(*)	<i>play-ref</i> (*)	引用剧。	M	1	空
0.7.1	引用	<i>ref</i>	同上	M	1	标识符引用
0.8	学习单元超链接	<i>unit-of-learning-href</i>	此元素可以作为计算或表达式中的操作数，用于引用外部学习单元的相应元素，或许在同一包内（因此href是相对URI），或许指向包外的学习单元（因此href是绝对URI），需要在文件引用时添加分片标识符（#ID），其用法与XML	M	1	空

显示—隐藏						
编号	名称	英文名	解释	先决条件	重数	类型
			文档内部使用标识符引用的方法一样，指向包含此外部学习单元的活动结构、学习活动、支持活动或环境元素的ID。注意，其等同于简单XPointer，格式是 URI#ID，是XML中类似于HTML分片标识符的标记。在XML schema中，任何URI结构支持该格式。			
0.8.1	href	href	引用URI。	M	1	任何URI

6.2.16 全局元素信息表

LD规范中有四种全局元素。

B层规范中此元素皆为空。全局元素必须分开使用，即在任何XML内容大纲(比如XHTML)内，没有“全局元素”的容器元素。全局元素的功能只是对不同全局元素暂时分组，注意全局元素不是学习设计模型的组成部分，它使用标准命名空间。

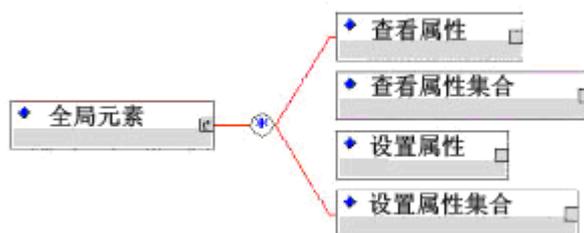


图32 全局元素信息图

表31 全局元素信息表

全局元素						
编号	名称	英文名	解释	先决条件	重数	类型
0	全局元素	global-elements	对元素的抽象包装，用于全局元素的暂时分组，无其他功能，不出现在学习内容或其它文档中，如同set-property，其用法自足，不依赖于包装的上下文。	—	—	选择
0.1	查看属性	view-property	全局元素，包含于外部XML内容大纲，资源类型是ldcontent。由此能够查看指定属性值，其作用于文本行之外(比如<p>元素之外)。设置查看属性，设置值或标题值是否将被发送。这涉及到一个集合属性或者超链接属性。为了避免混乱，最好把有属性操作的ld内容放在学习单元包中。在个	M	1	空

全局元素						
编号	名称	英文名	解释	先决条件	重数	类型
			人属性实例中，用户自身的属性值被返回了。在有一个监视对象（有指定的角色集合）的语境中，指定的角色的所有用户值都被返回。			
0.1.1	href	<i>href</i>	引用URI。	O	1	任何URI
0.1.2	属性源自	<i>property-of</i>	固定选择“self”或“supported-person”。选择“self”时，只引用个人属性值，选择“supported-person”时，引用所有个人属性值。 可能取值： <i>self</i> , <i>supported-person</i> 默认取值： <i>self</i>	O	1	标记
0.1.3	引用	<i>ref</i>	引用学习设计中的标识符。	O	1	标识符引用
0.1.4	查看	<i>view</i>	固定选择“value”或“title-value”。选择“value”时，只返回显示属性的值，选择“title-value”，返回标题的属性与值，对属性集合，也返回集合标题。 可能取值： <i>value</i> , <i>title-value</i> 默认取值： <i>value</i>	O	1	标记
0.2	查看属性集合	<i>view-property-group</i>	全局元素，包含于外部XML内容大纲，资源类型是ldcontent。由此能够查看指定属性集合中的属性值，其作用于文本行之外(比如<p>元素之外)。设置查看属性集合，设置值或标题值是否将被发送。这涉及到一个集合属性或者超链接属性。为了避免混乱，最好把有属性操作的ld内容放在学习单元包中。在个人属性实例中，用户自身的属性值被返回了。在有一个监视器对象（有指定的角色集合）的语境中，指定的角色的所有用户值都被返回。	M	1	空
0.2.1	href	<i>href</i>	同上	O	1	任何URI
0.2.2	属性源自	<i>property-of</i>	同上	O	1	标记
0.2.3	引用	<i>ref</i>	同上	O	1	标识符引用
0.2.4	查看	<i>view</i>	同上	O	1	标记
0.3	设置属性	<i>set-property</i>	全局元素，包含于外部XML内容大纲，资源类型是ldcontent。由此能够设定指定属性值，其作用于文本行之外(比如<p>元素之外)。设置属性，不管其值或标题与值是	M	1	空

全局元素						
编号	名称	英文名	解释	先决条件	重数	类型
			否发送。用户控制用户界面设置相应属性值，控制的类型取决于属性类型及其限制。控制时，显示当前值，明示数据类型及其限制，由此用户明确什么值有效或无效。此时允许客户端检查输入（取决于具体实现，也可以在服务器端处理），此运算引用属性URI或者有ref或 href之标识符。为了避免混淆，最好包含学习单元包中的属性运算与ldcontent。对于个人属性，设置用户自身的属性值。只有监控对象（指定角色引用），需要设置指定角色所有用户值。			
0.3.1	href	<i>href</i>	同上	O	1	任何URI
0.3.2	最大事务值	<i>max-transactions</i>	此属性表示用户可以设定属性的次数。技术上错误不计数，只计算成功事务。此属性未指定时，尝试次数设置为无限。	O	1	非负整数
0.3.3	属性源自	<i>property-of</i>	同上	O	1	标记
0.3.4	引用	<i>ref</i>	同上	O	1	标识符引用
0.3.5	事务类型	<i>transaction-type</i>	此属性用于未来扩展，比如安全事务。	O	1	字符串
0.3.6	查看	<i>view</i>	同上	O	1	标记
0.4	设置属性集合	<i>set-property-group</i>	全局元素，包含于外部XML内容大纲，资源类型是ldcontent。由此用户能够设定指定属性集合的值，其作用于文本行之外（比如<p>元素之外）。设置查看属性，决定显示相应值的标题，集合标题总是显示。用户控制用户界面设置相应属性集合的值，每个属性的控制类型取决于属性类型及其限制。控制时，显示当前值，明示数据类型及其限制，由此用户明确什么值有效或无效。此时允许客户端检查输入（取决于具体实现，也可以在服务器端处理）。所有属性的各个值在更新前必须由用户设置，事务只针对属性集合，不是单个属性，此运算引用属性集合标识符或者有ref或 href之URI。为了避免混淆，最好包含学习单元包中的属性运算与ldcontent。对于个人属性，设置用户自身的属性值。只有监控对象（指定角色引用），需要设置指定角色所有用户值。	M	1	空

全局元素						
编号	名称	英文名	解释	先决条件	重数	类型
0.4.1	引用	<i>href</i>	同上	O	1	任何URI
0.4.2	最大事务值	<i>max-transactions</i>	同上	O	1	非负整数
0.4.3	属性源自	<i>property-of</i>	同上	O	1	标记
0.4.4	引用	<i>ref</i>	同上	O	1	标识符引用
0.4.5	事务类型	<i>transaction-type</i>	同上	O	1	字符串
0.4.6	查看	<i>view</i>	同上	O	1	标记

6.2.17 全局属性“类”属性

条件判断能够显示或隐藏类属性中的元素，此属性是全局属性，定义于W3C的CSS中，存在于以下的学习设计模型：

- 学习对象
- 服务

除了学习设计，其能够添加于任意XML内容大纲。其存在于XHTML的所有元素中，条件不仅影响学习设计类属性中的元素，而且能够影响其内容，倘若其内容是ldcontent资源类型。

6.2.18 数据类型

以下数据类型用于属性声明，数据类型的格式如下。

1) *Boolean*: 表示二值逻辑，真或假(别名: yes/no; 1/0)。注意，如同其它数据类型，booleans型数据可以 <no-value>即无值。

2) *Integer*: 表示整数的标准数学概念，指整个正数、负数、零，取值范围-9223372036854775898至 922372036854775807 (别名longinteger)。

3) *Real*: 表示任意精确度的十进制数字的标准数学概念，至少能够处理18位有效数字。

4) *String*: 表示任意合法的字符串，最大字符长度2000。

5) *File*: 表示任意二进制文件类型，属性存储的是此种文件。

6) *Uri*: 依据IETF之RFC2396，表示URI。注意，依据W3C，未来只使用URI，不使用URL或URN (参见 <http://www.w3.org/TR/uri-clarification/>)。

7) *Datetime*: 表示日期时间，形式为CCYY-MM-DDThh:mm:ss。CC为世纪；YY为年份(禁止年份0000)；MM为月份；dd为日期，T为日期时间的分割符；hh为小时；mm为分钟；ss为秒数(参见ISO 8601)。时区为可选部分，不允许不完整表示形式。

8) *Duration*: 表示时间长度, 为事件的持续时间(比如学习单元执行的持续时间)。其表示形式也参照了W3C XML schema标准, 即PnYnMnDnHnMnS, 其中

P 必须保留的符合

n 表示需添入整数的变量

nY 表示年份的数字

nM 表示月份的数字

nD 表示日期的数字

T 表示日期时间的分割符, 必须保留, 倘若存在时间

nH 表示小时数

nM 表示分钟数

nS 表示秒数

比如P2Y0M1DT20H10M55S, 表示持续时间2年0月1天20小时10分钟55秒, 允许局部表示形式, 比如持续时间40分钟表示为PT40M, 持续30天表示为P30D。

9) *Text*: 表示任何合法字符串, 最大字符串长度64000 (大约A4的10页文本)。

6.2.19 限制类型

有时, 许多属性的数据值需要限制, 限制形式如下:

1) *length*: 限制文本数据类型(比如string、text、uri)的属性值的长度, 表示其可以拥有的字符数目。

2) *minLength*: 限制文本数据类型的属性值的最小长度。

3) *maxLength*: 限制文本数据类型的属性值的最大长度。

4) *enumeration*: 限制属性值取指定值(用于选择列表)。

5) *maxInclusive*: 限制有序属性(比如integer, real, datetime)值的指定上界。

6) *minInclusive*: 限制有序属性(比如integer, real, datetime)值的指定下界。

7) *maxExclusive*: 限制有序属性(比如integer, real, datetime)值的全局指定上界。

8) *minExclusive*: 限制有序属性(比如integer, real, datetime)值的全局指定下界。

9) *totalDigits*: 限制十进制数字属性值的指定有效数字长度。

10) *fractionDigits*: 限制十进制数字属性值的最大的小数有效数字长度。

11) *pattern*: 限制属性值取某固定表达式定义的模式。

6.3 C层信息模型

C层信息模型增加了一种功能，使得学习设计者可以指定基于确定性事件的消息发送和新活动的设置。我们希望执行时的系统，或者说“用户代理”能够支持通告机制。通告是事件驱动的机制，它们指向系统中的元素或人类用户。

通告会对B层模型中的以下一些内容产生作用：

- 1) “完成动作”模型通过一个通告元素得到扩展。
- 2) *then* 模型通过一个通告元素得到扩展。
- 3) 全局元素“属性设置”和“属性集合设置”通过一个通告元素得到扩展。

6.3.1 概念模型

图33提供了C层模型的概念化UML语言建模。图中灰色部分是相对于B层模型添加的类别。

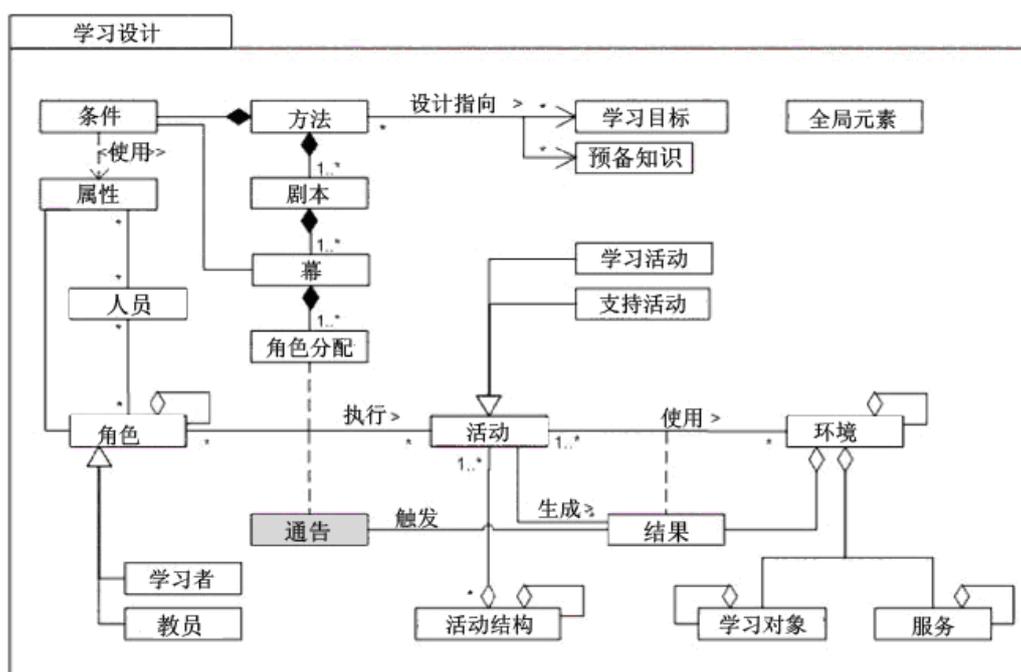


图 33 C层概念模型图

6.3.2 “通告”信息表



图 34 “通告”信息图

表 32 “通告”信息表

通告						
编号	名称	英文名	解释	先决条件	重数	类型
0	通告(**)	notification(**)	一则通告产生在一个被运行时环境所确定的事件之后。例如，这样一个事件可以是一个活动的完成，一个表达式被确定为真，或者一个属性的设定。通告能够为一个角色激活一项新的学习活动或支持活动，或者仅仅发送一条消息。一则通告拥有最高优先权，这意味着它可以让一个不可见的子内容项变得可见并被用户访问。在具体实现中，一条电子邮件消息可以被发送给用户，同时通知一个新的活动已经到达（通过消息中一个指向此活动的链接实现）。主题域能够被设定为一个特定的值（否则一条标准消息将被发送）。然而，一则通告可以被插入到外部词汇表中（在特定的事件之后插入，例如“属性设定”），之后必须在包中提供消息内容（因为消息内容包含了对包中标识符的引用）。当标识符无法被解析时，通告被忽略（但无法阻止正在被呈现的内容中的xhtml）。	—	—	序列
0.1	电子邮件数据	email-data	此元素在电子邮件发送中应用（作为环境或通告中的一项服务）。在B层模型中，这个元素的属性指向与连接角色相关的电子邮件数据的属性资源。在A层模型中，此元素未被精确指定，交由实施者来决定怎样表达所需数据。对所有赋予角色的人和邮件发送团体来说，这两个属性（电子邮件、用户名）都应当是可用的。	M	1..*	容器
0.1.1	电子邮件属性引用	email-property-ref	此属性包含一个被通知用户的电子邮件地址属性的引用。	M	1	任何URI
0.1.2	用户名属性引用	username-property-ref	此属性包含一个被通知用户的用户名属性的引用。	O	1	任何URI
0.1.3	角色引用	role-ref	指向角色资源的标识符。此元素在一个表达式中能被用作操作数。	M	1	空

通告						
编号	名称	英文名	解释	先决条件	重数	类型
0.1.3.1	引用	<i>ref</i>	指向一个在学习设计中的标识符。	M	1	标识符引用
0.2			选择	0	0..1	选择
0.2.1	学习活动引用	Learning-activity-ref	指向一个学习活动。此元素在一个算式或表达式中能被用作操作数。	M	1	空
0.2.1.1	引用	<i>ref</i>	同上	M	1	标识符引用
0.2.2	支持活动引用	Support-activity-ref	指向一个支持活动。此元素在一个算式或表达式中能被用作操作数。	M	1	空
0.2.2.1	引用	<i>ref</i>	同上	M	1	标识符引用
0.3	主题(**)	subject(**)	它指定了一个通告的主题，并且当该通告激活时呈现给被通知的活动者。例如，在邮件头部分（主题域）。	O	0..1	字符串

6.3.3 完成动作模型扩展

如下图，通告被添加进“完成动作”模型：

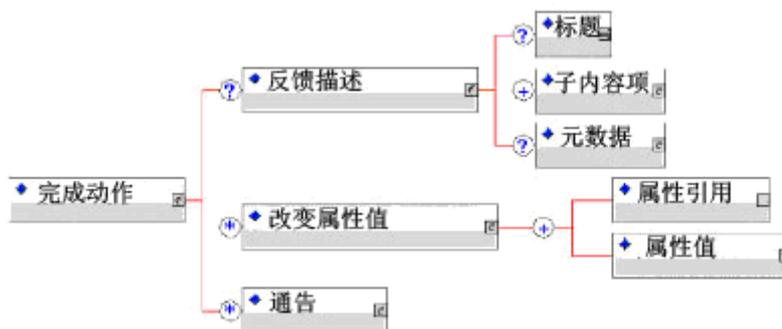


图 35 完成动作模型扩展信息图

6.3.4 then 模型扩展

如下图，通告被添加进“then”模型：

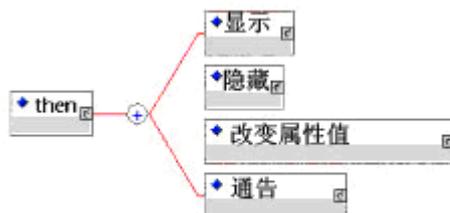


图 36 then 模型扩展信息图

6.3.5 全局元素模型拓展

如下图，通告被添加进“全局元素”模型：

- 属性设置
- 属性集合设置



图 37 全局元素模型拓展图

7. 行为模型

7.1 行为模型概述

学习设计的行为模型主要由两部分组成：

- 实例化
- 执行时

7.1.1 实例化

除了学习单元本身的实例化，实例化还有两个主要组成部分：

- 实例化角色
- 实例化服务

7.1.1.1 实例化角色

对于一个指定了角色但没有指定扮演角色人员的学习设计来说，对它的实例化在于为设计中指定的角色分配适合的人员。学习设计的 B 层模型中的属性或许将为众多角色提供实例化。

7.1.1.2 实例化服务

角色确定后，根据不同角色所需要的不同服务以及对不同角色许可范围的考虑，为实际的人员所准备的普遍服务便被建立起来。

7.1.2 执行

三个不同的学习设计级别为学习单元的执行行为提供了逐渐复杂的层次。

7.1.2.1 A 层学习设计

对 A 层学习设计来说，执行也分为两个主要部分：

- 活动结构编列和选择
- 方法

活动结构编列和选择：

一个活动结构能够以这些子元素的任意组合存在：活动引用、活动结构引用、学习外部引用单元。同时，提供了一套机制以决定怎样分配子元素以组成这些结构。

对 A 层学习设计来说，一个活动结构元素包含三个属性，它们影响着编列行为：

1) 结构类型

此属性的可选值为：{编列，选项集}

此属性决定子元素是否能被作为一个编列或者选择而递送。一个选择就是一个结构，在这个结构中用户可以任何次序选择和完成所包含的活动。就像活动结构可以被嵌套一样，选择也可以嵌套在其他编列或选择中。

2) 重复数

这是一个整数，它用来决定“结构类型”选择完成之前，有多少个子元素可以被递送。如果“结构类型”属性是选择并且“重复数”的值是 1，那么任何一个子元素都可被递送。如果“重复数”的值没有设置（或者其值刚好等于子元素的总数），那么这些子元素必须全部以用户所决定的序列被递送。

3) 排序

此属性的可选值为：{当——是，可见性次序}

“排序”属性决定与可见的活动有关的排列次序。默认状态下，可见活动的次序就是活动选择结构中指定的次序。当“排序”值被设置为“可见性次序”时，活动就以之前通过条件或通告将它们设为可见的那种序列呈现。这一过程如同模拟了一种收件箱，在这个收件箱中新的活动被“发送”出去（使之可见），由此随着时间的推移，对那些扮演相关角色的用户来说，这些活动逐渐变得可用。只有 B 层和 C 层学习设计才支持这一属性。

B 层学习设计通过增加属性和条件扩展了活动结构的控制选项，同时 C 层学习设计则增加了通告机制，用来建立动态编列。

方法：

方法支配着一个（包含学习设计的）学习单元的执行。一个方法有一个或多个剧。如果剧不止一个，则在逻辑上表现了学习设计的各个独立部分。因此，这些剧总是并发执行。并且，当学习单元处于激活状态时，这些剧对于执行它们的活动者来说总是可用的。

一个剧有一个或多个幕。幕总是按编列执行。这是一个基本的、高级别的、线性的编列方法，类似于戏剧中幕的次序

一幕中有一个或多个角色分配。这些角色分配和一个且只和一个角色相关联，这个角色指明了该角色在幕中扮演的部分和一些活动类型，这些活动类型描述了必须执行的活动和对这个活动来说可用的学习资源。一个角色可由一个或多个活动者扮演（例如，由真实的人来扮演学习者或支持员工的角色）。角色分配总是并发执行的，它使多位活动者参与到同一幕中。一个角色分配通过活动或包含一个或多个学习对象或服务的场景来与一个角色相关联。这种关联总是由引用一个角色和引用保留在组件部分中的活动或环境元素来完成。

“完成活动”即“活动完成的规则”，当一个活动完成的时候，它是学习活动和支活动声明和说明的一部分。在 A 层学习设计中，活动可通过用户选择和到达限制时间来完成。在 B 层学习设计中，通过“属性值被设定时”来扩展完成活动规则。

“完成动作”指定了活动完成所要执行的动作。在 A 层学习设计中，它只包含一个元素“反馈描述”，当用户完成活动时，其可将参考内容向用户显示。在 B 层学习设计中，通过“属性值更改”元素来进一步扩展“完成动作”；在 C 层学习设计中，则通过通告元素来完成扩展。

7.1.2.2 B 层学习设计

B 层学习设计包括属性和条件。条件用来个性化学习单元呈现。所有的条件都通过前导条件在设计期间定义，并且在执行期间的下述条件下必须接受评估：

- 当进入学习单元时（新的会话）。
- 每当一个属性值改变的时候。这条规则中的属性仅适用于以下情形：
 - ◆ 个人在学习单元的情境中所访问的属性。
 - ◆ 在学习单元的一个表达式中必须被评估的属性。

这些属性包括那些可用于表达式但可被自动设置的属性（例如：“学习单元开始时”）。

一个行为的执行或激发要根据条件是否满足，即其值为 **true** 还是 **false**。行为可以显示或隐藏各种对象，可以改变属性值，或通告一个角色。

显示行为和隐藏行为可设置不同对象的可见属性，这些对象是：活动，环境，子内容项，剧，活动结构，学习单元，包含其他对象的不同类（通过“类”属性设置）。

属性可以属于角色 (**local-role properties**)、扮演角色的个人 (**local-personal properties** 和 **global-personal properties**)，也可以属于学习单元的执行 (**local properties**) 或者其本身是全局属性 (**global properties**)。属性有五种不同的类型且可以选择组成属性集合。一个属性集合可包含另外的属性集合，从而可以创建任意的树型结构。

属性可以是个人的或非个人的

个人属性是个性化学习设计的关键。假如，许多活动者可以扮演一个相同的角色，那么个人属性虽然被定义为该角色的一部分，但它归属于每个扮演角色的活动者的个人档案（学习概况或学习记录）。于是，根据不同扮演者扮演该角色时的状态和成果，这些个人属性被赋予不同的值。

非个人属性可被置为全局属性（每个人的属性值都一样，独立于角色和学习单元），或者局部属性（这些局部属性适于学习单元的执行，而且能够对所有的用户产生不同的影响，包括扮演角色的用户和参与学习单元的用户）。局部属性在联系到特定实例或学习单元的执行时，总是在“局部”起作用（如下所述）。

属性可以是局部的或全局的

局部属性创建于一个学习单元的特定的执行期间，并且当该学习单元结束时便不复存在。因此，局部属性完全由学习设计者来控制，包括它们的名称、数据类型和给定条件下被赋予的值。

全局属性在一个学习单元的执行期结束后仍然存在。全局属性可被学习设计者用来在不同的执行之间和不同的学习单元之间保存属性值，也可被用来访问持久的学习者信息的值。然而，这已超出了学习设计规范的范围，可以参考 **LIP** 规范，并且可以通过为这些全局属性提供元素和属性值类型来考察其即时可访问性。

有五种类型的属性：

1) 局部属性（别称：执行属性）

此属性在执行时对每一个用户都具有相同的属性值，它归属于学习单元的执行。局部属性有一个标识符，用于在学习单元包中引用此属性，一个操作可以通过引用该标识符来操纵该局部属性值。

2) 局部个人属性

对于学习单元执行时所有角色的每一个扮演者，本属性都有不同的值。本属性归属于学习单元的一个执行中的所有用户，它为每个用户指定属性值。本属性有一个标识符，用于在学习单元包中引用此属性，一个操作可以通过引用该标识符来操纵该全局属性值。

3) 局部角色属性（别称：集合属性）

对于学习单元执行时特定角色的每个扮演者，局部角色属性都有相同的值。它归属于学习单元执行时的特定角色。局部角色属性有一个标识符，用于在学习单元包中引用此属

性，一个操作可以通过引用该标识符来操纵该局部角色属性值。

4) 全局个人属性（别称：学档属性）

全局个人属性独立于学习单元的不同执行，对于每个用户其有不同的属性值（属性值指明了用户的学档），此属性归属于个人。全局个人属性有一个标识符，用于在任何一个学习单元包中引用此属性，一个操作可以通过引用该标识符来操纵该全局个人属性值。

5) 全局属性

全局属性是一个全局性的唯一的属性，它只有一个属性值，而且独立于用户、学习单元和角色。全局属性有一个标识符，用于在任何一个学习单元包中引用此属性，一个操作可以通过引用该标识符来操纵该全局属性值。

7.1.2.3 C层学习设计

C层学习设计增加了通告。通告由能够被执行环境所感知的事件触发。例如，能够触发通告的事件包括活动的完成，计算表达式的值为真，或者设置了一个属性值。

执行时的通告可以设置活动的可见性，于是此活动能够被启动为角色服务。如果一个特定角色执行时的通告将给定活动的可见性属性值设置为真，那么无论其他的幕、编列或条件如何设置，这个活动立即对该角色变得可用。

通告拥有最高优先权，这意味着通过通告可以使得那些不可见的子内容项对于用户来说可见并且能够访问。

执行时系统给角色发送通信信息（可对所有扮演该角色的活动者产生影响）。当一定的条件得到满足，或当扮演角色（特别是助理教员角色）的活动者得到适当的允许发送信息时，通告被激活。根据具体实施情况，执行时系统有可能选择一个特定的活动者作为通告的接收者，但这取决于执行时系统而不是设计期间的决定，因为在设计期间无法得知确切的活动者。支持活动角色可以在扮演给定角色的活动者中选择，例如仅从给定导师集合的学习者当中接收通告是可行的。

执行时系统应当保留通告发起者的记录并使之对于接收方可见。在启动相关（支持）活动，挑选活动者以及激活一种挑战/应答交互时，这些上下文记录可被应用。

基于具体实施情况，一条电子邮件讯息可发送给用户，以告知他们一个新的活动已经到达（通过包含在讯息中的指向该活动的链接）。主题域可以包含一个特殊值（否则一个标准讯息将被发送）。

通告可被插入到外部词汇表中（在特定的事件之后插入，例如“属性设定”）。接下来，必须在包中提供消息内容（因为消息内容包含了对包中标识符的引用）。当标识符无法被解析时，通告被忽略（但无法阻止正在被呈现内容中的 xhtml）。

7.1.3 控制的层级

在学习设计规范中有一些结构，它们对学习活动和实体的可见性会产生影响。这意味着不同机制下的可见性存在冲突。为了解决这一冲突，设置了如下的控制层级：

通告 (C 层学习设计): 执行时的通告可以设置活动的可见性, 之后活动被启动。

幕 (A 层学习设计): 决定活动、资源结构、子内容项在何时、为何角色所用, 以及是否可用。

编列 (A 层学习设计): 是一种活动结构, 用来设置序列中活动的完成次序。不论条件和可见性属性的初始值如何, 它都将重置可见性属性的设置。

条件 (B 层学习设计): 不论活动、资源结构或子内容项的可见性属性的当前设置是什么, 都可被重置。

可见性 (A 层学习设计): 可见性属性决定活动、资源结构或子内容项能否被呈现给学习者。

以上控制的层级表明: 可见性属性由条件支配, 条件由编列支配, 编列由幕支配, 幕由通告支配。这意味着通告作为最健壮的机制, 其独立于方法选项中的任何声明, 并能够将组件选项中所定义的所有类型的活动设置为可见。

方法可用如下的图表示:

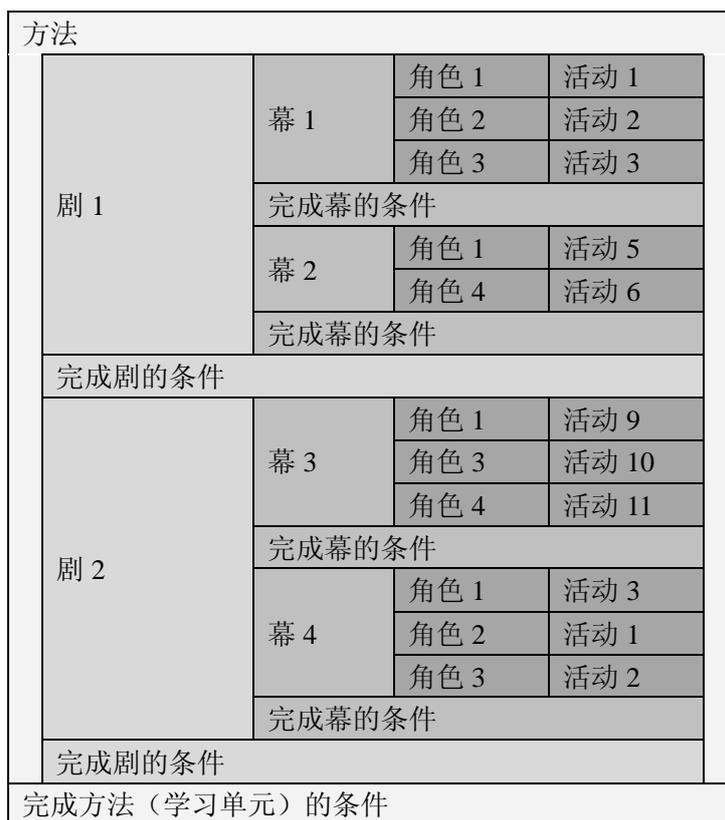


图 38 方法图

例如: 一个剧可用下图加以说明:

方法			
剧 1	幕	角色	角色分配
	1.1	教师	支持活动: 教师介绍
	1.2	学生	学习活动: 介绍

当所有学生完成活动时此幕结束		
2.1	学生	活动结构：课程和讨论
2.2	教师	活动结构：教学
当所有教师完成活动时此幕结束		
3.1	学生	学习活动：评估
3.2	教师	支持活动：结束活动
当教师完成活动时此幕结束		
最后一幕完成时此剧结束		
剧 1 完成时，此方法结束		

图 39 剧图

上例可在学习设计方法中做如下 XML 描述：

```

<method>
  <play id="play1">
    <act id="act1">
      <role-part id="part11"><role-ref ref="Teacher"/><support-activity-ref
ref="teacher-introduction"/></role-part>
      <role-part id="part12"><role-ref ref="Student"/><learning-activity-ref
ref="introduction"/></role-part>
      <complete-act><when-role-part-completed ref="part11"/></complete-act>
    </act>
    <act id="act2">
      <role-part id="part21"><role-ref ref="Student"/><activity-structure-ref
ref="lessons&discussions"/></role-part>
      <role-part id="part22"><role-ref ref="Teacher"/><activity-structure-ref
ref="teaching"/></role-part>
      <complete-act><when-role-part-completed ref="part22"/></complete-act>
    </act>
    <act id="act3">
      <role-part id="part31"><role-ref ref="Student"/><learning-activity-ref
ref="assessment"/></role-part>
      <role-part id="part32"><role-ref ref="Teacher"/><support-activity-ref
ref="closing-activities"/></role-part>
      <complete-act><when-role-part-completed ref="part32"/></complete-act>
    </act>
    <complete-play><when-last-act-completed/></complete-play>
  </play>
  <complete-unit-of-learning><when-play-completed ref="play1"/> </complete-unit-of-learning>
</method>

```

注意：在此例中，id 是 identifier 的缩写，标识符的名称也是任意指定的。

上面这段 XML 代码意思是：当进入学习单元时，整个剧以第一幕开始。所有被赋予教师角色的人员获得支持活动“教师介绍”，同时所有被赋予学生角色的人员获得被称为“介绍”的学习活动。

当所有扮演学生角色的人员完成介绍活动后，第一幕完成。接下来，第二幕开启，同时给所有学生角色人员分配“课程和讨论”的活动结构，给所有的教师角色人员分配“教学”活动结构。当教师完成其活动后第二幕结束（假设限定教师角色只能由一人担当）。接下来，第三幕开启，以此类推……当最后一幕完成，整个剧结束。

在一个方法中，可以指定多个剧，这些剧相互独立并可以同时执行，这对于更复杂的学习设计建模来说提供了便利。

通常采用 UML 活动图、甘特图或（相关的）时间表而非一张表格来进行建模，这使得学习设计便于上手。

图 4.1 是为每个角色配置泳道路径的 UML 活动图示例：

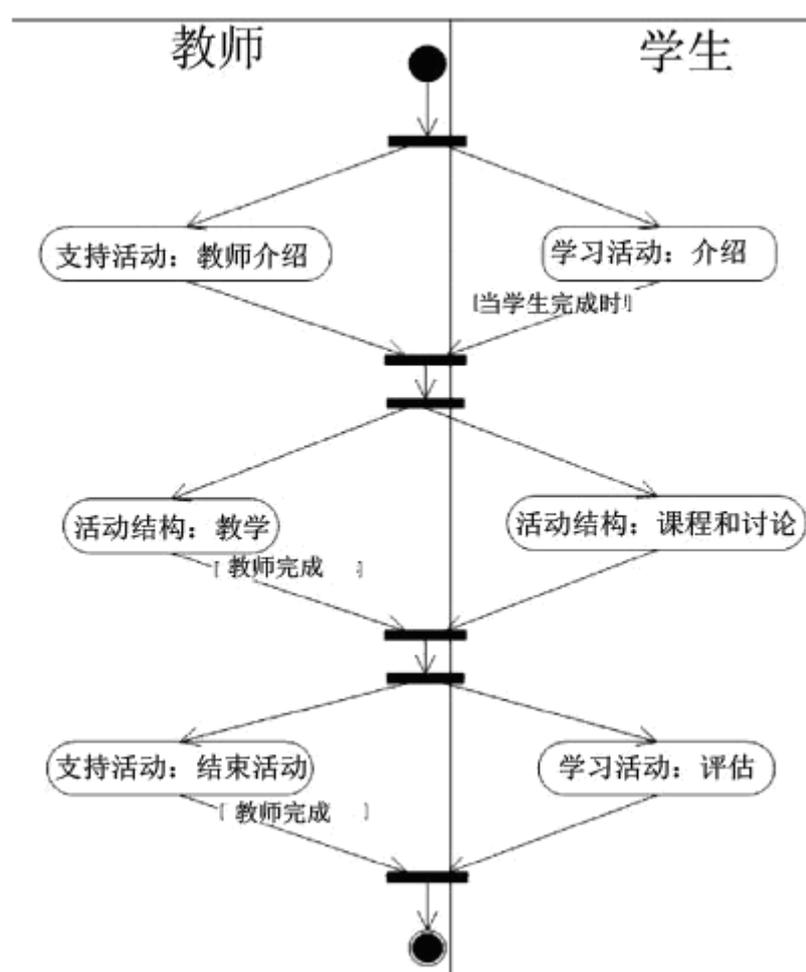


图 40 为每个角色配置泳道路径的 UML 活动图

图 7.2 是甘特图示例：



图 41 甘特图示例

7.2 学习单元实例化

包含一个学习设计的学习单元，在相同或不同的系统中、培训组织中以及教育机构中可被多次使用。每当实例创建的时候，执行时服务将需要建立一定数量的执行时特定数据结构，例如执行时标识符、启动时间等来支持该实例及其生命期。

需要采取一些步骤将学习设计 XML 文档转化为学生（和教师）能够在其间交互的现实“课程”形式，比如浏览器。这些步骤必须由学习设计的“用户代理”或执行时递送系统实施。步骤中需要指明的是：用哪种媒体来发布学习单元（典型的媒体是 Web，但学习单元也能够通过一些其他媒体或不同媒体的组合来递送）；有哪些特定的媒体元素（视频、音频，图片等）及对这些媒体可用性的支持；用何种语言；哪些人员会以什么角色参与，什么时间执行开始、什么时间结束？任何服务，例如通知频道、聊天室或论坛都必须按照设计中的指定，与扮演多种角色的活动者以及他们的“服务许可”设置一起建立起来。并且，数据库字段也得被建立起来以记录学习参与者的属性值并且保存他们的“档案”。

学习设计规范定义了部分执行时信息，但具体实施者更希望能够提供额外的执行时信息。

7.3 实例化建立角色和服务

实例以特定用户的使用为目的，因此需要向实例提供将要扮演学习者和学习支持员工角色的人员信息。

对学习设计的 B 层模型来说，当赋予一个人员角色时，任何属于该人员的属性需要定位在其档案中。所有归属于角色（由不同人员扮演）的局部个人属性将需要被建立起来并设定其默认值（假如有默认值的话）。那些扮演角色的所有人员的角色属性也需要被建立起来，就像在学习单元中扮演任何角色的所有人员的属性被建立起来一样。对于跨越所有学习单元和用户的全局属性来说，假如尚未被定位和实例化，它们也必须被定位和实例化。

一旦参与者和他们的角色已知，那么基于这些信息的服务便可被建立起来。学习单元可以被预先审视，以发现这些服务和这些服务建立起来并分配（执行时）URL 地址的实例。一个为学习单元特定实例服务的论坛或为学习单元中特定活动服务的论坛便是这样的例子，在这样的论坛中，不同的角色能够拥有关于读、写、删、编辑实体的不同权限。学习设计规范为这类信息提供了一种格式，使得这种服务的建立自动化。

7.4 启动过程

当启动一个学习单元时，方法元素（有且只有一个）必须被定位在学习单元中。方法元素和其子元素在总体上控制着学习单元的行为，同时也控制着众多角色扮演者的活动和他们对资源的利用。

这样便创建了一个“学习流”，其类似于工作流群件系统中的活动协调（但与面向文档的工作流系统中的文档段落不相类似）。

从功能上说，方法由一个或多个剧元素组成。剧元素在功能上是相互独立的且并发执行，所以每个剧元素在学习单元第一次初始化的时候必须被实例化。

用来描述方法的众多组成部分的术语，如剧、幕、角色和角色分配都来自于戏剧的隐喻，环境相当于戏剧中的舞台设置和道具。方法、剧、幕和角色分配均可以相互嵌套，前提是在方法中有三个级别。

在最高级别中，方法由剧和“完成学习单元”两个元素组成。完成学习单元元素在其所遵循的完成规则部分得到描述，它控制着两项条件，即学习单元的完成和完成后的可选择行动。

在初始化的时候，为了参与其中的角色成员的确定，所有的剧元素都将被激活。

剧元素以一系列一个或多个幕呈现，这些幕总是以一定的次序执行。只有当前一幕结束后，后一幕才对其中的扮演角色可见。这一因素可被用来同步执行在剧中扮演角色的人员的活动。

与剧中一定角色相关联的活动也许具有复杂的编列，所以只要扮演者的活动不要求同步执行，那么可能存在只有一个幕的剧，幕中有复杂的活动编列（例如，当对一个学习资源的使用完成时，所有的参与者前往论坛讨论）。

剧拥有一个标识符和可见性属性，同时具有标题和元数据。幕构成了剧的主体，“剧完成”元素明确了当剧完成的条件和接下来的可选择行动。

尽管在一个幕中可能存在复杂的编列，同时剧中也可能只有一个幕，但只要剧中存在一个以上的幕，那么这些幕是按照编列次序执行的。

因此，幕可被用来作为活动同步执行的交汇点，它可以等待所有用户完成一幕后再启动下一幕，或者当其中一些用户完成活动后强行终止该幕（被学习支持人员终止或其他条件下终止）。于是，在同步登陆执行时系统服务的条件下，所有参与下一幕演出的人员同时启动。当最后一幕结束的时候，整个学习单元也就完成了。

一幕中有一个或多个角色分配。在这种机制下，在同一时刻允许多个角色执行活动。因此，在幕中的角色分配总是并发执行的。

角色分配使得扮演相同或不同角色的人员能够参与到同一幕中。每个角色分配将一个角色和一种类型活动（包括另一个学习单元和活动结构的执行）或一个环境（等同于内容包装中的组织）精确联系起来。在幕中的多角色分配是并发执行的。

在不同的角色分配中，同一角色可关联于不同的活动或环境，同一活动或环境也可关联于不同的角色。然而，在同一幕中相同角色只能被引用一次。如果多重活动或环境需要关联到同一角色上，那么应当使用活动结构或包装环境。

当剧中的幕被启动的时候，幕中所有的角色分配被激活。于是根据具体执行情况，角色分配引用的角色扮演者能够得到相关的活动或环境（这些活动或环境在扮演者的活动树中是可见的），同时能够得到与可访问活动相关的内容。然而，如果活动或子内容项的可见性属性被设置为“假”，活动树中的链接可见，但内容却不能被访问。

7.5 完成规则

在方法中的每个级别上，当角色分配、幕、剧或完成学习单元的时候，指定角色是可能的。我们期望执行时系统能够保存这些不同实体的完成状态的记录。通过本规则中的一些构造，可重新得到完成状态的记录。

在方法的最低级别，角色分配必须完成。角色分配引用的活动可以是一个学习活动、一个支持活动、一个活动结构或（子）学习单元。当“完成活动”条件被满足（对学习活或支持活动来说）的时候，或者当一个活动结构、一个引用的学习单元完成的时候，这些活动就结束了。

尽管活动结构可包含（子）活动结构和（子）学习单元引用，但最终其将被分解为基本的学习活动或支持活动。当最后引用的实体完成的时候，被设置为“编列”类型的活动结构完成。当其中的所有引用实体完成，或被设置为“number-to-select”的实体成员完成的时候，被设置为“选择”类型的活动结构完成。由此，活动结构或（子）学习单元的完成取决于其原子结构——学习活动或支持活动的完成。如果包含子结构，那么所有这些子结构完成就标志着整个活动结构或学习单元的完成。

学习活动或支持活动的完成取决于用户的选择或时间限定（A 层学习设计中）。当没有精确的完成规则被指定的时候，完成被设置成没有限制条件的，这就是说活动总是能够被完成的。在 A 层学习设计中，从低到高的三个层次是幕、剧和学习单元，每层各自拥有三个完成选项。

在角色分配的上一级别，当一个或多个引用的角色分配完成，或经用户选择，或当时间限制到达的时候，一幕就结束了。再往上一个级别，当最后一幕完成，或经用户选择，或当时间限制到达的时候，一个剧就结束了。最后，在最高级别，当一个或多个引用剧完成，或经用户选择，或当时间限制到达的时候，学习单元就结束了。

注意：无论是用户选择或是时间限制到达，最高级别元素的完成将终止低级别中的元素。

B 层学习设计增加了“当属性值被设置”选项，以此作为上述 A 层学习设计中为幕、剧、学习单元的完成所设置的三级选项的补充。“当属性值被设置”元素包括一个“属性引用”和一个“属性值”。当引用属性被设定的值与“属性值”中指定的值相同时，条件为真并将幕、剧和学习单元设置为完成。

除了第一个完成条件各自独有外，幕、剧和学习单元的其他所有完成条件都是相同的。

B 层学习设计增加了更多的完成选项，这些选项通过属性值被设定为给定的值而被触发。当指定属性被设置为指定值时，完成条件“当属性值被设置”被触发。进行再次测试的属性值能够被指定为字母值、计算值或其他属性值。

7.6 完成动作

当学习活动、支持活动、幕、剧或者学习单元完成的时候，接下来要执行的活动包含在“完成动作”元素中。完成动作选项对于学习活动、支持活动、幕、剧和学习单元来说都是相同的。当一个角色分配的引用完成活动并且其本身没有完成元素的时候，那么该角色分配被认为已经完成。同样，当一个活动结构内的各个子活动完成且其本身没有完成元素的时候，此活动结构被认为已经完成。

在 A 层学习设计中，唯一可被确定用来执行完成任务的工作就是通过“反馈——描述”元素给用户反馈。这样便指向了一个资源，反馈描述在此资源中，完成活动后，Web 页将呈现给用户。

B 层学习设计增加了选项，用以通过“改变属性值”元素来更改一个或多个属性值。“改变属性值”元素同样也包含“属性引用”和“属性值”。该元素在完成时指定，“属性引用”的值被设置为指定的“属性值”。

C 层学习设计增加了发送一个或多个通告的选项（参见描述通告的行为模型概览）。

7.7 学习成果记录和其 LIP 规范映射

在学习单元中每个学习者参与活动的学习成果一般通过执行时服务记录并被保存。学习设计预支持一些形式的“档案”或学习者记录，这些档案和记录用来持有和保留隶属于 B 层学习设计的多种个人属性。局部个人属性只需要在学习单元执行期间被保留，尽管执行会跨越多种会话并耗时数天。全局个人属性趋向于持续的不确定，其应该成为学习者持久学习记录的一部分。

学习设计没有指定执行时服务应当怎样记录每个学习者的属性、属性值及其汇聚。

然而，值得注意的是学习设计属性和属性集合可直接映射为 LIP 规范的活动和评估元素。这意味着 LIP 规范可将学习者使用学习单元时产生的学习成果在需要这些成果的不同系统间传递。

8. 可扩展性

学习设计的 XML 绑定可通过 XML 名称空间和 XML Schemas 加以扩展，这样就使得开发人员有更大的灵活性。包含数据类型的元素（例如：字符串，整型）和拥有“封闭”数据模型的元素可能无法扩展。扩展必须通过名称空间来提供扩展来源的引用。

信息模型已经指示了占位符，它可被替代，也可通过其他的大纲加以扩展。

对于开发者来说，扩展至少在两种情况下会导致问题出现。第一种情况就是当需要与另外的内容包装工具和供应商进行互操作的时候，用户扩展必须在那些全局互操作极难实现的个别团体间达成一致。第二种情况就是当开发者希望扩展并且提供或改变了承认文档有效性的大纲，每个大纲 (DTD, XSD)需要不同的方法来处理那些有效的扩展。

附录 A 用户代理标准

学习设计提供3级标准，即基于内容包装的A层、B层、C层。A层支持所有的内容包装结构，B层建立于A层之上，因而兼容于A层，同样C层建立于B层之上，因而兼容于B层及A层。用户代理符合LD标准，意味着其有能力实现XML schema中的所有元素与属性，倘若Profile 3、4、5符合本规范描述。针对每个标准，期望用户代理有相应的执行时行为。

A.1 - A 层标准

1) 学习单元包囊括所有文件，由此创建这一学习单元的一个或多个执行。学习单元的URI能够唯一识别该包，即便是其升级的版本。学习单元包执行时，学习设计不能更新，但是其发送的组织结构与本地物理文件可以更新，不会影响用户的执行状态。

一旦新本地文件或学习设计创建完成，即可以为学习单元包创建一新的URI。依据实现情况，信息的多种内容可以存储于URI中（比如标识符+类型+版本号）。执行时，拥有新URI的新包可以在执行时发布（通过更新资源与文件）或者创建一个新的执行。执行时系统应该能够引入、发布，设置启动日程、管理用户角色、更新现有执行、创建新的执行。

2) 解释LD时，执行时系统读取学习设计、方法、剧元素的结构（叫做剧）。

倘若指定若干剧，就必须并发解释之。一个剧有若干幕，一个幕有若干角色分配。在各级学习设计中，必须有明确规则，指定角色分配、幕、剧乃至学习单元是如何完成的。并且期望执行时保留这些不同实体的完成状态。此完成状态由某些LD结构检索。

角色完成状态必须从幕活动状态乃至学习单元完成状态导出（假如角色分配引用了学习单元）。倘若无明确完成状态被指定，则对完成状态不作任何限制，即总是完成的。

3) 对每个剧，其幕是按照指定序诠释的，只有每个剧的某一幕能够出现于任意时间，倘若启动了剧的第一幕。一旦该幕完成，下一幕开始，直到演完所有幕。

4) 角色分配指定角色能够访问哪些活动。倘若剧中有若干角色分配，不同角色可以同时访问角色分配。

5) 执行时绑定于某角色的用户，可以访问该角色的可以访问并且可见的角色分配。用户绑定于执行时的角色（添加新用户，删除老用户）。

6) 倘若角色标记为允许创建新角色，可见性规则与父辈用户可应用于子用户。由于这些规则，新角色创建者可以将父辈角色中的老用户重组至新创建的角色中。

7) 元素的可见属性被诠释为初始化可见性取值，Level B之条件与Level C之通知能够

设置该值为真（倘若需要显示），或者设置为假（倘若需要隐藏）。

8) 学习设计的元数据，用户界面总是可以访问。在用户界面规范中，必须指定针对所有元素的某格式，倘若未指定，其显示格式为[元素名称]:[元素值]。

9) 其它元素的元数据之可访问与可见性，取决于用户界面设计（依赖于具体实现）。

10) 学习设计的标题在用户界面中总是可见的，并且不可改变。

11) 其它标题元素可以由界面处理，倘若具体实现时需要。模板可以覆盖现存规则。

12) 学习设计/学习目标/子内容项以及预备知识/子内容项必须为所有角色访问，在用户界面的任何时间。其或许需要用户动作（比如打开菜单，单击按钮或超链接）。其语义上必须表示为针对不同预备知识的学习目标。

13) 用户必须总是清楚自身的角色。当用户赋予若干角色时，其必须看到其所赋予的角色，并且能够在任何时间切换过去。角色/子内容项信息必须总是能够为用户角色访问。

14) 学习活动与支持活动的处理是在用户界面中，此时用户总是清楚其所处的学习活动（依据活动标题），并且切合于活动顺序及其选择，即关联于何种活动描述、何种学习目标与预备知识、何种环境或环境内容、如何完成活动（包括必要的控制）以及访问任意的某完成子内容项的反馈信息。

15) 一旦活动结构为用户了解，用户代理必须诠释活动结构，以如下的结构顺序，即信息元素的内容为用户可见，相关环境及其内容为用户可用，活动也可以被按照指定顺序依次访问，前者完成，后者必须立刻可用。活动选择的方式，应该让用户看到信息及其相关环境，可以访问相关活动，包括通知用户的信息，倘若限制所选子内容项的最大访问量。活动编列之完成，在于完成了相关的学习单元或最后一个活动（包括子编列、子选择）。选择结构的完成，在于用户完成了应该完成的众多子内容项（如果未设置重复数，必须完成选择项中的所有活动）。无论如何，用户必须明白此刻其所链接的活动应该访问的活动类型（学习活动，支持活动、子结构活动或学习单元）。

16) 支持活动对所支持角色中的各个用户都执行一遍（倘若指定角色），用户界面必须实现这种要求，任何时候都能够明确哪位用户或用户组要实施支持活动。用户可以选择所支持角色的若干用户。

17) 环境关联于活动、活动结构以及角色分配，一旦活动描述可见，总是可以看到相关的环境（包含环境的内容结构）。同时就可以访问、看到活动描述以及环境中的对象或服务的内容。

18) 环境中的各个学习对象或服务都有其自身的学习需求，实现者自主实现学习对象与服务，注意表示相关的所有的元素与功能，要求与信息模型所描述的元素一致。

19) 资源不直接显示于用户界面，其定义了学习设计中的子内容项元素所引用的资源集合，局部定义资源与文件（在超链接中用相对URIs描述）在学习单元包中，绝对URI指向包外的资源，改变外部资源不影响学习单元包（该包不知道外部资源的存在与改变）。

20) 子内容项元素出现于信息模型的许多地方。子内容项指向学习单元包中的资源，资源内容在用户界面中处理，内容本身可见，包括若干模型的结构也可见，同时子内容项标题也必须可视化，内容表格与分段文本以后处理，嵌套的子内容项对用户也必须可见（比如提供嵌套标题）。

A.2 – B 层标准

1) 条件在幕及其前继幕的活动范围内起作用，条件不能让活动可见，倘若其在当前剧的当前幕的角色分配中未指定，由此可见，幕的优先级高于条件。

2) 条件之判定，可以在用户访问树或内容的任何时候，只要系统在线。条件判定之执行在于其影响了所选的内容或树。同时，只有引用了if表达式中的属性并且其值发生改变从而不同于用户最近一次的取值时才执行条件表达式。

3) 用户代理用于为档案中的用户与角色保存记录属性值与属性定义。存在若干种属性类型。

局部属性存储学习单元的局部执行的结果，其定义并且使用于学习单元。全局属性在学习单元外才能访问（比如存在若干学习单元时），其定义于某学习单元但是在另外的学习单元使用。在LD中可以定义全局单元，执行时控制所定义的全局属性URI是否存在。一旦定义全局属性就不能改变，倘若属性已经存在就忽略此定义。

个人属性由个人拥有（局部或全局），角色属性由角色拥有（局部或全局）。档案属性定义或声明（倘若全局属性已经存在）于学习设计/角色/属性，其对属性运算元素的运算包括查看属性、设置属性、条件判定等。用户代理的运算需要在安全方式下，并且拥有最大操作数目（实现者细化）。一旦属性值改变，C层中的通知要发送给某角色（取决于声明）。

A.3 – C 层标准

1) 一旦通知发出，相关活动必须为通知到的角色中的用户可见。倘若角色分配于当前幕，活动即可见于当前幕的结构树中，倘若角色不在当前或其前继幕，活动被添加于结构树中，作为单独节点，直接位于当前幕的上方或下方。通知抵达后必须改变用户。

2) 通知的优先级高于幕，幕的优先级高于编列，编列的优先级高于条件

3) 显示与隐藏的条件冲突时，显示的优先级高于隐藏。

参考文献

- [1] EML reference manual (<http://eml.ou.nl>)
- [2] CELTS9.1 内容包装信息模型规范 (<http://www.celtsc.edu.cn>)
- [3] GB/T 21365-2008 信息技术 学习、教育和培训 学习对象元数据规范 (<http://www.celtsc.edu.cn>)
- [4] CELTS-8.1 简单课程编列规范：参考模型 (<http://www.celtsc.edu.cn>)
- [5] Modelling units of study from a pedagogical perspective: the pedagogical metamodel behind EML, Koper E. J. R., 2001: (<http://eml.ou.nl/introduction/docs/ped-metamodel.pdf>)
- [6] CELTS-10.1 测试互操作：信息模型标准 (<http://www.celtsc.edu.cn>)

- [7] IMS Reusable Definition of Competency or Educational Objective (RDCEO) Specification (<http://www.imslobal.org>)
- [8] CELTS-11.1 学习者模型规范：信息模型(<http://www.celtsc.edu.cn>)
- [9] IMS Enterprise Specification (<http://www.imslobal.org>)
- [10] ADL SCORM (<http://www.adlnet.org>)
- [11] CLEO, content aggregation (<http://www.lsal.cmu.edu/lsal/expertise/projects/cleo/report20010701/working/aggregation.html>)
- [12] OASIS DOCBOOK
(<http://www.oasis-open.org/docbook/documentation/reference/html/docbook.html>)
- [13] Unified Modeling Language (<http://www.omg.org/technology/documents/formal/uml.htm>)
- [14] W3C HTML 4.0 specification
(<http://www.w3.org/TR/REC-html40/struct/global.html#h-7.5.2>) (for the definition of the 'class' attribute)
- [15] IETF (<http://www.ietf.org>), relevant specifications: URI, ftp, news, smtp, http
- [16] W3C (<http://www.w3c.org>) consortium for Web-related interoperability specifications: HTML, XHTML, XML 1.0, XML schema, XML namespaces, XSLT
- [17] Vogten, H., Verhooren, M., & Koper, E. UML diagrams
(<http://eml.ou.nl/introduction/docs/uml.pdf>)