



中华人民共和国国家标准

GB Z×××—200×

信息技术 学习、教育和培训 学习系统体系结构与服务接口 第二部分：核心接口

Information technology - Learning, education and training -
Learning system architecture and service interface
Part II: core interface

(送审稿)

XXXXXXXXX 发布

XXXXXXXXX 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目 次

前 言	II
引 言	III
1 范围	1
2 术语和定义	1
3 接口	1
4 基础服务接口	1
4.1 基础 (base) 服务	1
4.2 认证 (authentication) 服务	3
4.3 数据库管理 (databasemanagement) 服务	4
4.4 文件 (File Management) 服务	6
4.5 标识 (Identification) 服务	9
4.6 日志 (Logging) 服务	9
4.7 授权 (Authorization) 服务	10
4.8 工作流 (Workflow) 服务	14
4.9 XML 服务	18
5 教育服务接口	19
5.1 评价 (assessment) 服务	19
5.2 课程管理 (Course Management) 服务	26
5.3 成员管理 (Membership Management) 服务	28
5.4 组管理 (Group Management) 服务	29
5.5 编列 (sequencing) 服务	31
5.6 学习对象管理 (Learning Object Management) 服务	33
5.7 内容包装 (Content Packaging) 服务	46
附录 A JAVA 接口	51
A.1 基础服务接口	51
A.2 教育服务接口	72

前 言

本标准分为 6 个部分：

- 第 1 部分：抽象框架
- 第 2 部分：核心接口
- 第 3 部分：学习资源服务接口
- 第 4 部分：学习者信息服务接口
- 第 5 部分：学习环境服务接口
- 第 6 部分：教育管理服务接口

本部分为系列标准的第 2 部分。

本部分由教育部提出。

本部分由全国信息技术标准化委员会归口。

本部分起草单位：清华大学、华东师范大学、华中师范大学。

本部分主要起草人：郑莉、张新钰、刘京、刘银涛、吴永和、刘清堂、王绚、许磊。

本部分的附录A为资料性附录。

引 言

在实施网络教育的过程中，经常会遇到互操作的问题，这种互操作包括学习资源的互操作以及学习系统间的互操作。本标准就是为了解决互操作的问题而制定的，它从多个角度描述数字化学习系统的软件体系结构、服务的分层结构，以及服务的接口定义。本部分定义了数字化学习系统软件中的核心服务接口。

信息技术 学习、教育和培训

学习系统体系结构与服务接口

第二部分：核心接口

1 范围

本部分遵循本标准第1部分所规定的学习系统体系结构抽象框架，定义了数字化学习系统软件中的核心服务接口。

本标准适用于不同的数字教育学习平台通用服务及接口的定义及数字教育学习平台的接口规范化。

2 术语和定义

本标准第1部分以及 ISO/IEC 2382-36: 2008《信息技术 词汇 第36部分：学习、教育和培训》确立的术语和定义适用于本标准。

3 服务接口

本部分中的服务接口按照本标准第1部分第5章“服务的分层结构”进行定义。本部分中所定义的服务接口与“服务的分层结构”中的服务有着直接的对应关系，一个服务的功能由多个接口来构成。

接口给出的是系统行为模型，并不强制绑定到某种具体语言。为了便于理解和直接用于开发，本标准附录给出了服务接口的 Java 语言绑定。

本规范给出的接口并不是学习系统所用到的服务接口全集，本部分所给出的是在本标准第1部分“抽象框架”指导下的核心集。根据本标准第1部分中的分层，本部分将核心接口分为：基础服务接口和教育服务接口。

4 基础服务接口

4.1 基础（base）服务

Base 包提供的是一些基础接口，供其它接口使用，这些接口通常会被继承。

4.1.1 Manager 接口

Manager 是将应用程序绑定到具体服务实现的关键，接口的描述需要定义一个接口扩展 Manager，而接口的实现要求以一个类实现具体的 Manager。应用程序直接或间接通过 Manager 访问服务对象。

Manager 接口定义			
成员	返回值/类型	参数说明	说明
getContext	CeltsContext	无	获得传递的上下文内容
assignContext	空	context: CeltsContext	给上下文赋值
assignConfiguration	空	configuration: Properties	对环境参数赋值

4.1.2 Properties 接口

Properties 接口用于返回对象的一些只读属性，这些属性通过一系列串行化的值得到，并用关键码标记。

Properties 接口定义			
成员	返回值/类型	参数说明	说明

getType	CeltsType	无	获得属性类型
getProperty	可序列化对象	key: 属性的关键码, 可序列化对象	获得属性
getKeys	ObjectIterator	无	得到属性的关键码

4.1.3 CeltsContext 接口

CeltsContext 包含应用程序共用的语境信息。

CeltsContext 接口定义			
成员	返回值/类型	参数说明	说明
assignContext	无	key: 字符串 context: 可序列化对象	为 CeltsContext 的语境信息的内容赋值
getContext	可序列化对象	key: 字符串	得到 CeltsContext 的语境信息

4.1.4 Iterator 接口

Iterator 提供的是迭代器接口, 用于数据的批量读取, 对于具体的对象类型, 将通过继承 Iterator 类实现自己的迭代器接口。

Iterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNext	布尔值	无	是否还有下一个元素
Next	空	无	取下一个元素

4.1.5 PropertiesIterator 接口

PropertiesIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextProperties	布尔值	无	是否还有下一个属性
nextProperties	Properties	无	取下一个属性

4.1.6 CeltsType 接口

CeltsType 接口按照事物的基本概念进行分类, 被用于各个接口中。CeltsType 的精确定义由使用者确定, CeltsType 接口的形式也有利于分类, CeltsType 有四个字符串属性: authority、domain、keyword 以及 description, 前三个元素被 isEqual 成员用来判断两个 CeltsType 对象是否相等, description 用来阐明实例的语义。

CeltsType 接口定义			
成员	返回值/类型	参数说明	说明
getAuthority	字符串	无	获得 CeltsType 的 authority
getDescription	字符串	无	得到 CeltsType 的描述
getDomain	字符串	无	得到使用域
getKeyword	字符串	无	得到关键字
isEqual	布尔值	celtsType2: CeltsType	判断两个 CeltsType 对象是否相同

4.1.7 TypeIterator 接口

TypeIterator 接口定义			
--------------------------	--	--	--

成员	返回值/类型	参数说明	说明
hasNextType	布尔值	无	是否还有下一个类型
nextType	CeltsType	无	取下一个类型

4.1.8 IdIterator 接口

IdIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextId	布尔值	无	是否还有下一个 ID
nextId	Id	无	取下一个 ID

4.1.9 ObjectIterator 接口

ObjectIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextObject	布尔值	无	是否还有下一个对象
nextObject	可序列化对象	无	取下一个对象

4.2 认证（authentication）服务

服务描述：认证服务从学习代理处收集必要的认证信息，确保认证的真实性。这样调用此服务的应用程序就可以确定并处理学习代理的认证状态，而不必管理一个特定机构环境的细节内容。此服务需要建立在各种认证设施之上，许多机构都已经有了或者正在努力建立自己的中心认证。技术上的具体例子有 Kerberos, X.509, 以及基于 cookie 的单点签入技术，如 webISO。

一些应用程序依靠用户 id/密码进行认证，也有的是基于授权证书，大部分用户认证是本地化的，但也有一些是远程认证。这种范围在接口上使用认证类型的概念来表示。并且一个机构范围内会有一个默认类型，和经常使用的一些通用认证方法。

学习代理，通过一个或多个这样的类型得到认证。很多情况下，在一次会话中一个代理可能会进行多次认证。这主要发生在一个学习代理从一种学习活动转向另一种学习活动是时候。例如检查班级进度日程，可能只要用户 id 和密码就足够了，而更新最终的学习成绩可能需要的是 X.509 证书以确保更高水平的安全认证。

4.2.1 AuthenticationManager 接口

AuthenticationManager 接口定义			
成员	返回值/类型	参数说明	说明
getAuthenticationTypes	TypeIterator	无	获得认证类型
authenticateUser	空	authenticationType: CeltsType	根据认证类型对用户认证
isUserAuthenticated	布尔值	authenticationType: CeltsType	判断用户是否已经过认证
getId	Id	authenticationType: CeltsType	得到认证用户的 ID
destroyAuthentication	空	无	清除认证记录
destroyAuthenticationForType	空	authenticationType: CeltsType	清除同类型的认证记录

4.3 数据库管理 (database management) 服务

Database Management 服务提供了数据库管理的基本操作，包括对关系数据库中常用数据类型的封装定义，以及存取操作及结果表达的定义。

4.3.1 Blob 接口

Blob 是关系数据库中常用的一种类型，在 Blob 列中可以存放大数据量的二进制数据，如图片，音频数据等，该接口提供对数据表中 Blob 列的访问。Blob 会记录当前的读写位置，是一个长型整数型整数。

Blob 接口定义			
成员	返回值/类型	参数说明	说明
length	长型整数	无	返回该 Blob 数据的长度，单位字节
getBytes	字节数组	pos: 长型整数 length: 整型	从 pos 指定的位置开始，读取 length 字节的数据，结果以字节数组形式返回
setBytes	整型	position: 长型整数 bytes: 字节数组	将 bytes 所含数据写入该 Blob 中从 position 开始的地方
setBytes	整型	position: 长型整数 bytes: 字节数组 offset: 整型 length: 整型	将 bytes 中从 offset 开始的长度为 length 的数据写入该 Blob 中从 position 开始的地方

4.3.2 Clob 接口

Clob 是关系数据库另一种常用的类型，在 Clob 列中可以存放大量的字符数据，如大量的文本，该接口提供对数据表中 Clob 列的访问。

Clob 接口定义			
成员	返回值/类型	参数说明	说明
length	长型整数	无	返回该 Clob 数据的长度，单位字节
getSubString	字符串	pos: 长型整数 length: 整型	从 pos 指定的位置开始，读取 length 个字符，结果以字符串返回
setString	整型	pos: 长型整数 string: 字符串	将 string 字符串写入该 Clob 中从 position 开始的地方，返回成功写入的字符数
setString	整型	pos: 长型整数 string: 字符串 offset: 整型 length: 整型	将 string 字符串中从 offset 开始的长度为 length 的子字符串写入该 Clob 中从 position 开始的地方，返回成功写入的字符数

4.3.3 Connection 接口

Connection 接口代表了一个到数据库的连接，通过该连接可以执行数据库的查询、修改操作，另外不再使用的数据库连接可以释放，以节省数据库连接资源。

Connection 接口定义			
成员	返回值/类型	参数说明	说明
executeQuery	ResultTable	Sql : 字符串	执行 sql 所表示的查询语句，结果以 ResultTable 返回
executeQueryWith	ResultTable	sql: 字符串	执行 sql 所表示的查询语句，sql 中

Params		params: 字符串数组	的占位符使用 params 提供的具体值替换, 结果以 ResultTable 返回
executeUpdate	整型	Sql: 字符串	执行 sql 所表示的更新语句
executeQueryWith Params	整型	sql: 字符串 params: 字符串数组	执行 sql 所表示的更新语句, sql 中的占位符使用 params 提供的具体值替换
close	空	无	关闭该连接

4.3.4 ResultMetaData 接口

ResultMetaData 接口可以获取关于一个 ResultTable 的列的信息, 如类型, 列名称等; ResultMetaData 通过 ResultTable 的 getMetaData 方法获取, 提供关于对应该 ResultTable 的元信息。ResultMetaData 提供如下信息: 列数目、列名称、列类型、列显示时需要的宽度 (字符宽度)、是否接受 NULL 值等。

ResultMetaData 接口定义			
成员	返回值/类型	参数说明	说明
getColumnCount	整型	无	获取列数目
getColumnName	字符串	columnIndex: 整型	获取列 columnIndex 的名称
getColumnType	整型	columnIndex: 整型	获取列 columnIndex 的类型常量
getTableName	字符串	columnIndex: 整型	获取列 columnIndex 原先所属的表名
getColumnDisplaySize	整型	columnIndex: 整型	获取列 columnIndex 显示时需要多少字符宽度
isNullable	布尔值	columnIndex: 整型	判断列 columnIndex 是否可接受 NULL 值
isReadOnly	布尔值	columnIndex: 整型	判断列 columnIndex 是否只读
isWritable	布尔值	columnIndex: 整型	判断列 columnIndex 是否可写
isSearchable	布尔值	columnIndex: 整型	判断列 columnIndex 是否可以被搜索

4.3.5 ResultTable 接口

ResultTable 用来表示数据库查询结果, 提供方法获得结果表格中每条记录的数据。ResultTable 继承 cn.edu.celtsc.intf.common.base.Iterator, 可以依次遍历 ResultTable 中的各个记录 (行), 进而读取数据。

ResultTable 接口定义			
继承: Iterator 接口			
成员	返回值/类型	参数说明	说明
getMetaData	ResultMetaData	无	获得结果表格的原数据, 如表名等
getBoolean	布尔值	columnIndex: 整型	获得当前行列序号为 columnIndex 的布尔值
getInt	整型	columnIndex: 整型	获得当前行列序号为 columnIndex 的整型值
getDouble	双精度型	columnIndex: 整型	获得当前行列序号为 columnIndex 的双精度值
get 字符串	字符串	columnIndex: 整型	获得当前行列序号为 columnIndex 的字符串值

getDate	字符串	columnIndex: 整型	获得当前行列序号为 columnIndex 的 Date 值
getTime	字符串	columnIndex: 整型	获得当前行列序号为 columnIndex 的 Time 值
getBlob	Blob	columnIndex: 整型	获得当前行列序号为 columnIndex 的 Blob 值
getClob	Clob	columnIndex: 整型	获得当前行列序号为 columnIndex 的 Clob 值
getBytes	字节数组	columnIndex: 整型	以字节数组的形式获得当前行列序号为 columnIndex 的值
wasNull	布尔值	无	判断结果表格是否为空
isFirst	布尔值	无	判断当前位置是否在表格的第一行
isLast	布尔值	无	判断当前位置是否是表格的最后一行

4.3.6 SQLFactory 接口

SQLFactory 接口提供工厂方法用来获取到数据库的连接, 使用该连接即可进行多种数据库相关的操作, 可以通过两种方法获得一个数据库连接。

SQLFactory 接口定义			
成员	返回值/类型	参数说明	说明
getConnection	Connection	url: 字符串	根据提供的 url 字符串创建一个数据库连接
getConnection	Connection	url: 字符串 user: 字符串 password: 字符串	根据提供的 url 字符串, 用户名 user 以及密码 password 创建一个数据库连接

4.4 文件 (File Management) 服务

服务描述: 该服务提供一种静态内容的存储和检索的方法。它是文件系统与应用程序之间的一个抽象层次。该服务也可能采用单个逻辑文件的多个物理拷贝提供自动恢复的功能。

4.4.1 Directory 接口

Directory 接口表示文件系统中的目录, 目录用来组织文件, 目录中可以包含子目录和文件。该接口提供方法供在目录树的不同层次间进行移动, 如 getParent(), getDirectories()等; 该接口提供了在本目录中创建子目录和文件的方法 createSubDirectory(), createFile(); 另外该接口还支持目录的拷贝和删除操作等。

Directory 接口定义			
成员	返回值/类型	参数说明	说明
getParent	Directory	无	获得该目录的父目录, 根目录则返回空
isRoot	布尔值	无	检测该目录是否为根目录
getPath	Path	无	获得该目录对应的 Path
Copy	空	dest: Path, 目录将被拷贝到的目的路径	将本目录及其子内容拷贝到 dest 所指定的目标目录
Delete	空	无	删除该目录及其内容
createSubDirectory	Directory	dirName: 字符串, 将	在该目录下创建名称为 dirName

		要创建的子目录名称	的子目录
createFile	File	filename: 字符串, 将要创建的文件名称	在该目录下创建名称为 fileName 的文件
getDirectoryRoot	Path	无	返回该目录所属的根目录
getName	字符串	无	获得该目录名称,
getFiles	File[]	无	获取该目录所含文件的列表, 返回代表该列表的数组
getDirectories	Directory[]	无	获取该目录下所有的子目录列表, 返回代表该列表点额数组

4.4.2 File 接口

File 接口用来表示一个文件, 提供基本的面向字节的文件读写操作, 并且提供用于访问和设置文件基本信息的接口, 如读取和更改文件的创建时间等, 判断文件是否可写等; 另外文件还提供 MIME 信息, 便于在网络环境下使用。

File 接口定义			
成员	返回值/类型	参数说明	说明
copy	空	dest: Path	将该文件拷贝到 dest 所示的目录
copy	空	dest: Path newName: 字符串	将该文件拷贝到 dest 所示的目录, 新文件名称为 newName
copy	空	newName: 字符串	将文件拷贝到同一目录下, 用文件名 newName 保存
delete	空	无	删除文件
isReadable	布尔值	无	检查文件是否可读
isWritable	布尔值	无	检查文件是否可写
getMimeType	字符串	无	获取文件的 MIME
updateMimeType	空	type: 字符串	设置文件的 MIME
updateReadOnly	空	无	将文件设置为只读
updateWritable	空	无	将文件设置为可写
length	长型整数	无	返回文件的大小, 以字节为单位
read	整型	buffer: 字节数组 offset: 整型 count: 整型	从文件当前位置读取 count 个字节的内容, 读取内容放入 buffer 中 offset 起始的空间
read	整型	无	从文件当前位置读取一个字节的内 容
write	空	buffer: 字节数组	将 buffer 中的数据写入文件当前位 置
write	空	b: 字节	将字节 b 写入文件当前位置
flush	空	无	
close	空	无	关闭文件
getPath	Path	无	获得文件路径
getDirectory	Directory	无	获得文件所处的目录
getCreationTime	字符串	无	获取文件创建时间

setCreationTime	空	time: 字符串	修改文件创建时间为 time
getLastAccessTime	字符串	无	获取文件上次访问时间
setLastAccessTime	空	time: 字符串	修改文件上次访问时间为 time
getLastWriteTime	字符串	无	获取文件上次修改时间
setLastWriteTime	空	time: 字符串	修改文件上次修改时间

4.4.3 Path 接口

Path 接口表示文件或者目录的路径，可以是相对路径或者绝对路径，提供方便的方法对路径进行操作，如获取一个路径的目录名，文件名，绝对路径等。File 服务中的 File 和 Directory 接口都是通过 Path 进行标识，它们的实例都包括一个对 Path 实例的引用。

Path 接口定义			
常量定义	类型	值	说明
directorySeparatorChar	字符	"\"或"/	路径中目录间的分隔符，通常在 Windows 系统中该字符为'\'，在 UNIX 系统中为'/'
invalidPathChars	字符数组	{',', '\\', ':'}	不允许出现在路径中的字符，这些字符称为非法字符。
pathSeparator	字符	','	多个路径之间的分隔符；像 Java 中常用的环境变量 ClassPath 通常包括多个路径，用来在多个位置寻找可能的类，这些路径之间使用 pathSeparator 进行分隔，大多数系统中该字符为','。
volumeSeparator	字符	':'	卷标分隔符
成员	返回值/类型	参数说明	说明
getDirectoryName	字符串	无	获取父目录名
getName	字符串	无	获取路径名
getExtension	字符串	无	获取后缀
getFullPath	字符串	无	获取路径全名
getPath	字符串	无	获取当前路径的抽象路径
isAbsolutePath	布尔值	无	判断是否为绝对路径
isDirectory	布尔值	无	判断是否为目录
isFile	布尔值	无	判断是否为文件

4.4.4 FileManager 接口

FileManager 采用工厂化方法负责此服务中其他接口实例的创建，可以分别创建 Path，File 和 Directory 的实例。

FileManager 接口定义			
成员	返回值/类型	参数说明	说明
createPath	Path	path: 字符串，表示文件系统中的路径	
openFile	File	filePath: Path，代表将要被打开的文件位置	
createFile	File	location: Path，表示将要创建的文件的路径	
openDirectory	Directory	dirPath: Path，代表将要被打开的目录位置	
createDirectory	Directory	path: Path，将要创建的文件的路径	

4.5 标识 (Identification) 服务

服务描述：该服务负责产生一个全局或局部的用户标识。

4.5.1 Id 接口

Id 接口定义			
包含：序列化			
成员	返回值/类型	参数说明	说明
getIdString	字符串	无	获取 ID 的字符串
isEqual	布尔值	id: Id	判断两个 ID 是否相等

4.5.2 IdManager 接口

IdManager 接口定义			
成员	返回值/类型	参数说明	说明
createId	Id	无	创建一个 id 实例，如果创建过程发生错误，则抛出 CeltscException。
getId	Id	id: 字符串	通过表示 id 的字符串获得相应 id 实例，如果此 id 还没有实例创建，则创建一个实例。

4.6 日志 (Logging) 服务

服务描述：该服务可以记录其它的服务和相应的信息、事件，目的是支持整个系统的日志，从而进行系统诊断和性能分析；如安全性，数据采集，性能，倾向分析等。

4.6.1 Entry 接口

包括被写入日志的项目 (item)，以及格式类型，优先级类型和被写入日志的时间。

Entry 接口定义			
成员	返回值/类型	参数说明	说明
getFormatType	CeltsType	无	获取格式类型
getItem	可序列化对象	无	获取 Item
getPriorityType	CeltsType	无	获取优先级类型
getTimestamp	长整型数	无	获取时间戳

4.6.2 EntryIterator 接口

EntryIterator 为一组 Entry 提供顺序访问，每次只能访问一个。

EntryIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextEntry	布尔值	无	是否还有下一个 Entry
nextEntry	Entry	无	取下一个 Entry

4.6.3 ReadableLog 接口

ReadableLog 接口对日志进行读操作。

ReadableLog 接口定义			
成员	返回值/类型	参数说明	说明
getDisplayName	字符串	无	获取显示名字
getEntries	EntryIterator	formatType: CeltsType priorityType: CeltsType	以后进先出的顺序获取所有条目

4.6.4 WritableLog 接口

WritableLog 接口允许对日志的条目、类型、时间戳等进行写操作，共有两种追加日志的方法。

WritableLog 接口定义			
成员	返回值/类型	参数说明	说明
appendLog	空	entryItem: 可序列化对象	追加日志
appendLogWithTypes	空	entryItem: 可序列化对象 formatType: CeltsType priorityType: CeltsType	追加有类型的日志, 包括其格式类型以及优先级类型
assignFormatType	空	formatType: CeltsType	设置格式类型
assignPriorityType	空	priorityType: CeltsType	设置优先级类型
getDisplayName	字符串	无	获取显示名字

4.6.5 LoggingManager 接口

LoggingManager 负责创建、删除、访问日志。

logging.Entry 接口定义			
成员	返回值/类型	参数说明	说明
createLog	WritableLog	logName: 字符串	创建一个 WritableLog
deleteLog	空	logName: 字符串	删除一个日志
getFormatTypes	TypeIterator	无	获取这个实现的所有格式类型
getLogForReading	ReadableLog	logName: 字符串	为读操作打开一个日志
getLogForWriting	WritableLog	logName: 字符串	为写操作打开一个日志
getLogNamesForReading	字符串迭代器	无	获取所有可以读的日志的名字
getLogNamesForWriting	字符串迭代器	无	获取所有可以写的日志的名字
getPriorityTypes	TypeIterator	无	获取所有可用的优先级类型
supportsReading	布尔值	无	该方法返回值表明当前的实现是否支持 ReadableLog 接口和 LoggingManager 接口的方法: getLogForReading, getLogNamesForReading.

4.7 授权 (Authorization) 服务

服务描述: 该服务允许教育应用和教育服务建立和查询授权。一个授权有三个组成部分: 授权的代理, 授权代理所要完成的功能, 以及代理表现其功能所需限定的上下文。

4.7.1 Qualifier 接口

Qualifier 接口定义			
成员	返回值/类型	参数说明	说明
addParent	空	parentQualifierId: Id	为该 Qualifier 添加一个父结点
changeParent	空	oldParentId: Id,	改变该 Qualifier 的父结点

		newParentId: Id	
getChildren	QualifierIterator	无	获取该 Qualifier 的所有子结点
getDescription	字符串	无	获取该 Qualifier 的描述
getId	Id	无	获取该 Qualifier 的 Id
getParents	QualifierIterator	无	获取该 Qualifier 的所有父结点
getQualifierType	CeltsType	无	获取该 Qualifier 的限定器类型
getReferenceName	字符串	无	获取该 Qualifier 的固定引用名
isChildOf	布尔值	parentId: Id	是否是结点 parentId 的子结点
isDescendantOf	布尔值	ancestorId: Id	是否是结点 ancestorId 的子孙结点
isParent	布尔值	无	是否有子结点
removeParent	空	parentQualifierId: Id	移除一个父结点
updateDescription	空	description: 字符串	更新该 Qualifier 的描述

4.7.2 QualifierIterator 接口

QualifierIterator 为一组 Qualifier 提供顺序访问，每次只能访问一个。

QualifierIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextQualifier	布尔值	无	是否还有下一个 Qualifier
nextQualifier	Qualifier	无	取下一个 Qualifier

4.7.3 Function 接口

Function 由 Id、显示名字 (displayName)、描述 (description)、类别 (category) 和限定器类型 (QualifierType) 组成。Id 是在外部定义的，具体实现必须保证 Id 的唯一性。

Function 接口定义			
成员	返回值/类型	参数说明	说明
getDescription	字符串	无	获取该 Function 的描述
getFunctionType	CeltsType	无	获取该 Function 的类型
getId	Id	无	获取该 Function 的 Id
getReferenceName	字符串	无	获取该 Function 的固定引用名
updateDescription	空	description: 字符串	更新该 Function 的描述

4.7.4 FunctionIterator 接口

FunctionIterator 为一组 Function 提供顺序访问，每次只能访问一个。

FunctionIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextFunction	布尔值	无	是否还有下一个 Function
nextFunction	Function	无	取下一个 Function

4.7.5 Authorization 接口

Authorization 指明一个 agentId 在 Qualifier 上下文中如何执行一个 Function。

Authorization 接口定义			
成员	返回值/类型	参数说明	说明
getAgentId	Id	无	获取该 Authorization 相关联的 AgentId
getEffectiveDate	长整型数	无	获取该 Authorization 的生效日期
getExpirationDate	长整型数	无	获取该 Authorization 的失效日期
getFunction	Function	无	获取该 Authorization 的 Function

getModifiedBy	Id	无	获取修改该 Authorization 的 Id
getModifiedDate	长整型数	无	获取该 Authorization 的修改日期
getQualifier	Qualifier	无	获取该 Authorization 的 Qualifier
isActiveNow	布尔值	无	该 Authorization 当前是否生效
isExplicit	布尔值	无	该 Authorization 是否显式存储，有的授权是显式的，有些是隐式的
updateEffectiveDate	空	effectiveDate: 长整型数	更新该 Authorization 生效日期
updateExpirationDate	空	expirationDate: 长整型数	更新该 Authorization 失效日期

4.7.6 AuthorizationIterator 接口

AuthorizationIterator 为一组 Authorization 提供顺序访问，每次只能访问一个。

AuthorizationIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextAuthorization	布尔值	无	是否还有下一个 Authorization
nextAuthorization	Authorization	无	取下一个 Authorization

4.7.7 AuthorizationManager 接口

授权管理器 (AuthorizationManager) 允许一个应用程序创建授权 (Authorization)，为授权设置标准，查询授权以决定一个 agentId 在 Qualifier 上下文中如何执行一个 Function。

Authorization 中的主要对象有 Authorization、Function、agentId 和 Qualifier。其他对象还有 Function Type 和 Qualifier type。

Id 是在外部定义的，具体实现必须保证 Id 的唯一性。

有两种方法创建 Authorization。一种是使用 agentId、Function 和 Qualifier。另一种是添加生效日期 (effective date) 和失效日期 (expiration date)。如果没有生效日期和失效日期，则生效日期为当天，且永不失效。

AuthorizationManager 接口定义			
成员	返回值/类型	参数说明	说明
agentExists	boolean	agentId: Id	用户是否存在
createAuthorization	Authorization	agentId: Id, functionId: Id, qualifierId: Id	创建 Authorization
createDatedAuthorization	Authorization	agentId: Id, functionId: Id, qualifierId: Id, effectiveDate: long, expirationDate: long	创建带日期的 Authorization
createFunction	Function	functionId: Id, displayName: String, description: String, functionType: CeltsType, qualifierHierarchyId: Id	创建 Function
createQualifier	Qualifier	qualifierId: Id,	创建 Qualifier

		displayName: String, description: String, qualifierType: CeltsType, parentId: Id	
createRootQualifier	Qualifier	qualifierId: Id, displayName: String, description: String, qualifierType: CeltsType, qualifierHierarchyId: Id	创建根 Qualifier
deleteAuthorization	void	authorization : Authorization	删除 Authorization
deleteFunction	void	functionId: Id	删除 Function
deleteQualifier	void	qualifierId: Id	删除 Qualifier
getAllAZs	AuthorizationIterator	agentId: Id, functionId: Id, qualifierId: Id, isActiveNowOnly: boolean	获取所有 Authorization
getAllAZsByFuncType	AuthorizationIterator	agentId: Id, functionType: CeltsType, qualifierId: Id, isActiveNowOnly: boolean	根据函数类型获取所有 Authorization
getAllUserAZs	AuthorizationIterator	functionId: Id, qualifierId: Id, isActiveNowOnly: boolean	获取所有用户 Authorization
getAllUserAZsByFuncType	AuthorizationIterator	functionType: CeltsType, qualifierId: Id, isActiveNowOnly: boolean	根据函数类型获取所有用户 Authorization
getExplicitAZs	AuthorizationIterator	agentId: Id, functionId: Id, qualifierId: Id, isActiveNowOnly: boolean	获取所有显式 Authorization
getExplicitAZsByFuncType	AuthorizationIterator	agentId: Id, functionType: CeltsType, qualifierId: Id, isActiveNowOnly: boolean	根据函数类型获取所有显式 Authorization
getExplicitUserAZs	AuthorizationIterator	functionId: Id, qualifierId: Id, isActiveNowOnly: boolean	获取所有显式用户 Authorization
getExplicitUserAZsByFuncType	AuthorizationIterator	functionType: CeltsType, qualifierId: Id, isActiveNowOnly: boolean	根据函数类型获取所有显式用户 Authorization
getExplicitUserAZsForImplicitAZ	AuthorizationIterator	implicitAuthorization : Authorization	为隐式授权获取显式用户 Authorization

getFunction	Function	functionId: Id	获取 Function
getFunctions	FunctionIterator	functionType: CeltsType	获取所有 Function
getFunctionTypes	TypeIterator	无	获取所有函数类型
getQualifier	Qualifier	qualifierId: Id	获取所有 Qualifier
getQualifierChildren	QualifierIterator	qualifierId: Id	获取限定器子结点
getQualifierDescendants	QualifierIterator	qualifierId: Id	获取限定器子孙结点
getQualifierHierarchies	IdIterator	无	获取限定器层次结构
getQualifierTypes	TypeIterator	无	获取所有限定器类型
getRootQualifiers	QualifierIterator	qualifierHierarchyId: Id	获取所有根限定器
getWhoCanDo	IdIterator	functionId: Id, qualifierId: Id	用户是否能在某上下文执行某函数
isAuthorized	boolean	agentId: Id, functionId: Id, qualifierId: Id	是否授权
isUserAuthorized	boolean	functionId: Id, qualifierId: Id	用户是否授权
supportsDesign	boolean	无	是否支持设计, 即是否支持以下方法: createFunction, deleteFunction
supportsMaintenance	boolean	无	是否支持维护, 即是否支持以下方法: createAuthorization, createDatedAuthorization, createQualifier, createRootQualifier, deleteAuthorization, deleteQualifier, getFunctionTypes, getQualifier, getQualifierChildren, getQualifierDescendants getQualifierHierarchies, getQualifierTypes, getRootQualifiers, getWhoCanDo

4.8 workflow (Workflow) 服务

服务描述: 该服务支持学习过程的自动控制, 允许定义一组的过程规则, 并通过一定数量的参与人员和支撑系统加以执行。

4.8.1 Expression 接口

Expression 标识一个在工作步被执行之前必须满足的输入条件。

Expression 接口定义

成员	返回值/类型	参数说明	说明
getDescription	字符串	无	获取该 Expression 的描述
getDisplayName	字符串	无	获取该 Expression 的显示名字
getType	CeltsType	无	获取该 Expression 的类型

4.8.2 ExpressionIterator 接口

ExpressionIterator 为一组 Expression 提供顺序访问，每次只能访问一个。

ExpressionIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextExpression	布尔值	无	是否还有下一个 Expression
nextExpression	Expression	无	取下一个 Expression

4.8.3 Step 接口

工作步 (Step) 是进程 (Process) 中的关键元素。Processes 由若干个 Step 组成。一个 Step 包括与其他 Step 的连接、输入条件和输出状态。

Step 接口定义			
成员	返回值/类型	参数说明	说明
addPredecessor	空	stepId: Id	获取前驱 Step
getDescription	字符串	无	获取描述
getDisplayName	字符串	无	获取显示名字
getId	Id	无	获取 Id
getInputConditions	ExpressionIterator	无	获取输入条件
getOutputStates	字符串迭代器	无	获取输出状态
getPredecessors	StepIterator	无	获取所有前驱
getRoleId	Id	无	获取与此 Step 相关联的角色 Id
getSuccessors	StepIterator	无	获取所有后继
isInitial	布尔值	无	该 Step 是否为 Process 的第一个
isTerminal	布尔值	无	该 Step 是否为 Process 的最后一个
removePredecessor	空	stepId: Id	移除前驱
updateDescription	空	description: 字符串	更新描述
updateDisplayName	空	displayName: 字符串	更新显示名字
updateInputConditions	空	inputConditions: Expression[]	更新输入条件
updateOutputStates	空	outputStates: 字符串数组	更新输出状态
updateRoleId	空	roleId: Id	更新角色 Id

4.8.4 StepIterator 接口

StepIterator 为一组 Step 提供顺序访问，每次只能访问一个。

StepIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextStep	布尔值	无	是否还有下一个 Step
nextStep	Step	无	取下一个 Step

4.8.5 WorkEvent 接口

一个 WorkEvent 是 Step 的一个输出状态，与一个 Agent 和时间戳相关联。

WorkEvent 接口定义			
成员	返回值/类型	参数说明	说明
getAgentId	Id	无	获取执行该事件的用户 Id
getOutputState	字符串	无	获取输出状态
getStep	Step	无	获取 Step
getTimestamp	长整型数	无	获取该事件发生的时间戳

4.8.6 WorkEventIterator 接口

WorkEventIterator 为一组 WorkEvent 提供顺序访问，每次只能访问一个。

WorkEventIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextWorkEvent	布尔值	无	是否还有下一个 WorkEvent
nextWorkEvent	WorkEvent	无	取下一个 WorkEvent

4.8.7 Work 接口

Work 是 Step 的集合，与一个或多个角色相关联。

Work 接口定义			
成员	返回值/类型	参数说明	说明
getDescription	字符串	无	获取描述
getDisplayName	字符串	无	获取显示名字
getHistory	WorkEventIterator	无	获取历史
getId	Id	无	获取 Id
getNextSteps	StepIterator	无	获取所有角色的后继 Step
getNextStepsForRole	StepIterator	roleId: Id	获取某角色的后继 Step
getProperties	PropertiesIterator	无	获取属性
getPropertiesByType	Properties	propertiesType: CeltsType	获取某类型的属性
getPropertyTypes	TypeIterator	无	获取所有属性类型
getQualifierId	Id	无	获取限定器的 Id
updateDescription	空	description: 字符串	更新描述
updateDisplayName	空	displayName: 字符串	更新显示名字
updateStepOutputState	空	outputState: 字符串	更新输出状态

4.8.8 WorkIterator 接口

WorkIterator 为一组 Work 提供顺序访问，每次只能访问一个。

WorkIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextWork	布尔值	无	是否还有下一个 Work
nextWork	Work	无	取下一个 Work

4.8.9 Process 接口

Process 是一个有组织的 Step 集合，包括一个初始 Step 和其它后继 Step。

Process 接口定义			
---------------------	--	--	--

成员	返回值/类型	参数说明	说明
createInitialStep	Step	displayName: 字符串, description: 字符串, roleId: Id	创建初始 Step
createStep	Step	predecessorStepId: Id, displayName: 字符串, description: 字符串, roleId: Id	创建 Step
createWork	Work	displayName: 字符串, description: 字符串, qualifierId: Id	创建 Work
deleteStep	空	stepId: Id	删除 Step
deleteWork	空	workId: Id	删除 Work
getAllWork	WorkIterator	无	获取所有 Work
getAvailableWork	WorkIterator	无	获取所有有效 Work
getAvailableWorkFor Role	WorkIterator	roleId: Id	获取所有对某角色有效 Work
getAvailableWorkFor Step	WorkIterator	stepId: Id	获取所有对某 Step 有效 Work
getDescription	字符串	无	获取描述
getDisplayName	字符串	无	获取显示名字
getId	Id	无	获取 Id
getStep	Step	stepId: Id	获取 Step
getSteps	StepIterator	无	获取所有 Step
getType	CeltsType	无	获取类型
getUnfinishedWork	WorkIterator	无	获取未完成 Work
getUnfinishedWorkF orRole	WorkIterator	roleId: Id	获取某角色未完成 Work
getUnfinishedWorkF orStep	WorkIterator	stepId: Id	获取某 Step 未完成 Work
getWork	Work	workId: Id	获取 Work
haltWork	空	workId: Id	停止 Work
isEnabled	布尔值	无	Process 是否有效
resumeWork	空	workId: Id	恢复停止的 Work
updateDescription	空	description: 字符串	更新描述
updateDisplayName	空	displayName: 字符串	更新显示名字
updateEnabled	空	enabled: 布尔值	更新有效状态

4.8.10 ProcessIterator 接口

ProcessIterator 为一组 Process 提供顺序访问，每次只能访问一个。

ProcessIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextProcess	布尔值	无	是否还有下一个 Process
nextProcess	Process	无	取下一个 Process

4.8.11 WorkflowManager 接口

WorkflowManager 负责管理 Process，发现未完成的 Work。

WorkflowManager 接口定义			
成员	返回值/类型	参数说明	说明
copyProcess	Process	originalProcessId: Id, newDisplayName: 字符串, newDescription: 字符串	拷贝一个已知的 Process，并赋予新的显示名称以及描述。
createProcess	Process	displayName: 字符串, description: 字符串	创建 Process
deleteProcess	空	processId: Id	删除 Process
getCompletedWork	WorkIterator	processId: Id	获取未完成 Work
getExpressions	ExpressionIterator	无	获取所有 Expression
getExpressionsByType	ExpressionIterator	expressionType : CeltsType	获取某类型 Expression
getExpressionTypes	TypeIterator	无	获取所有 Expression 类型
getOutputStates	字符串迭代器	无	获取输出状态
getProcess	Process	processId: Id	获取 Process
getProcesses	ProcessIterator	无	获取所有 Process
getUnfinishedWork	WorkIterator	processId: Id	获取未完成 Work
supportsDesign	布尔值	无	是否支持设计，即是否支持以下方法： copyProcess, createProcess, deleteProcess
supportsMaintenance	布尔值	无	是否支持维护，即是否支持以下方法： getCompletedWork, getExpressions, getExpressionsByType, getExpressionTypes, getOutputStates, getProcess, getProcesses, getUnfinishedWork

4.9 XML 服务

服务描述：该服务提供对XML文件的访问、存储、更新等操作。

4.9.1 XMLParser 接口

XMLParser 接口负责解析 XML 数据（可能是从文件中读取或者是程序生成的一个字符串），得到 XML 文档的根，以此作为对象模型的入口即可访问文档中的所有数据。

XMLParser 接口定义			
成员	返回值/类型	参数说明	说明
getSpecification	字符串	无	返回该解析器所对应的规范，如 ims-md, ims-cp 等

parse	对象	docString: 字符串 将被解析的文件路径, 可以是相对或者绝对路径	返回被解析文档的入口对象, 可以根据该解析器对应的规范转型为特定的对象
parse	对象	file: File 将被解析的文件实例	返回被解析文档的入口对象, 可以根据该解析器对应的规范转型为特定的对象
parse	对象	path: Path 将被解析的文件路径	返回被解析文档的入口对象, 可以根据该解析器对应的规范转型为特定的对象

4.9.2 XMLFormatter 接口

XMLFormatter 接口用于对 XML 数据进行简单的排版。

XMLParser 接口定义			
成员	返回值/类型	参数说明	说明
format	字符串	xmlStr: 字符串	对输入的 XML 数据字符串进行格式化, 返回结果字符串
format	空	xmlStr: 字符串 filePath: Path	对输入的 XML 字符串进行格式化, 格式化后的结果存入 filePath 指定文件

4.9.3 XMLFactory 接口

通过 XMLFactory 可以获得 XMLParser 和 XMLFormatter 的实例。

XMLFactory 接口定义			
成员	返回值/类型	参数说明	说明
getParser	XMLParser	specification: 字符串	获得一个特定规范的 XMLParser, 如 ims 元数据规范的 Parser
getFormatter	XMLFormatter	无	获取 s 一个 XMLFormatter 实例

5 教育服务接口

5.1 评价 (assessment) 服务

服务描述: 该服务负责显示和汇报一个评估, 如小测验或者考试, 这种评估的显示和汇报可以在组或者个人的级别上进行管理。

5.1.1 Item 接口

Item 包括问题、回复集、答案以及其它相关指导和媒体。这些元素都被包含在一个 Data 对象中, 该对象的内容没有在接口规范中确切说明。Item 由其对应的类型来区分。ItemType 对一个应用程序来说是有意义的, 但在接口规范中没有确切定义。与 Item 相关联的全局唯一的 Id 由 AssessmentManager 的 createItem 方法设置。

Item 接口定义			
成员	返回值/类型	参数说明	说明
getData	可序列化对象	无	获取该 Item 的 Data
getDescription	字符串	无	获取该 Item 的描述
getDisplayName	字符串	无	获取该 Item 的显示名字
getId	Id	无	获取该 Item 的 Id
getItemType	CeltsType	无	获取该 Item 的类型 ItemType

getProperties	PropertiesIterator	无	获取该 Item 的所有属性配置
getPropertiesByType	Properties	propertiesType: CeltsType	获取该 Item 的某种类型的属性配置
getPropertyTypes	TypeIterator	无	获取该 Item 的所有属性类型
updateData	空	data: 可序列化对象	更新该 Item 的 Data
updateDescription	空	description : 字符串	更新该 Item 的描述
updateDisplayName	空	displayName: 字符串	更新该 Item 的显示名字

5.1.2 ItemIterator 接口

ItemIterator 为一组 Item 提供顺序访问，每次只能访问一个。所有迭代器的目的就是在接口规范中为通用类型返回多个值，而不使用数组。如果返回值数据量很大，或者是远程访问，返回数组是不合适的。迭代器不允许使用序号（index）访问值，而只允许顺序访问。也就是说不能回头访问曾经访问过的数据。

ItemIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextItem	布尔值	无	是否还有下一个 Item
nextItem	Item	无	取下一个 Item

5.1.3 ItemTaken 接口

ItemTaken包含Item、提交的回答以及这些回答的Evaluation。

ItemTaken 接口定义			
成员	返回值/类型	参数说明	说明
createEvaluation	无	evaluationType: CeltsType	为该 Item 按照指定的 Type 创建 Evaluation
deleteEvaluation	无	evaluationId:Id	删除指定的 Evaluation
getData	可序列化对象	无	获取 ItemTaken 的数据
getDisplayName	字符串	无	获取 ItemTaken 的显示名称
getEvaluations	EvaluationIterator	无	获取该 Item 的所有 Evaluation
getEvaluationByType	EvaluationIterator	evaluationType: CeltsType	获取指定类型的 Evaluation
getId	Id	无	获取该 ItemTaken 的唯一 Id
getItem	Item	无	获取 ItemTaken 所属的 Item
getSectionTaken	SectionTaken	无	获取 ItemTaken 所属的 SectionTaken
updateData	无	data:可序列化对象	为 ItemTaken 更新数据
updateDisplayName	无	displayNames:字符串	为 ItemTaken 更新显示名称

5.1.4 ItemTakenIterator 接口

ItemTakenIterator为一组ItemTaken提供顺序访问，每次只能访问一个

EvaluationIterator 接口定义			
-------------------------	--	--	--

成员	返回值/类型	参数说明	说明
hasNextItemTaken	布尔值	无	是否还有下一个 ItemTaken
nextItemTaken	ItemTaken	无	返回下一个 ItemTaken

5.1.5 Evaluation 接口

Evaluation是评估报告的接口。

Evaluation 接口定义			
成员	返回值/类型	参数说明	说明
getData	可序列化对象	无	从 Evaluation 中取得数据
getDisplayName	字符串	无	获取 Evaluation 的显示名字
getId	Id	无	从 Evaluation 中获得唯一 Id
getModifiedDate	长整数	无	获取 Evaluation 的修改时间
getObjectTaken	Id	无	获取 objectTaken 的 Id
getType	CeltsType	无	获取 Evaluation 的 CeltsType
updateData	无	data:可序列化对象	更新 Evaluation 的数据
updateDisplayName	无	displayName:字符串	更新 Evaluation 的显示名字

5.1.6 EvaluationIterator 接口

EvaluationIterator为一组Evaluation提供顺序访问，每次只能访问一个。

EvaluationIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextEvaluation	布尔值	无	是否还有下一个 Evaluation
nextEvaluation	Evaluation	无	返回下一个 Evaluation

5.1.7 Section 接口

Section 包括 0 个或多个 Item 或 Section。Item 或 Section 按照先入先出(FIFO)方式加入一个 Section。SectionType 对一个应用程序来说是有意义的，但在接口规范中没有确切定义。与 Section 相关联的全局唯一的 Id 由 AssessmentManager 的 createSection 方法设置。

Section 接口定义			
成员	返回值/类型	参数说明	说明
addItem	空	item: Item	向该 Section 中添加一个 Item
addSection	空	section : Section	向该 Section 中添加一个 Section
getData	可序列化对象	无	获取该 Section 的 Data
getDescription	字符串	无	获取该 Section 的描述
getDisplayName	字符串	无	获取该 Section 的显示名字
getId	Id	无	获取该 Section 的 Id
getItems	ItemIterator	无	获取该 Section 中所有 Item
getProperties	PropertiesIterator	无	获取该 Section 的所有属性配置
getPropertiesByType	Properties	propertiesType : CeltsType	获取该 Section 的某种类型的属性配置
getPropertyTypes	TypeIterator	无	获取该 Section 的所有属性类型
getSections	SectionIterator	无	获取该 Section 中所有 Section

getSectionType	CeltsType	无	获取该 Section 的类型
orderItems	空	items : Item[]	改变该 Section 中 Item 的顺序
orderSections	空	sections : Section[]	改变该 Section 中 Section 的顺序
removeItem	空	itemId: Id	从该 Section 中删除一个 Item
removeSection	空	sectionId: Id	从该 Section 中删除一个 Section
updateData	空	data: 可序列化对象	更新该 Section 的 Data
updateDescription	空	description: 字符串	更新该 Section 的描述
updateDisplayName	空	displayName: 字符串	更新该 Section 的显示名字

5.1.8 SectionIterator 接口

SectionIterator 为一组 Section 提供顺序访问，每次只能访问一个。

SectionIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextSection	布尔值	无	是否还有下一个 Section
nextSection	Section	无	取下一个 Section

5.1.9 SectionTaken 接口

SectionTaken 含有 Section 里面的所有 Item 以及 Evaluation。

SectionTaken 接口定义			
成员	返回值/类型	参数说明	说明
createEvaluation	Evaluation	evaluationType: CeltsType	按照指定的 CeltsType 为 Section 创建 Evaluation
createItemTaken	ItemTaken	item:Item	在 SectionTaken 的基础上创建 ItemTaken
deleteEvaluation	无	evaluationId: Id	在 Section 中删除指定的 Evaluation
deleteItemTaken	无	itemTakenId: Id	删除一个 ItemTaken
getAssessmentTaken	AssessmentTaken	无	获取 SectionTaken 所属的 AssessmentTaken
getData	可序列化对象	无	获取 SectionTaken 的数据
getDisplayName	字符串	无	获取 SectionTaken 的显示名字
getEvaluations	EvaluationIterator	无	获取 Section 中所有的 Evaluation
getEvaluationsByType	EvaluationIterator	evaluationType: CeltsType	获取 Section 中指定 CeltsType 的 Evaluation
getId	Id	无	获取 SectionTaken 的唯一 Id
getItemTaken	ItemTakenIterator	无	获取该 SectionTaken 的所有 ItemTaken
getSection	Section	无	获得创建 SectionTaken 的 Section
updateData	无	data:可序列化对象	更新 SectionTaken 的数据

updateDisplayName	无	displayName: e:字符串	更新 SectionTaken 的显示名称
-------------------	---	-----------------------	-----------------------

5.1.10 SectionTakenIterator 接口

SectionTakenIterator为一组SectionTaken提供顺序访问，每次只能访问一个。

SectionTakenIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextSectionTaken	布尔值	无	是否还有下一个 SectionTaken
nextSectionTaken	SectionTaken	无	取下一个 SectionTaken

5.1.11 Assessment 接口

Assessment 包括 0 个或多个 Section。Section 按照先入先出（FIFO）方式加入一个 Assessment。AssessmentType 对一个应用程序来说是有意义的，但在接口规范中没有确切定义。Assessment 还包括一个 Topic。一个 Assessment 与一个 AssessmentPublished 相关联，该 AssessmentPublished 包括一个 Assessment 和与这个 Assessment 的有效性相关的数据集。一个 Assessment 还与一个 AssessmentTaken 相关联，该 AssessmentTaken 包括一个 Assessment 和与该 Assessment 集成的学生相关的数据集。

Assessment 接口定义			
成员	返回值/类型	参数说明	说明
addSection	空	section : Section	向该 Assessment 中添加一个 Section
getAssessmentType	CeltsType	无	获取该 Assessment 的类型
getData	可序列化对象	无	获取该 Assessment 的 Data
getDescription	字符串	无	获取该 Assessment 的描述
getDisplayName	字符串	无	获取该 Assessment 的显示名字
getId	Id	无	获取该 Assessment 的 Id
getItems	ItemIterator	无	获取该 Section 中所有 Item
getProperties	PropertiesIterator	无	获取该 Assessment 的所有属性配置
getPropertiesByType	Properties	propertiesType : CeltsType	获取该 Assessment 的某种类型的属性配置
getPropertyTypes	TypeIterator	无	获取该 Assessment 的所有属性类型
getSections	SectionIterator	无	获取该 Assessment 中所有 Section
getTopic	字符串	无	获取该 Assessment 的 Topic
orderSections	空	sections : Section[]	改变该 Assessment 中 Section 的顺序
removeSection	空	sectionId: Id	从该 Assessment 中删除一个 Section
updateData	空	data: 可序列化对象	更新该 Assessment 的 Data
updateDescription	空	description: 字符串	更新该 Assessment 的描述
updateDisplayName	空	displayName: e: 字符串	更新该 Assessment 的显示名字
updateTopic	空	topic: 字符串	更新该 Assessment 的 Topic

5.1.12 AssessmentIterator 接口

AssessmentIterator 为一组 Assessment 提供顺序访问，每次只能访问一个。

AssessmentIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextAssessment	布尔值	无	是否还有下一个 Assessment
nextAssessment	Assessment	无	取下一个 Assessment

5.1.13 AssessmentPublished 接口

AssessmentPublished 接口定义			
成员	返回值/类型	参数说明	说明
createAssessmentTaken	AssessmentTaken	agentId: Id	创建一个 AssessmentPublished
deleteAssessmentTaken	空	assessmentTakenId: Id	删除一个 AssessmentPublished
getAssessment	Assessment	无	获取要发布的 Assessment
getAssessmentsTaken	AssessmentTakenIterator	无	获取所有的 AssessmentTaken
getAssessmentsTakenBy	AssessmentTakenIterator	agentId: Id	获取所有与某个学生的 agentId 相关联的 Assessment
getCourseSectionId	Id	无	获取与该 AssessmentPublished 相关联的 CourseSection 的 Id
getData	可序列化对象	无	获取该 AssessmentPublished 的 Data
getDate	长整型数	无	获取该 AssessmentPublished 的发布日期
getDescription	字符串	无	获取该 AssessmentPublished 的描述
getDisplayName	字符串	无	获取该 AssessmentPublished 的显示名字
getId	Id	无	获取该 AssessmentPublished 的 Id
updateCourseSectionId	空	courseSectionId: Id	更新该 AssessmentPublished 的 CourseSectionId
updateData	空	data: 可序列化对象	更新该 AssessmentPublished 的 Data
updateDescription	空	description: 字符串	更新该 AssessmentPublished 的描述
updateDisplayName	空	displayName: 字符串	更新该 AssessmentPublished 的显示名字

5.1.14 AssessmentPublishedIterator 接口

AssessmentPublishedIterator 为一组 AssessmentPublished 提供顺序访问，每次只能访问一个。

AssessmentPublishedIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextAssessmentPublished	布尔值	无	是否还有下一个 AssessmentPublished
nextAssessmentPublished	AssessmentPublished	无	取下一个 AssessmentPublished

5.1.15 AssessmentTaken 接口

AssessmentTaken 接口定义			
成员	返回值/类型	参数说明	说明
createEvaluation	Evaluation	evaluationType : CeltsType	为该 Assessment 创建一个特定类型的评估报告
createSectionTaken	SectionTaken	section : Section	在一个 AssessmentPublished 和学生的 AgentId 的基础上创建一个 SectionTaken
deleteEvaluation	空	evaluationId : Id	删除一个评估报告
deleteSectionTaken	空	sectionTake nId: Id	删除一个 SectionTaken
getAgentId	Id	无	获取接受评价的学生 agentId
getAssessmentPublished	AssessmentPublished	无	获取采用的 AssessmentPublished
getData	可序列化对象	无	获取该 AssessmentTaken 的 Data
getDate	长整型数	无	获取该 AssessmentTaken 的执行日期
getDisplayName	字符串	无	获取该 AssessmentTaken 的显示名字
getEvaluations	EvaluationIterator	无	获取该 AssessmentTaken 的所有评估报告
getEvaluationsByType	EvaluationIterator	evaluationType : CeltsType	获取该 AssessmentTaken 的某种类型的评估报告
getId	Id	无	获取该 AssessmentTaken 的 Id
getSectionsTaken	SectionTakenIterator	无	获取该 AssessmentTaken 的所有 SectionsTaken
updateData	空	data: 可序列化对象	更新该 AssessmentTaken 的 Data
updateDisplayName	空	displayName: 字符串	更新该 AssessmentTaken 的显示名字

5.1.16 AssessmentTakenIterator 接口

AssessmentTakenIterator 为一组 AssessmentTaken 提供顺序访问，每次只能访问一个。

AssessmentTakenIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextAssessmentTaken	布尔值	无	是否还有下一个 AssessmentTaken
nextAssessmentTaken	AssessmentTaken	无	取下一个 AssessmentTaken

5.1.17 AssessmentManager 接口

AssessmentManager 主要负责创建和删除 Assessment、AssessmentPublished、Item 和 Section。以及获取 Assessment、AssessmentPublished、Item、Section 和各种实现类型的值。

AssessmentManager 接口定义			
成员	返回值/类型	参数说明	说明
createAssessment	Assessment	name: 字符串, description: 字	创建一个新的 Assessment 并将其加入 Assessment 库

		字符串, assessmentType: CeltsType	
createAssessmentPublished	AssessmentPublished	assessment : Assessment	根据 Assessment 创建一个 AssessmentPublished
createItem	Item	name: 字符串, description: 字符串, itemType : CeltsType	创建一个 Item
createSection	Section	name: 字符串, description: 字符串, sectionType : CeltsType	创建一个 Section
deleteAssessment	空	assessmentId: Id	从 Assessment 库中删除一个 Assessment
deleteAssessmentPublished	空	assessmentPublishedId: Id	删除一个 AssessmentPublished
deleteItem	空	itemId: Id	删除一个 Item
deleteSection	空	sectionId: Id	删除一个 Section
getAssessment	Assessment	assessmentId: Id	根据 Id 获取一个 Assessment
getAssessmentPublished	AssessmentPublished	assessmentPublishedId: Id	根据 Id 获取一个 AssessmentPublished
getAssessments	AssessmentIterator	无	获取所有的 Assessment
getAssessmentsByType	AssessmentIterator	assessmentType: CeltsType	获取特定类型的 Assessment
getAssessmentsPublished	AssessmentPublishedIterator	无	获取所有的 AssessmentPublished
getAssessmentTypes	TypeIterator	无	获取所有的 Assessment 类型
getItem	Item	itemId: Id	根据 Id 获取一个 Item
getItems	ItemIterator	无	获取所有的 Item
getItemsByType	ItemIterator	itemType : CeltsType	获取特定类型的 Item
getItemTypes	TypeIterator	无	获取所有的 Item 类型
getSection	Section	sectionId: Id	根据 Id 获取一个 Section
getSections	SectionIterator	无	获取所有的 Section
getSectionsByType	SectionIterator	sectionType : CeltsType	获取特定类型的 Section
getSectionTypes	TypeIterator	无	获取所有的 Section 类型

5.2 课程管理 (Course Management) 服务

服务描述：该服务负责管理课程相关的内容，包括创建课程，课程的存储，查询等。

5.2.1 Course 接口

Course 接口是课程模型的基本接口，按照不同的标准需求可以以此接口进行扩展，Course 接口所提供的是课程最一般化的信息，更具体化的课程在 CourseOffering 接口中提供。

Course 接口定义			
成员	返回值/类型	参数说明	说明
getCredits	浮点数类型	无	获得课程学分数
setCredits	空	credits: 浮点数类型	设置课程学分
getCourseType	CeltsType	无	获得课程类型
setCourseType	空	courseType: CeltsType	设置课程类型
getDescription	字符串	无	获得课程的基本描述
setDescription	空	description: 字符串	设置课程的描述信息
getDisplayName	字符串	无	获得课程的显示名称
setDisplayName	空	displayName: 字符串	设置课程的显示名称
getId	Id	无	获得课程 ID
getProperties	PropertiesIterator	无	获得课程相关的配置属性
getStatus	CeltsType	无	获得课程的状态
setStatus	空	statusType: CeltsType	设置课程状态
getTitle	字符串	无	获得课程标题
setTitle	空	title: 字符串	设置课程标题

5.2.2 CourseOffering 接口

Course 所提供的课程是十分泛化的课程，而 CourseOffering 则是针对某个具体学期开设的具体一门课程，对于一个 Course 可以有若干个 CourseOffering，每个 CourseOffering 有自己的名册等信息并有与之相关联的课程章节内容，课程的片断模型由 CourseSection 接口定义。

CourseOffering 接口定义			
成员	返回值/类型	参数说明	说明
addAsset	空	assetId: Id	向一门课程添加素材
removeAsset	空	assetId: Id	从课程中删除一个素材
getAssets	IdIterator	无	获得课程的全部素材
addStudent	空	studentId: Id regType: CeltsType	向名册中添加一个学生，并为其注册类型赋值。
removeStudent	空	studentId: Id	从该课程注册的学生名册中删除一个学生
changeRegType	空	studentId: Id regType: CeltsType	变更一个学生的注册类型
getRoster	Iterator	无	得到课程的学生名册
getCourse	Course	无	得到包括此 CourseOffering 的课程实例
getId	Id	无	得到当前 CourseOffering 的 ID
setDescription	空	description: 字符串	设置描述信息
getDescription	字符串	无	获得描述信息
getDisplayName	字符串	无	获得显示名称
setDisplayName	空	displayName: 字符	设置显示名称

		串	
getProperties	PropertiesIterator	无	获得配置信息
getStatus	CeltsType	无	获得 CourseOffering 的状态
setStatus	空	statusType : CeltsType	设置 CourseOffering 状态
getTitle	字符串	无	获得 CourseOffering 标题
setTitle	空	title: 字符串	设置 CourseOffering 标题

5.2.3 CourseManager 接口

CourseManager 负责创建删除课程，并且可获得课程成绩的管理及分组管理。

CourseManager 接口定义			
成员	返回值/类型	参数说明	说明
createCourse	Course	title: 字符串 number: 字符串 description: 字符串 courseType: CeltsType courseStatusType : CeltsType credits: 浮点数类型	创建一门课程，包括课程标题，课容量，描述，课程和状态类型，学分数。返回一个课程实例，拥有课程名、描述和统一的 ID。
deleteCourse	空	courseId: Id	根据 ID 删除一门课程

5.3 成员管理 (Membership Management) 服务

服务描述：该服务提供成员的管理，一个组可以是另一个组的成员，成员信息描述了成员承担的角色。

5.3.1 Member 接口

Member 是 eLearning 中虚拟成员的抽象，包括的属性有 DisplayName, Id, CeltsType 和 Properties, Member 通过 MemberManager 统一创建。

Member 接口定义			
成员	返回值/类型	参数说明	说明
getDisplayName	字符串	无	获得成员显示的名称
getId	Id	无	获得成员 ID
getType	CeltsType	无	获得成员类型
getPropertiesByType	Properties	propertiesType : CeltsType	获得同类型的所有属性
getPropertyTypes	TypeIterator	无	获得属性的类型
getProperties	PropertiesIterator	无	获得属性

5.3.2 MemberIterator 接口

MemberIterator 接口用于 Member 的查询等操作，是获取一组成员对象的迭代器接口。

MemberIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextMember	布尔值	无	判断是否集合中是否还有成员

nextMember	Member	无	获得下一个成员实例
------------	--------	---	-----------

5.3.3 MemberManager 接口

MemberManager 接口提供创建，删除成员，以及得到成员实例的功能。

MemberManager 接口定义			
成员	返回值/类型	参数说明	说明
createMember	Member	displayName: 字符串 memberType: CeltsType properties: Properties	通过 display name, type, properties 创建一个成员
deleteMember	空	id: Id	通过 ID 删除指定成员
getMember	Member	id: Id	根据唯一的id获得相应的成员实例，由于其它的属性并不唯一，我们不提供其它获取实例的方式
getMembers	MemberIterator	无	通过迭代器获得全部成员
getMemberTypes	TypeIterator	无	获得成员的所有类型
getPropertyTypes	TypeIterator	无	获得所有属性类型
getMembersByType	MemberIterator	memberType: CeltsType	获得指定类型的所有成员实例
getMemberSearchTypes	TypeIterator	无	获得全部成员的查询类型
getMembersBySearch	MemberIterator	searchCriteria: 可序列化对象 memberSearchType : CeltsType	按查询类型和查询标准检索成员

5.4 组管理 (Group Management) 服务

服务描述：该服务可以对组进行管理，一个组可以包含其它的组，也可以有自己的子组。

5.4.1 Group 接口

组严格意义上说也可以看作一种成员，在 OSID 的早期版本内，组合成员是相同对待的，但 v3 的新近更新意见是将组与成员分离开来，以便于在其它的地方使用分组。组可以包括成员，也可以有子组，因此组接口中包括一些测试方法，验证成员和组的包含关系。

Group 接口定义			
成员	返回值/类型	参数说明	说明
setDescription	空	description: 字符串	设置组描述，description 是组特有的描述信息
getDescription	字符串	无	获得组描述信息
getId	Id	无	获得唯一的组 id
getDisplayName	字符串	无	获得显示名称

getType	CeltsType	无	获得类型
add	空	memberOrGroup: Member	添加一个结点，可以是成员或者子组
remove	空	memberOrGroup: Member	删除一个结点
getMembers	MemberIterator	includeSubgroups: 布尔值	获得组的全部成员实例，输入参数可以确定是否递归查找子组
getGroups	GroupIterator	includeSubgroups: 布尔值	获得全部子组，输入参数可以确定是否递归查找子组
contains	布尔值	memberOrGroup: Member searchSubgroups: 布尔值	验证一个组或成员是否属于当前的组
getPropertiesByType	Properties	propertiesType: CeltsType	获得同类型的所有属性
getPropertyTypes	TypeIterator	无	获得所有类型
getProperties	PropertiesIterator	无	获得所有属性

5.4.2 GroupIterator 接口

GroupIterator 接口用于 Member 的查询等操作，是获取一组成员对象的迭代器类接口。

GroupIterator 接口定义			
成员	返回值/类型	参数说明	说明
hasNextGroup	布尔值	无	判断是否集合中是否还有组
nextGroup	Group	无	获得下一个组实例

5.4.3 GroupManager 接口

GroupManager 接口提供创建，删除组，以及得到组实例的功能。

GroupManager 接口定义			
成员	返回值/类型	参数说明	说明
createGroup	Group	displayName: 字符串 groupType: CeltsType description: 字符串 properties: Properties	通过 display name, type, description, properties 创建一个组
deleteGroup	空	id: Id	通过 ID 删除指定组
getGroup	Group	id: Id	根据唯一的 id 获得相应的组实例
getGroups	GroupIterator	无	通过迭代器获得全部组
getGroupTypes	TypeIterator	无	获得组的所有类型
getPropertyTypes	TypeIterator	无	获得所有属性类型
getGroupsByType	GroupIterator	groupType: CeltsType	获得指定类型的所有组实例
getGroupSearchTypes	TypeIterator	无	获得全部组的查询类型
getGroupsBySearch	GroupIterator	searchCriteria: 可序列化对象	按查询类型和查询标准检索组

		groupSearchType CeltsType	:	
--	--	------------------------------	---	--

5.5 编列 (sequencing) 服务

服务描述: 该服务可以让一系列学习对象按照特定的序列执行, 可能的序列集合由相应的编列规则加以定义。

5.5.1 SequenceRule 接口

SequenceRule 接口是编列规则的基本数据模型, 其中给出的是对一个编列活动的评价函数, 具体的编列规则是由规范所确定的, 此处编列规则的模型仅仅是对上层开发的应用接口。

SequenceRule 接口定义			
成员	返回值/类型	参数说明	说明
evaluate	字符串	iType: CeltsType iThisActivity: SequenceActivity iRetry: 布尔值	根据编列活动, 类型参数, 获得一个编列规则的评价结果。

5.5.2 SequenceActivity 接口

SequenceActivity 是编列活动接口, 也是编列活动的基本数据模型, 此接口提供类似活动树结点结构的管理操作。

SequenceActivity 接口定义			
成员	返回值/类型	参数说明	说明
addChild	空	ioChild: SequenceActivity	添加一个孩子结点
setChildren	空	ioChildren: 向量数组 iAll: 布尔值	为全部孩子结点设置
getChildren	向量数组	iAll: 布尔值	得到当前结点的全部孩子结点
hasChildren	布尔值	iAll: 布尔值	判断当前结点是否拥有子结点
getNextSibling	SequenceActivity	iAll: 布尔值	获得直接后继兄弟活动结点
getPrevSibling	SequenceActivity	iAll: 布尔值	获得直接前驱兄弟活动结点
getParent	SequenceActivity	无	获得父活动结点
getParentID	Id	无	获得父活动结点 ID
getActiveOrder	整型	无	得到相关的兄弟结点的活动种类
getObjStatusSet	向量数组	无	获得对象状态集
setDepth	空	iDepth: 整型	设置活动深度
getDepth	整型	无	获得活动深度
setActivityCount	空	iCount: 整型	设置活动计数
getActivityCount	整型	无	获得活动计数
setSelection	空	iSelection: 布尔值	设置当前活动的选中状态
isSelected	布尔值	无	判断当前活动是否被选

			中
--	--	--	---

5.5.3 SequenceActivityTree 接口

SequenceActivityTree 接口是编列活动基础上的树型结构模型，管理整个编列活动树。

SequenceActivityTree 接口定义			
成员	返回值/类型	参数说明	说明
setRoot	空	iRoot: SequenceActivity	将一个活动设置为活动树的根结点
getRoot	SequenceActivity	无	获得活动树的根结点活动
setLastLeaf	空	iLastLeaf: 字符串	设置最后一个叶结点
getLastLeaf	字符串	无	获得最后一个叶结点
setCurrentActivity	空	iCurrent: SequenceActivity	设置当前活动结点
getCurrentActivity	SequenceActivity	无	获得当前活动结点
setFirstCandidate	空	iFirst: SequenceActivity	设置第一个候选活动
getFirstCandidate	SequenceActivity	无	获得第一个候选活动
setSuspendAll	空	iSuspendTarget : SequenceActivity	将活动设置为 SuspendAll 状态
getSuspendAll	SequenceActivity	无	获得挂起状态 SuspendAll 的活动结点
getActivity	SequenceActivity	iActivityID: Id	获得指定 ID 的结点活动
getObjMap	向量数组	obj: 字符串	获得指定学习目标的活动集
getGlobalObjectives	向量数组	无	获得学习目标集
clearSessionState	空	无	清除目前活动树的状态
setDepths	空	无	计算当前树的深度
setTreeCount	空	无	设置当前活动树的活动计数

5.5.4 Sequencer 接口

Sequencer 接口是编列引擎的基本操作，是联系用户与具体活动树的部件。

Sequencer 接口定义			
成员	返回值/类型	参数说明	说明
getObjStatusSet	IdIterator	activityId: Id	获得指定 ID 活动的学习目标状态集
setActivityTree	空	iTree : SequenceActivityTree	设置编列活动树
getActivityTree	SequenceActivityTree	无	获得编列活动树
clearSequenceState	空	无	清除编列引擎状态参数

reportSuspension	空	activityId: 字符串 iSuspended: 布尔值	告知编列引擎当前活动成为挂起状态, 当此活动结束后, 此状态生效
setAttemptDuration	空	activityId: Id iDur: 字符串	告知编列引擎此活动的尝试持续时间改变
setAttemptObjSatisfied	空	activityId: Id objectiveId: Id iStatus: 字符串	告知编列引擎一个活动目标的满足状态改变
setAttemptProgressStatus	空	activityId: Id iProgress: 字符串	告知编列引擎编列活动的过程状态改变

5.5.5 SequenceNavigation 接口

SequenceNavigation 是编列导航接口, 对应于接收到 IMS SS 中的导航请求时所做的导航行为。

SequenceNavigation 接口定义			
成员	返回值/类型	参数说明	说明
navigate	空	iRequest: 整型	处理导航请求
navigate	空	iTarget: 字符串	处理"Choice"导航请求

5.5.6 SequenceParser 接口

SequenceParser 是编列解析器接口, 其类似 XML 的解析器, 直接用于解析包含编列信息的 XML 文件, 并提供获得相应结点的方法。现有的编列引擎应该支持 CELTSC 和 IMS Simple Sequence 两种规范。在此接口中使用了 DOM 中的 Node 类, 在未来的版本中期望这一操作被隔离出去。

SequenceParser 接口定义			
成员	返回值/类型	参数说明	说明
initParser	空	seqFileName: 字符串	初始化编列信息解析器
parseFile	布尔值	无	解析文件, 解析出错则返回 false, 正确解析则为 true
getCourseID	Id	无	获得 Course 的 ID
getOrganizationID	Id	无	获得 Organization 的 ID
getScopeID	Id	无	获得 Scope 的 ID
findDefaultOrganization	Node	无	找到 DefaultOrganization 的结点
getSequencingCollection	Node	无	获得编列集的结点
getAttribute	字符串	iNode: Node attributeName: 字符串	得到 XML 结点的属性值

5.6 学习对象管理 (Learning Object Management) 服务

服务描述: 该服务提供对学习对象的管理, 包括学习对象的导入, 导出, 存储, 检索等功能。

5.6.1 LangString 接口

LangString 是多语言字符串接口。多语言字符串容器用于满足标准中多语言字符串可以一次出现多

个的情况。如标准所示：一个多语言字符串的值可以包含多个语义相同的字符串，如不同的翻译或选择性的描述。

LangString 接口定义			
成员	返回值/类型	参数说明	说明
getLang	字符串	无	获得字符串语种
getContent	字符串	无	获得字符串内容
setLang	空	Lang: 字符串	设置字符串语种
setContent	空	Content: 字符串	设置字符串内容

5.6.2 LangStringContainer 接口

LangStringContainer 是多语言字符串容器接口。多语言字符串容器应用在很多元素中，因此避免了大量的重复接口定义。很多元数据元素即可简单的以该容器子接口的形式出现即可。在应用中，只需要指定多语言容器的名称，即“NAME_”属性，就可以确定具体是哪个元素。

LangStringContainer 接口定义			
常量	类型	说明	
NAME_TITLE	字符串	用于标题元素	
NAME_KEYWORD	字符串	用于关键字元素	
NAME_DESCRIPTION	字符串	用于描述元素	
NAME_COVERAGE	字符串	用于覆盖区元素	
NAME_TYPICAL_AGE_RANGE	字符串	用于典型年龄范围元素	
NAME_VERSION	字符串	用于版本元素	
NAME_INSTALLATIONREMARKS	字符串	用于安装描述元素	
NAME_OTHERPLATFORMREQUIREMENTS	字符串	用于其他平台要求元素	
NAME_TAXONENTRY	字符串	用于分类单元条目元素	
NAME_TAXONSOURCE	字符串	用于分类来源元素	
成员	返回值/类型	参数说明	说明
getName	字符串	空	获取元素类型名称
getLangStrings	LangString[]	空	获得所有字符串
setLangStrings	空	langStrs: LangString[]	设置所有字符串
getByLang	LangString	lang: 字符串	获得规定语种的字符串
emptyLangString	空	空	清空所有字符串
getContentInString	字符串	空	获得所有字符串内容
contains	布尔值	content: 字符串	是否包含指定字符串

表 5-1 包含了所有的继承了该接口的字符串容器，它们都是简单的继承该接口，没有增加其他成员、方法。

表5-1 使用多语言字符串容器的元素列表

中文元素名	英文元素名
1.2:通用.标题	1.2:General.Title
1.4:通用.描述	1.4:General.Description
1.5:通用.关键字	1.5:General.Keyword

1.6:通用.覆盖区	1.6:General.Coverage
2.1:生命期.版本	2.1:LifeCycle.Version
4.5:技术.安装描述	4.5:Technical.InstallationRemarks
4.6:技术.其他平台要求	4.6:Technical.OtherPlatformRequirements
5.7:教育.典型年龄范围	5.7:Educational.TypicalAgeRange
9.2.1:分类.分类路径.来源	9.2.1:Classification.TaxonPath.Source
9.2.2.2:分类.分类路径.分类单元.条目	9.2.2.2:Classification.TaxonPath.Taxon.Entry

5.6.3 SourceValue 接口

SourceValue 是词汇表接口。词汇表学习对象元数据标准中的基本数据类型，也是标准推荐使用的取值列表，以“词汇表”作为数据类型的数据元素的值是一个二元组（来源、值）。

SourceValue 接口定义			
常量	类型	说明	
CELTS_SOURCE	字符串数组	CELTS 标准类型的词汇表	
成员	返回值/类型	参数说明	说明
getName	字符串	空	获取元素类型名称
getLangStrings	LangString[]	空	获得所有字符串
setLangStrings	空	langStrs: LangString[]	设置所有字符串
getByLang	LangString	lang: 字符串	获得规定语种的字符串
emptyLangString	空	空	清空所有字符串
getContentInString	字符串	空	获得所有字符串内容
contains	布尔值	content: 字符串	是否包含指定字符串

由于在学习对象元数据标准中有很多的词汇表类型元素，因此与多语言字符串容器类似，词汇表作为学习对象元数据标准的一个基本数据结构，被很多元素所继承，用来代表具体的词汇表型数据元素。通过调用 getName 方法来判断当前词汇表的具体类型。表 5-2 展示了所有的词汇表型数据元素。

表5-2 词汇表型数据元素

中文元素名	英文元素名
1.7:通用.结构	1.8:General.Structure
1.8:通用.聚合度	1.8:General.AggregationLevel
2.2:生命期.状态	2.2:LifeCycle.Status
2.3.1:生命期.贡献.角色	2.3.1:LifeCycle.Contribute.Role
3.2.1:元-元数据.贡献.角色	3.2.1:MetametaData1.Contribute.Role
4.4.1.1:技术.要求.或组合.类型	4.4.1.1:Technical.Requirement.OrComposite.Type
4.4.1.2:技术.要求.或组合.名称	4.4.1.2:Technical.Requirement.OrComposite.Name
5.1:教育.交互类型	5.1:Educational.InteractivityType
5.2:教育.学习资源类型	5.2:Educational.LearningResourceType

5.3:教育.交互程度	5.3:Educational.InteractivityLevel
5.4:教育.语义密度	5.4:Educational.SemanticDensity
5.5:教育.终端用户类型	5.5:Educational.IntendedEndUserRole
5.6:教育.语境	5.6:Educational.Context
5.8:教育.难度	5.6:Educational.Difficulty
6.1:权利.费用	6.1:Rights.Cost
6.2:权利.版权和限制	6.2:Rights.CopyrightAndOtherRestrictions
7.1:关系.类型	7.1:Relation.Kind
9.1 分类.目的	9.1:Classification.Purpose

5.6.4 DateTime 接口

DateTime 是日期时间的接口。日期时间是学习对象元数据标准的基本数据类型，用于标准的表示日期时间，是一个时间点。

DateTime 接口定义			
成员	返回值/类型	参数说明	说明
getTime	字符串	空	获取日期时间
setDateTime	空	dateTime: 字符串	设置日期时间
setDescription	空	desc: Description	设置描述
getDescription	Description	空	获得描述

5.6.5 Duration 接口

Duration 是持续时间的接口。持续时间用于描述标准中设计到学习时长。

Duration 接口定义			
成员	返回值/类型	参数说明	说明
getDuration	字符串	空	获取时长
setDuration	空	duration: 字符串	设置时长
setDescription	空	desc: Description	设置描述
getDescription	Description	空	获得描述

表 5-3 是该接口的子接口以及其功能说明，其子接口只是简单的继承该接口，没有增加其他成员、方法。

表 5-3 持续时间子接口及功能介绍

子接口	功能
TypicalLearningTime	记录对于典型的目标用户来说，使用该学习对象一般或大约所需要的时间。

5.6.6 VCard 接口

VCard 数据类型被广泛的用于描述实体信息，在元数据标准中同样使用 VCard 作为实体信息描述的数据结构。

VCard 接口定义			
成员	返回值/类型	参数说明	说明
getName	字符串	空	获取 FN
setName	空	name: 字符串	设置 FN
setOrganization	空	org: 字符串	设置单位信息

getOrganization	字符串	空	获得单位信息
setTel	空	tel: 字符串	设置电话
getTel	字符串	空	获得电话
setEmail	空	email: 字符串	设置电子邮箱
getEmail	字符串	空	获得电子邮箱
setAddress	空	addr: 字符串	设置地址
getAddress	字符串	空	获得地址
setContent	空	content: 字符串	输入整个 VCard 数据并解析其中内容
getContent	字符串	空	返回标准的 VCard 串

5.6.7 Entity 接口

Entity 是实体的接口。元数据中通过实体作为 VCard 的载体来描述实体信息。

Entity 接口定义			
成员	返回值/类型	参数说明	说明
getVCard	字符串	空	获取 VCard 的字符串
setVCard	空	vcard: 字符串	通过字符串方式设置 VCard
getVCardAsObject	空	vcard: VCard	设置 VCard 对象
setVCardAsObject	VCard	空	获得 VCard 对象

5.6.8 Identifier 接口

Identifiers 是标识符接口。标识符用来对学习对象、元数据实例等内容进行标识。

Identifier 接口定义			
成员	返回值/类型	参数说明	说明
getCatalog	字符串	空	获得类别
setCatalog	空	catalog: 字符串	设置类别
getEntry	字符串	空	获得表项
setEntry	空	entry: 字符串	设置表项

5.6.9 Contribute 接口

Contribute 是贡献的接口。贡献元素用于记录对学习对象、学习对象元数据做出贡献的相关记录。

Contribute 接口定义			
成员	返回值/类型	参数说明	说明
getRole	Role	空	获得角色
setRole	空	role: Role	设置角色
getEntities	Entity[]	空	获得所有实体
setEntities	空	ents: Entity[]	设置所有实体
addEntity	空	entity: Entity	添加一个实体
emptyEntity	空	空	清空所有实体

5.6.10 OrComposite 接口

OrComposite 是或组合接口。或组合用于表示元数据技术类别中多个要求的技术组合方案。

OrComposite 接口定义			
成员	返回值/类型	参数说明	说明
getType	Type	空	获得技术类型
setType	空	type: Type	设置技术类型
getName	Name	空	获得技术名称
setName	空	name: Name	设置技术名称

setMinVersion	空	min: 字符串	获得技术的最低版本
getMinVersion	字符串	空	设置技术的最低版本
setMaxVersion	空	max: 字符串	获得技术的最高版本
getMaxVersion	字符串	空	设置技术的最高版本

5.6.11 Requirement 接口

Requirement 是要求接口。表示使用学习对象所需要的技术上的要求。

Requirement 接口定义			
成员	返回值/类型	参数说明	说明
getOrComposites	OrComposite[]	空	获得所有或组合
setOrComposites	空	ors: OrComposite[]	设置所有或组合
addOrComposite	空	or: OrComposite	添加一个或组合
emptyOrComposite	空	空	清空所有或组合

5.6.12 Resource 接口

Resource 是资源接口。资源元素表示该学习对象对其他学习对象的引用记录。

Resource 接口定义			
成员	返回值/类型	参数说明	说明
getIdentifiers	Identifier[]	空	获得所有资源标识符
setIdentifiers	空	ids: Identifier[]	设置所有资源标识符
addIdentifier	空	id: Identifier	添加一个资源标识符
getDescriptions	Description[]	空	获得所有描述信息
setDescriptions	空	descs: Description[]	设置所有描述信息
addDescription	空	desc: Description	添加一个描述信息
emptyDescriptions	空	空	清空所有描述信息

5.6.13 Taxon 接口

Taxon 是分类单元接口。分类单元是具有确定标号或术语的结点。这是从通用条目到具体条目的一条细化路径。一般用<标识, 条目>二元组来表示。

Taxon 接口定义			
成员	返回值/类型	参数说明	说明
getId	字符串	空	获得分类单元分类标识
setId	空	id: 字符串	设置分类单元分类标识
getEntry	TaxonEntry	空	获得分类条目
setEntry	空	entry: TaxonEntry	设置分类条目

5.6.14 TaxonPath 接口

TaxonPath 是分类路径接口。表示学习对象在某一特定分类系统中的分类路径。

TaxonPath 接口定义			
成员	返回值/类型	参数说明	说明
getSource	TaxonSource	空	获得分类系统的名称
setSource	空	source: TaxonSource	设置分类系统的名称
getTaxons	Taxon[]	空	获得所有分类单元信息

setTaxons	空	taxons: Taxon[]	设置所有分类单元信息
addTaxon	空	taxon: Taxon	添加一个分类单元信息
emptyTaxons	空	空	清空所有分类单元信息

5.6.15 General 接口

General 是通用类接口。该顶级元素对应的接口描述了学习对象的通用信息。

General 接口定义			
成员	返回值/类型	参数说明	说明
getIdentifiers	Identifier[]	空	获得所有标识符
setIdentifiers	空	ids: Identifier[]	设置所有标识符
addIdentifier	空	唯一标志: Identifier	添加一个标识符
emptyIdentifiers	空	空	清空所有标识符
getTitle	字符串	空	获得标题
setTitle	空	title: 字符串	设置标题
getLanguages	字符串数组	空	获得所有语种
setLanguages	空	langs: 字符串数组	设置所有语种
addLanguage	空	taxon: Taxon	添加一个语种
emptyLanguages	空	空	清空所有语种
getDescriptions	Description[]	空	获得所有描述信息
setDescriptions	空	descs: Description[]	设置所有描述信息
addDescription	空	desc: Description	添加一个描述信息
emptyDescriptions	空	空	清空所有描述信息
getKeywords	Keyword[]	空	获得所有关键字
setKeywords	空	keys: Keyword[]	设置所有关键字
addKeyword	空	key: Keyword	添加一个关键字
emptyKeywords	空	空	清空所有关键字
getCoverages	Coverage[]	空	获得所有覆盖区
setCoverages	空	covs: Coverage[]	设置所有覆盖区
addCoverage	空	cov: Coverage	添加一个覆盖区
emptyCoverages	空	空	清空所有覆盖区
getStructure	Structure	空	获得结构信息
setStructure	空	struct: Structure	设置结构信息
getAggregationLevel	AggregationLevel	空	获得聚合度信息
setAggregationLevel	空	aggr: AggregationLevel	设置聚合度信息

5.6.16 LifeCycle 接口

LifeCycle 是生命期接口。生命期类描述了学习对象的历史和当前状态以及那些对学习对象的发展过程发生作用的实体。

LifeCycle 接口定义			
-----------------------	--	--	--

成员	返回值/类型	参数说明	说明
getVersion	Version	空	获得版本信息
setVersion	空	version: Version	设置版本信息
getStatus	Status	空	获得状态信息
setStatus	空	status: Status	设置状态信息
getContributes	Contribute[]	空	获得所有贡献信息
setContributes	空	cons: Contribute[]	设置所有贡献信息
addContribute	空	con: Contribute	添加一个贡献信息
emptyContributes	空	空	清空所有贡献信息

5.6.17 Metametadata 接口

Metametadata 是元-元数据类接口。该类别描述了元数据实例自身的信息。

Metametadata 接口定义			
成员	返回值/类型	参数说明	说明
getIdentifiers	Identifier[]	空	获得所有标识符
setIdentifiers	空	ids: Identifier[]	设置所有标识符
addIdentifier	空	id: Identifier	添加一个标识符
emptyIdentifiers	空	空	清空所有标识符
getContributes	Contribute[]	空	获得所有贡献信息
setContributes	空	cons: Contribute[]	设置所有贡献信息
addContribute	空	con: Contribute	添加一个贡献信息
emptyContributes	空	空	清空所有贡献信息
getMetadataSchemas	字符串数组	空	获得所有元数据方案
setMetadataSchemas	空	schemas: 字符串数组	设置所有元数据方案
addMetadataSchema	空	空	添加一个元数据方案
emptyMetadataSchemas	空	空	清空所有元数据方案
getLanguage	字符串	空	获得元数据语言
setLanguage	空	lang: 字符串	设置元数据语言

5.6.18 Technical 接口

Technical 是技术类接口。技术类别描述了学习对象的技术要求及其相关特征。

Technical 接口定义			
成员	返回值/类型	参数说明	说明
getFormats	字符串数组	空	获得所有资源格式
setFormats	空	formats: 字符串数组	设置所有资源格式
addFormat	空	format: 字符串	添加一个资源格式
emptyFormats	空	空	清空所有资源格式
getLocations	字符串数组	空	获得所有资源路径
setLocations	空	locs: 字符串数组	设置所有资源路径

addLocation	空	loc: 字符串	添加一个资源路径
emptyLocations	空	空	清空所有资源路径
getRequirements	Requirement[]	空	获得所有技术需求
setRequirements	空	reqs: Requirement[]	设置所有技术需求
addRequirement	空	req: Requirement	添加一个技术需求
emptyRequirements	空	空	清空所有技术需求
getSize	长整型数	空	获得资源大小
setSize	空	size: 长整型数	设置资源大小
getInstallationRemarks	InstallationRemarks []	空	获得所有学习对象的安装描述信息
setInstallationRemarks	空	install: InstallationRemarks []	设置所有学习对象的安装描述信息
addInstallationRemark	空	install: InstallationRemarks	添加一个学习对象的安装描述信息
emptyInstallationRemarks	空	空	清空所有学习对象的安装描述信息
getOtherPlatformRequirements	OtherPlatformRequirements	空	获得其他平台需求
setOtherPlatformRequirements	空	other: OtherPlatformRequirements	设置其他平台需求
getDuration	Duration	空	获得持续时间
setDuration	空	duration: Duration	设置持续时间

5.6.19 Educational 接口

Educational 是教育类接口。该类别描述了学习对象在教育 and 教学方面的一些关键特征。

Educational 接口定义			
成员	返回值/类型	参数说明	说明
getLearningResourceTypes	LearningResourceType[]	空	获得所有学习资源类型
setLearningResourceTypes	空	types: LearningResourceTypes []	设置所有学习资源类型
addLearningResourceType	空	type: LearningResourceType	添加一个学习资源类型
emptyLearningResourceTypes	空	空	清空所有学习资源类型
getIntendedEndUse	IntendedEndUse	空	获得所有终端用户类型

rRoles	erRole[]		
setIntendedEndUserRoles	空	roles: IntendedEndUser Role []	设置所有终端用户类型
addIntendedEndUserRole	空	role: IntendedEndUser Role	添加一个终端用户类型
emptyIntendedEndUserRoles	空	空	清空所有终端用户类型
getContexts	Context[]	空	获得所有语境
setContexts	空	conts: Context[]	设置所有语境
addContext	空	cont: Context	添加一个语境
emptyContexts	空	空	清空所有语境
getTypicalAgeRanges	TypicalAgeRange[]	空	获得所有语境
setTypicalAgeRanges	空	conts : TypicalAgeRange []	设置所有语境
addTypicalAgeRange	空	cont : TypicalAgeRange	添加一个语境
emptyTypicalAgeRanges	空	空	清空所有语境
getDescriptions	Description[]	空	获得所有描述信息
setDescriptions	空	descs: Description[]	设置所有描述信息
addDescription	空	desc: Description	添加一个描述信息
emptyDescriptions	空	空	清空所有描述信息
getLanguages	字符串数组	空	获得所有语种
setLanguages	空	langs: 字符串数组	设置所有语种
addLanguage	空	taxon: Taxon	添加一个语种
emptyLanguages	空	空	清空所有语种
getInteractivityType	InteractivityType	空	获得交互类型
setInteractivityType	空	intType: InteractivityType	设置交互类型
getInteractivityLevel	InteractivityLevel	空	获得交互等级
setInteractivityLevel	空	intLevel: InteractivityLevel	设置交互等级
getSemanticDensity	SemanticDensity	空	获得语义密度
setSemanticDensity	空	semanticDens: SemanticDensity	设置语义密度

getDifficulty	Difficulty	空	获得难度
setDifficulty	空	difficulty: Difficulty	设置难度
getTypicalLearningTime	TypicalLearningTime	空	获得典型学习时间
setTypicalLearningTime	空	learningTime: TypicalLearningTime	设置典型学习时间

5.6.20 Rights 接口

Rights 是权利类接口。该类别描述了学习对象的知识产权和使用条件等信息。

Rights 接口定义			
成员	返回值/类型	参数说明	说明
getCost	Cost	空	获得是否收费信息
setCost	空	cost: Cost	设置是否收费信息
getCopyrightAndOtherRestrictions	CopyrightAndOtherRestriction[]	空	获得所有版权和其他限制信息
setCopyrightAndOtherRestrictions	空	copyrights: CopyrightAndOtherRestriction[]	设置所有版权和其他限制信息
addCopyrightAndOtherRestriction	空	copyright: CopyrightAndOtherRestriction	添加一条版权或使用限制信息
emptyCopyrightAndOtherRestrictions	空	空	清空所有版权限制信息

5.6.21 Relation 接口

Relation 是关系类接口。该类别定义了学习对象与其他学习对象（目标学习对象）之间的关系。

Relation 接口定义			
成员	返回值/类型	参数说明	说明
getKind	Kind	空	获得学习对象与指定目标的关系种类
setKind	空	kind: Kind	设置学习对象与指定目标的关系种类
getResource	Resource	空	获得与本学习对象关联的资源信息
setResource	空	resource: Resource	设置与本学习对象关联的资源信息

5.6.22 Annotation 接口

Annotation 是评注类接口。该类别提供了学习对象在教学使用方面的一些评价，以及这些评论的作者和创作时间。

Annotation 接口定义			
成员	返回值/类型	参数说明	说明
getEntity	Entity	空	获得学习对象评注信息对应的实体
setEntity	空	ent: Entity	设置学习对象评注信息对应的实体
getDate	DateTime	空	获得学习对象评注信息的时间
setDate	空	date: DateTime	设置学习对象评注信息的时间
getDescription	Description	空	获得评注信息

setDescription	空	desc: Description	设置评注信息
----------------	---	-------------------	--------

5.6.23 Classification 接口

Classification 是分类类接口。

Classification 接口定义			
成员	返回值/类型	参数说明	说明
getKeywords	Keyword[]	空	获得所有关键字
setKeywords	空	keys: Keyword[]	设置所有关键字
addKeyword	空	key: Keyword	添加一个关键字
emptyKeywords	空	空	清空所有关键字
getTaxonPaths	TaxonPath[]	空	获得所有分类路径
setTaxonPaths	空	paths: TaxonPath[]	设置所有分类路径
addTaxonPath	空	path: TaxonPath	添加一个分类路径
emptyTaxonPaths	空	空	清空所有分类路径
getDescription	Description	空	获得分类描述信息
setDescription	空	desc: Description	设置分类描述信息
getPurpose	Purpose	空	获得分类目的
setPurpose	空	purpose: Purpose	设置分类目的

5.6.24 LORoot 接口

LORoot 是学习对象根结点接口。该对象用于只是一个完整的学习对象的跟结点。该结点同时提供创建所有 9 个顶层元素的工厂方法。

LORoot 接口定义			
成员	返回值/类型	参数说明	说明
getGeneral	General	空	获得通用类
setGeneral	空	general: General	设置通用类
getLifeCycle	LifeCycle	空	获得生命期类
setLifeCycle	空	life: LifeCycle	设置生命期类
getMetametadata	Metametadata	空	获得元-元数据类
setMetametadata	空	metametadata: Metametadata	设置元-元数据类
getTechnical	Technical	空	获得技术类
setTechnical	空	tech: Technical	设置技术类
getEducational	Educational[]	空	获得所有教育类
setEducational	空	keys: Educational[]	设置所有教育类
addEducational	空	key: Educational	添加一个教育类
emptyEducational	空	空	清空所有教育类
getRelations	Relation[]	空	获得所有关系类
setRelations	空	relations: Relation[]	设置所有关系类
addRelation	空	relation: Relation	添加一个关系类
emptyRelations	空	空	清空所有关系类

getRights	Rights	空	获得权利类
setRights	空	rights: Rights	设置权利类
getAnnotations	Annotation[]	空	获得所有评注类
setAnnotations	空	annos: Annotation[]	设置所有评注类
addAnnotation	空	anno: Annotation	添加一个评注类
emptyAnnotations	空	空	清空所有评注类
getClassifications	Classification[]	空	获得所有分类类
setClassifications	空	classes: Classification[]	设置所有分类类
addClassification	空	cla: Classification	添加一个分类类
emptyClassifications	空	空	清空所有分类类
createGeneral	General	空	生成一个通用类实例
createLifeCycle	LifeCycle	空	生成一个生命期类实例
createMetametadata	Metametadata	空	生成一个元-元数据类实例
createTechnical	Technical	空	生成一个技术类实例
createEducational	Educational	空	生成一个教育类实例
createRelation	Relation	空	生成一个关系类实例
createRights	Rights	空	生成一个权利类实例
createAnnotation	Annotation	空	生成一个评注类实例
createClassification	Classification	空	生成一个分类类实例

5.6.25 LOFactory 接口

LOFactory 是学习对象工厂接口。该接口使用了设计模式中常用的工厂模式，该工厂用于创建一个完整的学习对象实例。

LOFactory 接口定义			
成员	返回值/类型	参数说明	说明
createRoot	LORoot	空	创建学习对象元数据实例

5.6.26 LOAction 接口

LOAction 是学习对象动作接口。该接口用于进行学习对象元数据相关的操作的父类接口。

LOAction 接口定义			
成员	返回值/类型	参数说明	说明
execute	空	空	执行该动作对应的功能

表 5-4 中是该接口的子接口，它们规定了对于学习对象的各种操作，通过实现 execute 方法可以将操作具体化。

表5-4 动作接口及功能介绍

动作接口	功能
UploadAction	上载学习资源并导入默认的学习对象元数据模板
GenLOMAAction	生成学习对象元数据，并存储到学习对象元数据库

ViewLOMAction	浏览学习对象元数据
DeleteLOMAction	删除学习对象元数据
SearchLOMAction	检索学习对象元数据
ChooseTemplateAction	选择学习对象元数据录入模板
SaveAsTemplateAction	将学习对象元数据存储为模板

5.7 内容包装 (Content Packaging) 服务

服务描述：该服务提供对学习内容包装的支持，包括学习内容包的导入，导出，存储，编辑等功能。

5.7.1 HasMetadata 接口

HasMetadata 接口代表一段元数据，在学习内容包中的很多数据元素都包含元数据，因此有必要抽象出来单独的作为接口。

HasMetadata 接口定义			
成员	返回值/类型	参数说明	说明
getId	字符串	空	获得元数据标识符
setId	空	id: 字符串	设置元数据标识符
getMetadata	字符串	空	获得元数据内容
setMetadata	空	metadata: 字符串	设置元数据内容

5.7.2 Level 接口

Level 是等级接口，代表学习内容包中结点的共性操作属性接口。

Level 接口定义			
成员	返回值/类型	参数说明	说明
getTitle	字符串	空	获得元素标题
setTitle	空	title: 字符串	设置元素标题
getItems	Item[]	空	获得当前结点所有子结点元素
setItems	空	items: Item[]	设置当前结点所有子结点元素
addItem	空	item: Item	顺序地添加一个子结点
emptyItems	空	空	清空所有子结点
addItem	空	item: Item, location: 整型	按照指定位置插入子结点
deleteItem	空	location: 整型	删除指定位置的子结点
deleteItem	空	content: 字符串	删除包含指定内容的子结点
getItemById	Item	id: 字符串	根据 ID 获得子结点
getChildren	Iterator	空	返回子结点迭代器
getSize	整型	空	返回子结点数量
find	整型	content: 字符串	根据内容查找子结点位置

5.7.3 Item 接口

Item 是结点接口，表示内容结构中的一个结点。结点中包括了对资源信息的引用，结点终包含的标题信息等属性。

Item 接口定义			
成员	返回值/类型	参数说明	说明
isVisible	布尔值	空	元素是否可见

setVisible	空	visible: 布尔值	设置元素是否可见
getParameters	字符串	空	获得当前结点所有参数
setParameters	空	params: 字符串	设置当前结点所有参数
getIDRef	Resource	空	获得结点关联资源
setIDRef	空	idref: Resource	设置结点关联资源
isLeaf	布尔值	空	当前结点是否叶结点

5.7.4 Organization 接口

Organization 是组织接口，表示对学习内容包清单中所有实际物理资源和所需媒体素材的引用。

Organization 接口定义			
成员	返回值/类型	参数说明	说明
isDefault	布尔值	空	是否为默认?
setDefault	空	default: 布尔值	设置为默认
getStructure	字符串	空	获得结构信息
setStructure	空	structure: 字符串	设置结构信息

5.7.5 Organizations 接口

Organizations 是组织集接口，表示学习内容包清单中一种或多种学习内容的组织方法。

Organizations 接口定义			
成员	返回值/类型	参数说明	说明
getDefaultID	字符串	空	获得默认结点标识符
setDefaultID	空	default: 字符串	设置默认结点标识符
getOrganizations	Organization[]	空	获得所有组织结点
getOrganization	Organization	id: 字符串	根据标识符获取组织结点
getOrganization	Organization	location: 整型	根据位置获取组织结点
addOrganization	空	org: Organization, location: 整型	将组织结点插入到指定位置
setOrganizations	空	orgs: Organization[]	设置所有组织结点
emptyOrganizations	空	空	清空所有组织结点
remove	空	location: 整型	删除指定位置的组织结点
find	整型	id: 字符串	根据标识符查找组织结点位置
iterator	Iterator	空	获得组织结点迭代器
size	整型	空	获得组织结点个数
getAllItemID	字符串数组	空	获得所有 Item 结点标识符
getItemByID	Item	id: 字符串	根据标识符获得 Item 结点

5.7.6 ResourceFile 接口

ResourceFile 是资源文件接口，包含了对本地资源文件的引用信息。

ResourceFile 接口定义			
成员	返回值/类型	参数说明	说明
getHref	字符串	空	获得资源文件路径
setHref	空	href: 字符串	设置资源文件路径

5.7.7 Dependency 接口

Dependency 是依赖接口，表示对引用资源所依赖的文件。

Dependency 接口定义			
成员	返回值/类型	参数说明	说明
getIDRef	字符串	空	获得对本地资源文件的引用
setIDRef	空	idref: 字符串	设置对本地资源文件的引用

5.7.8 Resource 接口

Resource 是资源接口，表示对学习内容有实际物理资源和所需媒体素材的引用。

Resource 接口定义			
成员	返回值/类型	参数说明	说明
getType	字符串	空	获得资源 MIME 类型
setType	空	type: 字符串	设置资源 MIME 类型
getXMLBase	字符串	空	获得 xml:base 属性
setXMLBase	空	base: 字符串	设置 xml:base 属性
getHref	字符串	空	获得资源文件路径
setHref	空	href: 字符串	设置资源文件路径
getResourceFiles	ResourceFile[]	空	获得所有资源文件
setResourceFiles	空	resources: ResourceFile[]	设置所有资源文件
addResourceFile	空	resources: ResourceFile	添加一个资源文件
emptyResourceFiles	空	空	清空所有资源文件
getDependencies	Dependency[]	空	获得所有引用文件列表
setDependencies	空	deps: Dependency[]	设置所有引用文件列表
addDependency	空	dep: Dependency	添加一个引用文件
emptyDependencies	空	空	清空所有引用文件

5.7.9 Resources 接口

Resources 是资源列表接口，表示一个资源引用的集合。

Resources 接口定义			
成员	返回值/类型	参数说明	说明
getXMLBase	字符串	空	获得 xml:base 属性
setXMLBase	空	base: 字符串	设置 xml:base 属性
getResources	Resource[]	空	获得所有资源结点
getResource	Resource	id: 字符串	根据标识符获取资源结点
getResource	Resource	location: 整型	根据位置获取资源结点
addResource	空	org: Resource, location: 整型	将资源结点插入到指定位置
setResources	空	orgs: Resource[]	设置所有资源结点
emptyResources	空	空	清空所有资源结点

remove	空	id: 字符串	删除指定标识符的资源结点
remove	空	location: 整型	删除指定位置的资源结点
find	整型	id: 字符串	根据标识符查找资源结点位置
iterator	Iterator	空	获得资源结点迭代器
size	整型	空	获得资源结点个数
getAllResourceID	字符串数组	空	获得所有资源标识符

5.7.10 ManifestMetadata 接口

ManifestMetadata 是内容清单数据接口，表示内容清单使用的元数据信息。

ManifestMetadata 接口定义			
成员	返回值/类型	参数说明	说明
getSchema	字符串	空	获得该学习内容包使用的规范
setSchema	空	schema: 字符串	设置该学习内容包使用的规范
getSchemaVersion	字符串	空	获得该学习内容包使用规范的版本
setSchemaVersion	空	schemaVersion: 字符串	设置该学习内容包使用规范的版本

5.7.11 Manifest 接口

Manifest 是内容清单接口，表示学习内容包的根结点。

Manifest 接口定义			
成员	返回值/类型	参数说明	说明
getManifests	Manifest[]	空	获得所有子内容清单
setManifests	空	subManifests: Manifest[]	设置所有子内容清单
addManifest	空	subManifest: Manifest	添加一个子内容清单
emptyManifests	空	空	清空所有子内容清单
getMetadata	ManifestMetadata	空	获得内容清单元数据
setMetadata	空	metadata: ManifestMetadata	设置内容清单元数据
getVersion	字符串	空	获得版本属性
setVersion	空	version: 字符串	设置版本属性
getXMLBase	字符串	空	获得 xml:base 属性
setXMLBase	空	base: 字符串	设置 xml:base 属性
getId	字符串	空	获得元数据标识符
setId	空	id: 字符串	设置元数据标识符
getOrganizations	Organizations	空	获得组织列表
setOrganizations	空	orgs: Organizations	设置组织列表
getDefaultOrg	字符串	空	获得默认组织
setDefaultOrg	空	id: 字符串	设置默认组织
getResources	Resources	空	获得资源列表
setResources	空	resources: Resources	设置资源列表

getOrganization	Organization	id: 字符串	根据标识符获取组织
getResource	Resource	id: 字符串	根据标识符获取资源
addResource	空	resource: Resource	添加一个资源
deleteResource	空	id: 字符串	根据标识符删除资源

5.7.12 CAction 接口

CAction 是内容包装动作接口。该接口用于进行内容包装相关的操作的父类接口。

CAction 接口定义			
成员	返回值/类型	参数说明	说明
execute	空	空	执行该动作对应的功能

附录 A
(资料性附录)
JAVA 接口

A.1 基础服务接口

A.1.1 基础 (base) 服务

base 包提供的是一些基础接口，供其它接口使用，这些接口通常会被继承，base 所提供的正是这样一个基类库。

A.1.1.1 Manager 接口

Manager 是将应用程序绑定到具体服务实现的关键，接口的描述需要定义一个接口扩展 Manager，而接口的实现要求以一个类实现具体的 Manager。应用程序直接或间接通过 Manager 访问服务对象。

cn.edu.celtsc.intf.common.base.Manager 接口定义			
import: cn.edu.celtsc.shared.CeltsException java.io.Serializable java.util.Properties			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
getContext	CeltsContext	无	获得传递的上下文内容
assignContext	void	context: CeltsContext	给上下文赋值
assignConfiguration	void	configuration: Properties	对环境参数赋值

A.1.1.2 Properties 接口

Properties 接口用于返回对象的一些只读属性，这些属性通过一系列串行化的值得到，并用关键码标记。

cn.edu.celtsc.intf.common.base.Properties 接口定义			
import: java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
getType	CeltsType	无	获得属性类型
getProperty	Serializable	key: Serializable, 属性的 关键码	获得属性
getKeys	ObjectIterator	无	得到属性的关键码

A.1.1.3 CeltsContext 接口

CeltsContext 包含应用程序共用的语境信息。

cn.edu.celtsc.intf.common.base.CeltsContext 接口定义			
import: java.io.Serializable java.lang.Object			
type: public			
implements: Serializable			
extends: Object			
成员	返回值/类型	参数说明	说明
assignContext	无	key: 字符串	为 CeltsContext 的语境信息

		context: 可序列化对象	的内容赋值
getContext	可序列化对象	key: 字符串	得到 CeltsContext 的语境信息

A.1.1.4 Iterator 接口

Iterator 提供的是迭代器接口，用于数据的批量读取，对于具体的对象类型，将通过继承 Iterator 类实现自己的迭代器接口。

cn.edu.celtsc.intf.common.base.Iterator 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
hasNext	boolean	无	是否还有下一个元素
Next	void	无	取下一个元素

A.1.1.5 PropertiesIterator 接口

cn.edu.celtsc.intf.common.base.PropertiesIterator 接口定义			
import: java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextProperties	boolean	无	是否还有下一个属性
nextProperties	Properties	无	取下一个属性

A.1.1.6 CeltsType 接口

CeltsType 接口按照事物的基本概念进行分类，被用于各个接口中。CeltsType 的精确定义由使用者确定，CeltsType 接口的形式也有利于分类，CeltsType 有四个字符串属性：authority、domain、keyword 以及 description，前三个元素被 isEqual 成员用来判断两个 CeltsType 对象是否相等，description 用来阐明实例的语义。

cn.edu.celtsc.intf.common.base.CeltsType 接口定义			
import: java.io.Serializable			
import: java.lang.Object			
type: public			
implements: Serializable			
extends: Object			
成员	返回值/类型	参数说明	说明
getAuthority	字符串	无	获得 CeltsType 的 authority
getDescription	字符串	无	得到 CeltsType 的描述
getDomain	字符串	无	得到使用域
getKeyword	字符串	无	得到关键字
isEqual	布尔值	celtsType2: CeltsType	判断两个 CeltsType 对象是否相同

A.1.1.7 TypeIterator 接口

cn.edu.celtsc.intf.common.base.TypeIterator 接口定义			
import: java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextType	boolean	无	是否还有下一个类型

nextType	CeltsType	无	取下一个类型
----------	-----------	---	--------

A.1.1.8 IdIterator 接口

cn.edu.celtsc.intf.common.base.IdIterator 接口定义			
import: cn.edu.celtsc.intf.common.id.Id			
java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextId	boolean	无	是否还有下一个 ID
nextId	Id	无	取下一个 ID

A.1.1.9 ObjectIterator 接口

cn.edu.celtsc.intf.common.base.ObjectIterator 接口定义			
import: java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextObject	boolean	无	是否还有下一个对象
nextObject	Serializable	无	取下一个对象

A.1.2 认证 (authentication) 服务

Authentication 提供的是认证相关的服务，认证服务是要帮助应用程序确定一个用户是否经过了认证，认证服务可以处理不同类型的认证，具体的认证过程要由服务的具体实现确定。有些认证仅仅依靠用户密码进行，有的则是基于证书的认证，也有些是远程认证，这些都是认证类型所要表达的。

A.1.2.1 AuthenticationManager 接口

cn.edu.celtsc.intf.common.authentication.AuthenticationManager 接口定义			
import: cn.edu.celtsc.intf.common.id.Id			
cn.edu.celtsc.intf.common.base.Manager			
cn.edu.celtsc.intf.common.base.TypeIterator			
cn.edu.celtsc.shared.CeltsException			
cn.edu.celtsc.intf.common.base.CeltsType			
type: public			
extends: cn.edu.celtsc.intf.common.base.Manager			
成员	返回值/类型	参数说明	说明
getAuthenticationTypes	TypeIterator	无	获得认证类型
authenticateUser	void	authenticationType: CeltsType	根据认证类型对用户认证
isUserAuthenticated	boolean	authenticationType: CeltsType	判断用户是否已经过认证
getUserId	Id	authenticationType: CeltsType	得到认证用户的 ID
destroyAuthentication	void	无	清除认证记录
destroyAuthenticationForType	void	authenticationType: CeltsType	清除同类型的认证记录

A.1.3 数据库管理 (databasemanagement) 服务

databasemanagement 服务提供了数据库管理的基本操作，包括对关系数据库中常用数据类型的封装

定义，以及存取操作及结果表达的定义。

A. 1. 3. 1 Blob 接口

Blob 是关系数据库中常用的一种类型，在 Blob 列中可以存放大数据量的二进制数据，如图片、音频、视频数据等，该接口提供对数据表中 Blob 列的访问。Blob 会记录当前的读写位置，是一个 long 型整数。

cn.edu.celtsc.intf.common.databasesmanagement.Blob 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
length	long	无	返回该 Blob 数据的长度，单位字节
getBytes	byte[]	pos: long length: int	从 pos 指定的位置开始，读取 length 字节的数据，结果以字节数组形式返回
setBytes	int	position: long bytes: byte[]	将 bytes 所含数据写入该 Blob 中从 position 开始的地方
setBytes	int	position: long bytes: byte[] offset: int length: int	将 bytes 中从 offset 开始的长度为 length 的数据写入该 Blob 中从 position 开始的地方

A. 1. 3. 2 Clob 接口

Clob 是关系数据库另一种常用的类型，在 Clob 列中可以存放大量的字符数据，如大量的文本，该接口提供对数据表中 Clob 列的访问。

cn.edu.celtsc.intf.common.databasesmanagement.Clob 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
Length	long	无	返回该 Clob 数据的长度，单位字节
getSubString	String	pos: long length: int	从 pos 指定的位置开始，读取 length 个字符，结果以字符串返回
setString	int	pos: long string: String	将 string 字符串写入该 Clob 中从 position 开始的地方，返回成功写入的字符数
setString	int	pos: long string: String offset: int length: int	将 string 字符串中从 offset 开始的长度为 length 的子字符串写入该 Clob 中从 position 开始的地方，返回成功写入的字符数

A. 1. 3. 3 Connection 接口

Connection 接口代表了一个到数据库的连接，通过该连接可以执行数据库的查询、修改操作，另外不再使用的数据库连接可以释放，以节省数据库连接资源。

cn.edu.celtsc.intf.common.databasesmanagement.Connection 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
executeQuery	ResultTable	sql : String	执行 sql 所表示的查询语句，结果以 ResultTable 返回
executeQueryWithParams	ResultTable	sql: String params: String[]	执行 sql 所表示的查询语句，sql 中的占位符使用 params 提供的具体值替换，结果以 ResultTable 返回
executeUpdate	int	sql: String	执行 sql 所表示的更新语句

executeQueryWithParams	int	sql: String params: String[]	执行 sql 所表示的更新语句, sql 中的占位符使用 params 提供的具体值替换
Close	void	无	关闭该连接

A.1.3.4 ResultMetaData 接口

ResultMetaData 接口可以获取关于一个 ResultTable 的列的信息, 如类型, 列名称等; ResultMetaData 通过 ResultTable 的 getMetaData 方法获取, 提供关于对应该 ResultTable 的元信息。ResultMetaData 提供如下信息: 列数目、列名称、列类型、列显示时需要的宽度 (字符宽度)、是否接受 NULL 值等。

cn.edu.celtsc.intf.common.databasemanagement.ResultMetaData 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
getColumnCount	int	无	获取列数目
getColumnName	String	columnIndex: int	获取列 columnIndex 的名称
getColumnType	int	columnIndex: int	获取列 columnIndex 的类型常量
getTableName	String	columnIndex: int	获取列 columnIndex 原先所属的表名
getColumnDisplaySize	int	columnIndex: int	获取列 columnIndex 显示时需要多少字符宽度
isNullable	boolean	columnIndex: int	判断列 columnIndex 是否可接受 NULL 值
isReadOnly	boolean	columnIndex: int	判断列 columnIndex 是否只读
isWritable	boolean	columnIndex: int	判断列 columnIndex 是否可写
isSearchable	boolean	columnIndex: int	判断列 columnIndex 是否可以被搜索

A.1.3.5 ResultTable 接口

ResultTable 用来表示数据库查询结果, 提供方法获得结果表格中每条记录的数据。ResultTable 继承 cn.edu.celtsc.intf.common.base.Iterator, 可以依次遍历 ResultTable 中的各个记录 (行), 进而读取数据。

cn.edu.celtsc.intf.common.databasemanagement.ResultTable 接口定义			
import: cn.edu.celtsc.intf.common.base.Iterator			
type: public			
extends: Iterator			
成员	返回值/类型	参数说明	说明
getMetaData	ResultMetaData	无	获得结果表格的原数据, 如表名等
getBoolean	Boolean	columnIndex: int	获得当前行列序号为 columnIndex 的 boolean 值
getInt	Int	columnIndex: int	获得当前行列序号为 columnIndex 的 int 值
getDouble	double	columnIndex: int	获得当前行列序号为 columnIndex 的 double 值
getString	String	columnIndex: int	获得当前行列序号为 columnIndex 的 String 值
getDate	String	columnIndex: int	获得当前行列序号为 columnIndex 的 Date 值
getTime	String	columnIndex: int	获得当前行列序号为 columnIndex 的 Time 值
getBlob	Blob	columnIndex: int	获得当前行列序号为 columnIndex 的 Blob

			值
getClob	Clob	columnIndex: int	获得当前行列序号为 columnIndex 的 Clob 值
getBytes	byte[]	columnIndex: int	以字节数组的形式获得当前行列序号为 columnIndex 的值
wasNull	boolean	无	判断结果表格是否为空
isFirst	boolean	无	判断当前位置是否在表格的第一行
isLast	boolean	无	判断当前位置是否是表格的最后一行

A.1.3.6 SQLFactory 接口

SQLFactory 接口提供工厂方法用来获取到数据库的连接，使用该连接即可进行多种数据库相关的操作，可以通过两种方法获得一个数据库连接。

cn.edu.celtsc.intf.common.databasesmanagement.SQLFactory 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
createConnection	Connection	url: String	根据题提供的 url 字符串创建一个数据库连接
createConnection	Connection	url: String user: String password: String	根据提供的 url 字符串，用户名 user 以及密码 password 创建一个数据库连接

A.1.4 文件 (file) 服务

file 服务提供文件及目录的访问，支持常见的文件操作，如文件/目录的读写、创建、删除等。

A.1.4.1 Directory 接口

Directory 接口表示文件系统中的目录，目录用来组织文件，目录中可以包含子目录和文件。该接口提供方法供在目录树的不同层次间进行移动，如 getParent(), getDirectories() 等；该接口提供了在本目录中创建子目录和文件的方法 createSubDirectory(), createFile(); 另外该接口还支持目录的拷贝和删除操作等。

cn.edu.celtsc.intf.common.file.Directory 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
getParent	Directory	无	获得该目录的父目录，根目录则返回空
isRoot	boolean	无	检测该目录是否为根目录
getPath	Path	无	获得该目录对应的 Path
Copy	void	dest: Path, 目录将被拷贝到的目的路径	将本目录及其子内容拷贝到 dest 所指定的目标目录
delete	void	无	删除该目录及其内容
createSubDirectory	Directory	dirName: String, 将要创建的子目录名称	在该目录下创建名称为 dirName 的子目录
createFile	File	filename: String, 将要创建的文件名称	在该目录下创建名称为 fileName 的文件
getDirectoryRoot	Path	无	返回该目录所属的根目录
getName	String	无	获得该目录名称，
getFiles	File[]	无	获取该目录所含文件的列表，返回代表该列表的数组

getDirectories	Directory[]	无	获取该目录下所有的子目录列表，返回代表该列表点额数组
----------------	-------------	---	----------------------------

A.1.4.2 File 接口

File 接口用来表示一个文件，提供基本的面向字节的文件读写操作，并且提供用于访问和设置文件基本信息的接口，如读取和更改文件的创建时间等，判断文件是否可写等；另外文件还提供 MIME 信息，便于在网络环境下使用。

cn.edu.celtsc.intf.common.file.File 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
Copy	void	dest: Path	将该文件拷贝到 dest 所示的目录
Copy	void	dest: Path newName: String	将该文件拷贝到 dest 所示的目录，新文件名称为 newName
Copy	void	newName: String	将文件拷贝到同一目录下，用文件名 newName 保存
Delete	void	无	删除文件
isReadable	boolean	无	检查文件是否可读
isWritable	boolean	无	检查文件是否可写
getMimeType	String	无	获取文件的 MIME
updateMimeType	void	type: String	设置文件的 MIME
updateReadOnly	void	无	将文件设置为只读
updateWritable	void	无	将文件设置为可写
Length	long	无	返回文件的大小，以字节为单位
Read	int	buffer: byte[] offset: int count: int	从文件当前位置读取 count 个字节的内容，读取内容放入 buffer 中 offset 起始的空间
Read	int	无	从文件当前位置读取一个字节的内 容
Write	void	buffer: byte[]	将 buffer 中的数据写入文件当前位置
Write	void	b: byte	将字节 b 写入文件当前位置
Flush	void	无	
Close	void	无	关闭文件
getPath	Path	无	获得文件路径
getDirectory	Directory	无	获得文件所处的目录
getCreationTime	String	无	获取文件创建时间
setCreationTime	void	time: String	修改文件创建时间为 time
getLastAccessTime	String	无	获取文件上次访问时间
setLastAccessTime	void	time: String	修改文件上次访问时间为 time
getLastWriteTime	String	无	获取文件上次修改时间
setLastWriteTime	void	time: String	修改文件上次修改时间

A.1.4.3 Path 接口

Path 接口表示文件或者目录的路径，可以是相对路径或者绝对路径，提供方便的方法对路径进行操作，如获取一个路径的目录名，文件名，绝对路径等。File 服务中的 File 和 Directory 接口都是通过 Path 进行标识，它们的实例都包括一个对 Path 实例的引用。

cn.edu.celtsc.intf.common.file.Path 接口定义			
type: public			
常量定义	类型	值	说明
directorySeparatorChar	char	\"或'/'	路径中目录间的分隔符，通常在 Windows 系统中该字符为\"，在 UNIX 系统中为'/'
invalidPathChars	char[]	{',', '\\', ':'}	不允许出现在路径中的字符，这些字符称为非法字符。
pathSeparator	char	';	多个路径之间的分隔符；像 Java 中常用的环境变量 ClassPath 通常包括多个路径，用来在多个位置寻找可能的类，这些路径之间使用 pathSeparator 进行分隔，大多数系统中该字符为';'。
volumeSeparator	char	':'	卷标分隔符
成员	返回值/类型	参数说明	说明
getDirectoryName	String	无	获取父目录名
getName	String	无	获取路径名
getExtension	String	无	获取后缀
getFullPath	String	无	获取路径全名
getPath	String	无	获取当前路径的抽象路径
isAbsolutePath	boolean	无	判断是否为绝对路径
isDirectory	boolean	无	判断是否为目录
isFile	boolean	无	判断是否为文件

A.1.4.4 FileManager 接口

FileManager 采用工厂化方法负责此服务中其他接口实例的创建，可以分别创建 Path，File 和 Directory 的实例。

cn.edu.celtsc.intf.common.file.FileManager 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
createPath	Path	path: String, 表示文件系统中的路径	
openFile	File	filePath: Path, 代表将要被打开的文件位置	
createFile	File	location: Path, 表示将要创建的文件的路径	
openDirectory	Directory	dirPath: Path, 代表将要被打开的目录位置	
createDirectory	Directory	path: Path, 将要创建的文件的路径	

A.1.5 标识 (id) 服务

ID 是系统中唯一的标识符，id 服务即提供 ID 的数据模型以及 ID 的生成管理等功能。

A.1.5.1 Id 接口

cn.edu.celtsc.intf.common.id.Id 接口定义			
import: java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
getIdString	String	无	获取 ID 的字符串

isEqual	boolean	id: Id	判断两个 ID 是否相等
---------	---------	--------	--------------

A.1.5.2 IdManager 接口

cn.edu.celtsc.intf.common.id.IdManager 接口定义			
import: cn.edu.celtsc.intf.common.base.CeltsException cn.edu.celtsc.shared.Manager			
type: public			
extends: Manager			
成员	返回值/类型	参数说明	说明
createId	Id	无	创建一个 id 实例，如果创建过程发生错误，则抛出 CeltsException。
getId	Id	idString: String	通过表示 id 的字符串获得相应 id 实例，如果此 id 还没有实例创建，则创建一个实例。

A.1.6 日志 (logging) 服务

所有的日志条目 (log entry) 都有一个格式类型，一个优先级类型和一个时间戳。

A.1.6.1 Entry 接口

包括被写入日志的项目 (item)，以及格式类型，优先级类型和被写入日志的时间。

cn.edu.celtsc.intf.common.logging.Entry 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.base.CeltsType			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
getFormatType	CeltsType	无	获取格式类型
getItem	Serializable	无	获取 Item
getPriorityType	CeltsType	无	获取优先级类型
getTimestamp	long	无	获取时间戳

A.1.6.2 EntryIterator 接口

EntryIterator 为一组 Entry 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.logging.EntryIterator 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.logging.Entry			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextEntry	boolean	无	是否还有下一个 Entry
nextEntry	Entry	无	取下一个 Entry

A.1.6.3 ReadableLog 接口

ReadableLog 接口对日志进行读操作。

cn.edu.celtsc.intf.common.logging.ReadableLog 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.base.CeltsType cn.edu.celtsc.intf.common.logging.EntryIterator			
type: public			

extends: Serializable			
成员	返回值/类型	参数说明	说明
getDisplayName	String	无	获取显示名字
getEntries	EntryIterator	formatType: CeltsType priorityType : CeltsType	以后进先出的顺序获取所有条目

A.1.6.4 WritableLog 接口

WritableLog 接口允许对日志的条目、类型、时间戳等进行写操作，共有两种追加日志的方法。

cn.edu.celtsc.intf.common.logging.WritableLog 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.base.CeltsType			
type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
appendLog	void	entryItem: Serializable	追加日志
appendLogWithTypes	void	entryItem: Serializable, formatType: CeltsType, priorityType: CeltsType	追加有类型日志
assignFormatType	void	formatType: CeltsType	设置格式类型
assignPriorityType	void	priorityType: CeltsType	设置优先级类型
getDisplayName	String	无	获取显示名字

A.1.6.5 LogManager 接口

LogManager 负责创建、删除、访问日志。

cn.edu.celtsc.intf.common.logging.Entry 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.base.TypeIterator cn.edu.celtsc.intf.common.logging.Entry cn.edu.celtsc.intf.common.logging.EntryIterator cn.edu.celtsc.intf.common.logging.ReadableLog cn.edu.celtsc.intf.common.logging.WritableLog			
type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
createLog	WritableLog	logName: String	创建一个 WritableLog
deleteLog	void	logName: String	删除一个日志
getFormatTypes	TypeIterator	无	获取这个实现的所有格式类型
getLogForReading	ReadableLog	logName: String	为读操作打开一个日志
getLogForWriting	WritableLog	logName: String	为写操作打开一个日志
getLogNamesForReading	StringIterator	无	获取所有可以读的日志的名字
getLogNamesForWriting	StringIterator	无	获取所有可以写的日志的名字
getPriorityTypes	TypeIterator	无	获取所有可用的优先级类型
supportsReading	boolean	无	该方法返回值表明当前的实现是否支持 ReadableLog 接口和 LogManager 接口的方法：

			getLogForReading, getLogNamesForReading.
--	--	--	---

A.1.7 授权 (authorization) 服务

授权服务完成对用户的授权处理。其中限定器 (Qualifier) 是一个上下文, 包括一个由 Id 构成的有效授权 (Authorization)、描述和限定器类型。授权中的 Id 是在外部定义的, 具体实现必须保证 Id 的唯一性。

A.1.7.1 Qualifier 接口

cn.edu.celtsc.intf.common.authorization.Qualifier 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.base.CeltsType cn.edu.celtsc.intf.common.id.Id			
type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
addParent	void	parentQualifierId: Id	为该 Qualifier 添加一个父结点
changeParent	void	oldParentId: Id, newParentId: Id	改变该 Qualifier 的父结点
getChildren	QualifierIterator	无	获取该 Qualifier 的所有子结点
getDescription	String	无	获取该 Qualifier 的描述
getId	Id	无	获取该 Qualifier 的 Id
getParents	QualifierIterator	无	获取该 Qualifier 的所有父结点
getQualifierType	CeltsType	无	获取该 Qualifier 的限定器类型
getReferenceName	String	无	获取该 Qualifier 的固定引用名
isChildOf	boolean	parentId: Id	是否是结点 parentId 的子结点
isDescendantOf	boolean	ancestorId: Id	是否是结点 ancestorId 的子孙结点
isParent	boolean	无	是否有子结点
removeParent	void	parentQualifierId: Id	移除一个父结点
updateDescription	void	description: String	更新该 Qualifier 的描述

A.1.7.2 QualifierIterator 接口

QualifierIterator 为一组 Qualifier 提供顺序访问, 每次只能访问一个。

cn.edu.celtsc.intf.common.authorization.QualifierIterator 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.authorization.Qualifier			
type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextQualifier	boolean	无	是否还有下一个 Qualifier
nextQualifier	Qualifier	无	取下一个 Qualifier

A.1.7.3 Function 接口

Function 由 Id、显示名字 (displayName)、描述 (description)、类别 (category) 和限定器类型 (QualifierType) 组成。Id 是在外部定义的, 具体实现必须保证 Id 的唯一性。

cn.edu.celtsc.intf.common.authorization.Function 接口定义
--

<pre>import: java.io.Serializable cn.edu.celtsc.intf.common.base.CeltsType cn.edu.celtsc.intf.common.id.Id type: public extends: Serializable</pre>			
成员	返回值/类型	参数说明	说明
getDescription	String	无	获取该 Function 的描述
getFunctionType	CeltsType	无	获取该 Function 的类型
getId	Id	无	获取该 Function 的 Id
getReferenceName	String	无	获取该 Function 的固定引用名
updateDescription	void	description: String	更新该 Function 的描述

A. 1. 7. 4 FunctionIterator 接口

FunctionIterator 为一组 Function 提供顺序访问，每次只能访问一个。

<p>cn.edu.celtsc.intf.common.authorization.FunctionIterator 接口定义</p> <pre>import: java.io.Serializable cn.edu.celtsc.intf.common.authorization.Function type: public extends: Serializable</pre>			
成员	返回值/类型	参数说明	说明
hasNextFunction	boolean	无	是否还有下一个 Function
nextFunction	Function	无	取下一个 Function

A. 1. 7. 5 Authorization 接口

Authorization 指明一个 agentId 在 Qualifier 上下文中如何执行一个 Function。

<p>cn.edu.celtsc.intf.common.authorization.Authorization 接口定义</p> <pre>import: java.io.Serializable cn.edu.celtsc.intf.common.base.CeltsType cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.authorization.Qualifier cn.edu.celtsc.intf.common.authorization.Function type: public extends: Serializable</pre>			
成员	返回值/类型	参数说明	说明
getAgentId	Id	无	获取该 Authorization 相关联的 AgentId
getEffectiveDate	long	无	获取该 Authorization 的生效日期
getExpirationDate	long	无	获取该 Authorization 的失效日期
getFunction	Function	无	获取该 Authorization 的 Function
getModifiedBy	Id	无	获取修改该 Authorization 的 Id
getModifiedDate	long	无	获取该 Authorization 的修改日期
getQualifier	Qualifier	无	获取该 Authorization 的 Qualifier
isActiveNow	boolean	无	该 Authorization 当前是否生效
isExplicit	boolean	无	该 Authorization 是否显式存储，有的授权是显式的，有些是隐式的
updateEffectiveDate	void	effectiveDate: long	更新该 Authorization 生效日期

updateExpirationDate	void	expirationDate: long	更新该 Authorization 失效日期
----------------------	------	----------------------	------------------------

A.1.7.6 AuthorizationIterator 接口

AuthorizationIterator 为一组 Authorization 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.authorization.AuthorizationIterator 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.authorization.Authorization			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextAuthorization	boolean	无	是否还有下一个 Authorization
nextAuthorization	Authorization	无	取下一个 Authorization

A.1.7.7 AuthorizationManager 接口

授权管理器（AuthorizationManager）允许一个应用程序创建授权（Authorization），为授权设置标准，查询授权以决定一个 agentId 在 Qualifier 上下文中如何执行一个 Function。

Authorization 中的主要对象有 Authorization、Function、agentId 和 Qualifier。其他对象还有 Function Type 和 Qualifier type。

Id 是在外部定义的，具体实现必须保证 Id 的唯一性。

有两种方法创建 Authorization。一种是使用 agentId、Function 和 Qualifier。另一种是添加生效日期（effective date）和失效日期（expiration date）。如果没有生效日期和失效日期，则生效日期为当天，且永不失效。

cn.edu.celtsc.intf.common.authorization.AuthorizationManager 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.base.CeltsType			
cn.edu.celtsc.intf.common.id.Id			
cn.edu.celtsc.intf.common.authorization.Qualifier			
cn.edu.celtsc.intf.common.authorization.Function			
cn.edu.celtsc.intf.common.authorization.Authorization			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
agentExists	boolean	agentId: Id	用户是否存在
createAuthorization	Authorization	agentId: Id, functionId: Id, qualifierId: Id	创建 Authorization
createDatedAuthorization	Authorization	agentId: Id, functionId: Id, qualifierId: Id, effectiveDate: long, expirationDate: long	创建带日期的 Authorization
createFunction	Function	functionId: Id, displayName: String, description: String,	创建 Function

		functionType: CeltsType, qualifierHierarchyId: Id	
createQualifier	Qualifier	qualifierId: Id, displayName: String, description: String, qualifierType: CeltsType, parentId: Id	创建 Qualifier
createRootQualifier	Qualifier	qualifierId: Id, displayName: String, description: String, qualifierType: CeltsType, qualifierHierarchyId: Id	创建根 Qualifier
deleteAuthorization	void	authorization : Authorization	删除 Authorization
deleteFunction	void	functionId: Id	删除 Function
deleteQualifier	void	qualifierId: Id	删除 Qualifier
getAllAZs	AuthorizationIterator	agentId: Id, functionId: Id, qualifierId: Id, isActiveNowOnly: boolean	获取所有 Authorization
getAllAZsByFuncType	AuthorizationIterator	agentId: Id, functionType: CeltsType, qualifierId: Id, isActiveNowOnly: boolean	根据函数类型获取所有 Authorization
getAllUserAZs	AuthorizationIterator	functionId: Id, qualifierId: Id, isActiveNowOnly: boolean	获取所有用户 Authorization
getAllUserAZsByFuncType	AuthorizationIterator	functionType: CeltsType, qualifierId: Id, isActiveNowOnly: boolean	根据函数类型获取所有用户 Authorization
getExplicitAZs	AuthorizationIterator	agentId: Id, functionId: Id, qualifierId: Id, isActiveNowOnly: boolean	获取所有显式 Authorization
getExplicitAZsByFuncType	AuthorizationIterator	agentId: Id, functionType: CeltsType, qualifierId: Id, isActiveNowOnly: boolean	根据函数类型获取所有显式 Authorization
getExplicitUserAZs	AuthorizationIterator	functionId: Id, qualifierId: Id, isActiveNowOnly: boolean	获取所有显式用户 Authorization
getExplicitUserAZsByFuncType	AuthorizationIterator	functionType: CeltsType, qualifierId: Id, isActiveNowOnly: boolean	根据函数类型获取所有显式用户 Authorization

getExplicitUserAZsForImplicitAZ	AuthorizationIterator	implicitAuthorization : Authorization	为隐式授权获取显式用户 Authorization
getFunction	Function	functionId: Id	获取 Function
getFunctions	FunctionIterator	functionType: CeltsType	获取所有 Function
getFunctionTypes	TypeIterator	无	获取所有函数类型
getQualifier	Qualifier	qualifierId: Id	获取所有 Qualifier
getQualifierChildren	QualifierIterator	qualifierId: Id	获取限定器子结点
getQualifierDescendants	QualifierIterator	qualifierId: Id	获取限定器子孙结点
getQualifierHierarchies	IdIterator	无	获取限定器层次结构
getQualifierTypes	TypeIterator	无	获取所有限定器类型
getRootQualifiers	QualifierIterator	qualifierHierarchyId: Id	获取所有根限定器
getWhoCanDo	IdIterator	functionId: Id, qualifierId: Id	用户是否能在某上下文执行某函数
isAuthorized	boolean	agentId: Id, functionId: Id, qualifierId: Id	是否授权
isUserAuthorized	boolean	functionId: Id, qualifierId: Id	用户是否授权
supportsDesign	boolean	无	是否支持设计，即是否支持以下方法： createFunction, deleteFunction
supportsMaintenance	boolean	无	是否支持维护，即是否支持以下方法： createAuthorization, createDatedAuthorization, createQualifier, createRootQualifier, deleteAuthorization, deleteQualifier, getFunctionTypes, getQualifier, getQualifierChildren, getQualifierDescendants getQualifierHierarchies, getQualifierTypes, getRootQualifiers, getWhoCanDo

A.1.8 工作流 (workflow) 服务

工作流 (workflow) 服务提供对系统中的工作流提供支持。

A.1.8.1 Expression 接口

Expression 标识一个在工作步被执行之前必须满足的输入条件。

cn.edu.celtsc.intf.common.workflow.Expression 接口定义

```
import: java.io.Serializable
       cn.edu.celtsc.intf.common.base.CeltsType
type: public
extends: Serializable
```

成员	返回值/类型	参数说明	说明
getDescription	String	无	获取该 Expression 的描述
getDisplayName	String	无	获取该 Expression 的显示名字
getType	CeltsType	无	获取该 Expression 的类型

A.1.8.2 ExpressionIterator 接口

ExpressionIterator 为一组 Expression 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.workflow.ExpressionIterator 接口定义

```
import: java.io.Serializable
       cn.edu.celtsc.intf.common.workflow.Expression
type: public
extends: Serializable
```

成员	返回值/类型	参数说明	说明
hasNextExpression	boolean	无	是否还有下一个 Expression
nextExpression	Expression	无	取下一个 Expression

A.1.8.3 Step 接口

工作步 (Step) 是进程 (Process) 中的关键元素。Processes 由若干个 Step 组成。一个 Step 包括与其他 Step 的连接、输入条件和输出状态。

cn.edu.celtsc.intf.common.workflow.Step 接口定义

```
import: java.io.Serializable
       cn.edu.celtsc.intf.common.base.CeltsType
       cn.edu.celtsc.intf.common.id.Id
       cn.edu.celtsc.intf.common.workflow.Expression
type: public
extends: Serializable
```

成员	返回值/类型	参数说明	说明
addPredecessor	void	stepId: Id	获取前驱 Step
getDescription	String	无	获取描述
getDisplayName	String	无	获取显示名字
getId	Id	无	获取 Id
getInputConditions	ExpressionIterator	无	获取输入条件
getOutputStates	StringIterator	无	获取输出状态
getPredecessors	StepIterator	无	获取所有前驱
getRoleId	Id	无	获取与此 Step 相关联的角色 Id
getSuccessors	StepIterator	无	获取所有后继
isInitial	boolean	无	该 Step 是否为 Process 的第一个
isTerminal	boolean	无	该 Step 是否为 Process 的最后一个
removePredecessor	void	stepId: Id	移除前驱
updateDescription	void	description : String	更新描述

updateDisplayName	void	displayName : String	更新显示名字
updateInputConditions	void	inputConditions : Expression[]	更新输入条件
updateOutputStates	void	outputStates : String[]	更新输出状态
updateRoleId	void	roleId: Id	更新角色 Id

A.1.8.4 StepIterator 接口

StepIterator 为一组 Step 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.workflow.StepIterator 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.workflow.Step			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextStep	boolean	无	是否还有下一个 Step
nextStep	Step	无	取下一个 Step

A.1.8.5 WorkEvent 接口

一个 WorkEvent 是 Step 的一个输出状态，与一个 Agent 和时间戳相关联。

cn.edu.celtsc.intf.common.workflow.WorkEvent 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.workflow.Step			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
getAgentId	Id	无	获取执行该事件的用户 Id
getOutputState	String	无	获取输出状态
getStep	Step	无	获取 Step
getTimestamp	long	无	获取该事件发生的时间戳

A.1.8.6 WorkEventIterator 接口

WorkEventIterator 为一组 WorkEvent 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.workflow.WorkEventIterator 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.workflow.WorkEvent			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextWorkEvent	boolean	无	是否还有下一个 WorkEvent
nextWorkEvent	WorkEvent	无	取下一个 WorkEvent

A.1.8.7 Work 接口

Work 是 Step 的集合，与一个或多个角色相关联。

cn.edu.celtsc.intf.common.workflow.Work 接口定义

import: java.io.Serializable cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.workflow.Step cn.edu.celtsc.intf.common.workflow.WorkEvent cn.edu.celtsc.intf.common.base.CeltsType type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
getDescription	String	无	获取描述
getDisplayName	String	无	获取显示名字
getHistory	WorkEventIterator	无	获取历史
getId	Id	无	获取 Id
getNextSteps	StepIterator	无	获取所有角色的后继 Step
getNextStepsForRole	StepIterator	roleId: Id	获取某角色的后继 Step
getProperties	PropertiesIterator	无	获取属性
getPropertiesByType	Properties	propertiesType: CeltsType	获取某类型的属性
getPropertyTypes	TypeIterator	无	获取所有属性类型
getQualifierId	Id	无	获取限定器的 Id
updateDescription	void	description: String	更新描述
updateDisplayName	void	displayName: String	更新显示名字
updateStepOutputState	void	outputState: String	更新输出状态

A. 1. 8. 8 WorkIterator 接口

WorkIterator 为一组 Work 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.workflow.WorkIterator 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.workflow.Work type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextWork	boolean	无	是否还有下一个 Work
nextWork	Work	无	取下一个 Work

A. 1. 8. 9 Process 接口

Process 是一个有组织的 Step 集合，包括一个初始 Step 和其它后继 Step。

cn.edu.celtsc.intf.common.workflow.Process 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.base.CeltsType cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.workflow.Step cn.edu.celtsc.intf.common.workflow.Work type: public			

extends: Serializable			
成员	返回值/类型	参数说明	说明
createInitialStep	Step	displayName: String, description: String, roleId: Id	创建初始 Step
createStep	Step	predecessorStepId: Id displayName: String, description: String, roleId: Id	创建 Step
createWork	Work	displayName: String, description: String, qualifierId: Id	创建 Work
deleteStep	void	stepId: Id	删除 Step
deleteWork	void	workId: Id	删除 Work
getAllWork	WorkIterator	无	获取所有 Work
getAvailableWork	WorkIterator	无	获取所有有效 Work
getAvailableWorkFor Role	WorkIterator	roleId: Id	获取所有对某角色有效 Work
getAvailableWorkFor Step	WorkIterator	stepId: Id	获取所有对某 Step 有效 Work
getDescription	String	无	获取描述
getDisplayName	String	无	获取显示名字
getId	Id	无	获取 Id
getStep	Step	stepId: Id	获取 Step
getSteps	StepIterator	无	获取所有 Step
getType	CeltsType	无	获取类型
getUnfinishedWork	WorkIterator	无	获取未完成 Work
getUnfinishedWorkF orRole	WorkIterator	roleId: Id	获取某角色未完成 Work
getUnfinishedWorkF orStep	WorkIterator	stepId: Id	获取某 Step 未完成 Work
getWork	Work	workId: Id	获取 Work
haltWork	void	workId: Id	停止 Work
isEnabled	boolean	无	Process 是否有效
resumeWork	void	workId: Id	恢复停止的 Work
updateDescription	void	description: String	更新描述
updateDisplayName	void	displayName: String	更新显示名字
updateEnabled	void	enabled: boolean	更新有效状态

A.1.8.10 ProcessIterator 接口

ProcessIterator 为一组 Process 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.workflow.ProcessIterator 接口定义

```
import: java.io.Serializable
        cn.edu.celtsc.intf.common.workflow.Process
```

type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextProcess	boolean	无	是否还有下一个 Process
nextProcess	Process	无	取下一个 Process

A.1.8.11 WorkflowManager 接口

WorkflowManager 负责管理 Process，发现未完成的 Work。

cn.edu.celtsc.intf.common.workflow.WorkflowManager 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.base.CeltsType			
cn.edu.celtsc.intf.common.id.Id			
cn.edu.celtsc.intf.common.workflow.Expression			
cn.edu.celtsc.intf.common.workflow.Step			
cn.edu.celtsc.intf.common.workflow.Work			
cn.edu.celtsc.intf.common.workflow.Process			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
copyProcess	Process	originalProcessId: Id, newDisplayName : String, newDescription: String	拷贝一个已知的 Process，并赋予新的显示名称以及描述。
createProcess	Process	displayName: String, description: String	创建 Process
deleteProcess	void	processId: Id	删除 Process
getCompletedWork	WorkIterator	processId: Id	获取未完成 Work
getExpressions	ExpressionIterator	无	获取所有 Expression
getExpressionsByType	ExpressionIterator	expressionType : CeltsType	获取某类型 Expression
getExpressionTypes	TypeIterator	无	获取所有 Expression 类型
getOutputStates	StringIterator	无	获取输出状态
getProcess	Process	processId: Id	获取 Process
getProcesses	ProcessIterator	无	获取所有 Process
getUnfinishedWork	WorkIterator	processId: Id	获取未完成 Work
supportsDesign	boolean	无	是否支持设计，即是否支持以下方法： copyProcess, createProcess, deleteProcess
supportsMaintenance	boolean	无	是否支持维护，即是否支持以下方法： getCompletedWork, getExpressions, getExpressionsByType,

			getExpressionTypes, getOutputStates, getProcess, getProcesses, getUnfinishedWork
--	--	--	--

A.1.9 XML服务

xml 服务对 xml 数据的访问细节进行封装。通过 xml 服务，用户不必了解 xml 文档的解析，即可得到一个文档的对象模型，该模型能够间接对 xml 数据进行访问和修改，减少了直接解析 xml 带来的复杂性。在 xml 服务中还提供一个格式化处理的接口——XMLFormatter，该接口可以对 XML 数据进行格式化，添加缩进，换行等等，提高 XML 数据的可读性。

A.1.9.1 XMLParser 接口

XMLParser 接口负责解析 XML 数据（可能是从文件中读取或者是程序生成的一个字符串），得到 XML 文档的根，以此作为对象模型的入口即可访问文档中的所有数据。

cn.edu.celtsc.intf.common.xml.XMLParser 接口定义			
import: cn.edu.celtsc.intf.common.file.File cn.edu.celtsc.intf.common.file.Path			
type: public			
成员	返回值/类型	参数说明	说明
getSpecification	String	无	返回该解析器所对应的规范，如 ims-md, ims-cp 等
parse	Object	docString: String 将被解析的文件路径，可以是相对或者绝对路径	返回被解析文档的入口对象，可以根据该解析器对应的规范转型为特定的对象
parse	Object	file: File 将被解析的文件实例	返回被解析文档的入口对象，可以根据该解析器对应的规范转型为特定的对象
parse	Object	path: Path 将被解析的文件路径	返回被解析文档的入口对象，可以根据该解析器对应的规范转型为特定的对象

A.1.9.2 XMLFormatter 接口

XMLFormatter 接口用于对 XML 数据进行简单的排版。

cn.edu.celtsc.intf.common.xml.XMLParser 接口定义			
import: cn.edu.celtsc.intf.common.file.Path			
type: public			
成员	返回值/类型	参数说明	说明
format	String	xmlStr: String	对输入的 XML 数据字符串进行格式化，返回结果字符串
format	void	xmlStr: String filePath: Path	对输入的 XML 字符串进行格式化，格式化后的结果存入 filePath 指定文件

A.1.9.3 XMLFactory 接口

通过 XMLFactory 可以获得 XMLParser 和 XMLFormatter 的实例。

cn.edu.celtsc.intf.common.xml.XMLFactory 接口定义			
type: public			
成员	返回值/类型	参数说明	说明
getParser	XMLParser	specification: String	获得一个特定规范的 XMLParser，如

			ims 元数据规范的 Parser
getFormatter	XMLFormatter	无	获取一个 XMLFormatter 实例

A.2 教育服务接口

元数据、学习内容包装服务接口中都集成了 Serialize 接口，用于按照给定的绑定规范输出对应数据元素的字符串。

cn.edu.celtsc.intf.education.Serialize 接口定义			
type: public			
常量	类型	说明	
XML_BINDING	String	Xml 绑定类型	
成员	返回值/类型	参数说明	说明
getString	String	type: String	按照绑定方法返回字符串
isSupport	boolean	type: String	是否支持某种绑定方法

A.2.1 评价 (assessment) 服务

assessment 服务可以对系统中任意类型的条目 (Item) 或由条目组成的区域 (Section) 进行评价。

A.2.1.1 Item 接口

Item 包括问题、回复集、答案以及其它相关指导和媒体。这些元素都被包含在一个 Data 对象中，该对象的内容没有在接口规范中确切说明。Item 由其对应的类型来区分。ItemType 对一个应用程序来说是有意义的，但在接口规范中没有确切定义。与 Item 相关联的全局唯一的 Id 由 AssessmentManager 的 createItem 方法设置。

cn.edu.celtsc.intf.common.assessment.Item 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.base.CeltsType			
cn.edu.celtsc.intf.common.id.Id			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
getData	Serializable	无	获取该 Item 的 Data
getDescription	String	无	获取该 Item 的描述
getDisplayName	String	无	获取该 Item 的显示名字
getId	Id	无	获取该 Item 的 Id
getItemType	CeltsType	无	获取该 Item 的类型 ItemType
getProperties	PropertiesIterator	无	获取该 Item 的所有属性配置
getPropertiesByType	Properties	propertiesType: CeltsType	获取该 Item 的某种类型的属性配置
getPropertyTypes	TypeIterator	无	获取该 Item 的所有属性类型
updateData	void	data : Serializable	更新该 Item 的 Data
updateDescription	void	description : String	更新该 Item 的描述
updateDisplayName	void	displayName: String	更新该 Item 的显示名字

A.2.1.2 ItemIterator 接口

ItemIterator 为一组 Item 提供顺序访问，每次只能访问一个。所有迭代器的目的就是在接口规范中

为通用类型返回多个值，而不使用数组。如果返回值数据量很大，或者是远程访问，返回数组是不合适的。迭代器不允许使用序号（index）访问值，而只允许顺序访问。也就是说不能回头访问曾经访问过的数据。

cn.edu.celtsc.intf.common.assessment.ItemIterator 接口定义

```
import: java.io.Serializable
       cn.edu.celtsc.intf.common.assessment.Item
type: public
extends: Serializable
```

成员	返回值/类型	参数说明	说明
hasNextItem	boolean	无	是否还有下一个 Item
nextItem	Item	无	取下一个 Item

A.2.1.3 ItemTaken 接口

ItemTaken包含Item、提交的回答以及这些回答的Evaluation。

cn.edu.celtsc.intf.common.assessment.ItemTaken 接口定义

```
import: java.io.Serializable
       cn.edu.celtsc.intf.common.base.CeltsType
       cn.edu.celtsc.intf.common.id.Id
       cn.edu.celtsc.intf.common.assessment.Item
       cn.edu.celtsc.intf.common.assessment.SectionTaken
       cn.edu.celtsc.intf.common.assessment.EvaluationIterator
type: public
extends: Serializable
```

成员	返回值/类型	参数说明	说明
createEvaluation	无	evaluationType: CeltsType	为该 Item 按照指定的 CeltsType 创建 Evaluation
deleteEvaluation	无	evaluationId: Id	删除指定的 Evaluation
getData	可序列化对象	无	获取 ItemTaken 的数据
getDisplayName	字符串	无	获取 ItemTaken 的显示名称
getEvaluations	EvaluationIterator	无	获取该 Item 的所有 Evaluation
getEvaluationByType	EvaluationIterator	evaluationType: CeltsType	获取指定类型的 Evaluation
getId	Id	无	获取该 ItemTaken 的唯一 Id
getItem	Item	无	获取 ItemTaken 所属的 Item
getSectionTaken	SectionTaken	无	获取 ItemTaken 所属的 SectionTaken
updateData	无	data: 可序列化对象	为 ItemTaken 更新数据
updateDisplayName	无	displayNames: 字符串	为 ItemTaken 更新显示名称

A.2.1.4 ItemTakenIterator 接口

ItemTakenIterator为一组ItemTaken提供顺序访问，每次只能够访问一个

cn.edu.celtsc.intf.common.assessment.ItemTakenIterator 接口定义

```
import: java.io.Serializable
```

cn.edu.celtsc.intf.common.assessment.ItemTaken			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextItemTaken	布尔值	无	是否还有下一个 ItemTaken
nextItemTaken	ItemTaken	无	返回下一个 ItemTaken

A.2.1.5 Evaluation 接口

Evaluation是评估报告的接口。

cn.edu.celtsc.intf.common.assessment.ItemTaken 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.base.CeltsType			
cn.edu.celtsc.intf.common.id.Id			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
getData	可序列化对象	无	从 Evaluation 中取得数据
getDisplayName	字符串	无	获取 Evaluation 的显示名字
getId	Id	无	从 Evaluation 中获得唯一 Id
getModifiedDate	长整数	无	获取 Evaluation 的修改时间
getObjectTaken	Id	无	获取 objectTaken 的 Id
getType	CeltsType	无	获取 Evaluation 的 CeltsType
updateData	无	data:可序列化对象	更新 Evaluation 的数据
updateDisplayName	无	displayName:字符串	更新 Evaluation 的显示名字

A.2.1.6 EvaluationIterator 接口

EvaluationIterator为一组Evaluation提供顺序访问，每次只能够访问一个。

cn.edu.celtsc.intf.common.assessment.EvaluationIterator 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.assessment.Evaluation			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextEvaluation	布尔值	无	是否还有下一个 Evaluation
nextEvaluation	Evaluation	无	返回下一个 Evaluation

A.2.1.7 Section 接口

Section 包括 0 个或多个 Item 或 Section。Item 或 Section 按照先入先出(FIFO)方式加入一个 Section。SectionType 对一个应用程序来说是有意义的，但在接口规范中没有确切定义。与 Section 相关联的全局唯一的 Id 由 AssessmentManager 的 createSection 方法设置。

cn.edu.celtsc.intf.common.assessment.Section 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.base.CeltsType			
cn.edu.celtsc.intf.common.id.Id			
cn.edu.celtsc.intf.common.assessment.Item			

type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
addItem	void	item: Item	向该 Section 中添加一个 Item
addSection	void	section : Section	向该 Section 中添加一个 Section
getData	Serializable	无	获取该 Section 的 Data
getDescription	String	无	获取该 Section 的描述
getDisplayName	String	无	获取该 Section 的显示名字
getId	Id	无	获取该 Section 的 Id
getItems	ItemIterator	无	获取该 Section 中所有 Item
getProperties	PropertiesIterator	无	获取该 Section 的所有属性配置
getPropertiesByType	Properties	propertiesType: CeltsType	获取该 Section 的某种类型的属性配置
getPropertyTypes	TypeIterator	无	获取该 Section 的所有属性类型
getSections	SectionIterator	无	获取该 Section 中所有 Section
getSectionType	CeltsType	无	获取该 Section 的类型
orderItems	void	items: Item[]	改变该 Section 中 Item 的顺序
orderSections	void	sections : Section[]	改变该 Section 中 Section 的顺序
removeItem	void	itemId: Id	从该 Section 中删除一个 Item
removeSection	void	sectionId: Id	从该 Section 中删除一个 Section
updateData	void	data : Serializable	更新该 Section 的 Data
updateDescription	void	description : String	更新该 Section 的描述
updateDisplayName	void	displayName: String	更新该 Section 的显示名字

A. 2. 1. 8 SectionIterator 接口

SectionIterator 为一组 Section 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.assessment.SectionIterator 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.assessment.Section			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextSection	boolean	无	是否还有下一个 Section
nextSection	Section	无	取下一个 Section

A. 2. 1. 9 SectionTaken 接口

SectionTaken 含有 Section 里面的所有 Item 以及 Evaluation。

cn.edu.celtsc.intf.common.assessment.SectionTaken 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.base.CeltsType			

cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.assessment.Item cn.edu.celtsc.intf.common.assessment.ItemTaken cn.edu.celtsc.intf.common.assessment.ItemTakenIterator cn.edu.celtsc.intf.common.assessment.Section cn.edu.celtsc.intf.common.assessment.Evaluation cn.edu.celtsc.intf.common.assessment.EvaluationIterator cn.edu.celtsc.intf.common.assessment.AssessmentTaken type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
createEvaluation	Evaluation	evaluationType:CeltsType	按照指定的 CeltsType 为 Section 创建 Evaluation
createItemTaken	ItemTaken	item:Item	在 SectionTaken 的基础上创建 ItemTaken
deleteEvaluation	无	evaluationId:Id	在 Section 中删除指定的 Evaluation
deleteItemTaken	无	itemTakenId:Id	删除一个 ItemTaken
getAssessmentTaken	AssessmentTaken	无	获取 SectionTaken 所属的 AssessmentTaken
getData	可序列化对象	无	获取 SectionTaken 的数据
getDisplayName	字符串	无	获取 SectionTaken 的显示名字
getEvaluations	EvaluationIterator	无	获取 Section 中所有的 Evaluation
getEvaluationsByType	EvaluationIterator	evaluationType:CeltsType	获取 Section 中指定 CeltsType 的 Evaluation
getId	Id	无	获取 SectionTaken 的唯一 Id
getItemsTaken	ItemTakenIterator	无	获取该 SectionTaken 的所有 ItemTaken
getSection	Section	无	获得创建 SectionTaken 的 Section
updateData	无	data:可序列化对象	更新 SectionTaken 的数据
updateDisplayName	无	displayName:字符串	更新 SectionTaken 的显示名称

A. 2. 1. 10 SectionTakenIterator 接口

SectionTakenIterator为一组SectionTaken提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.assessment.SectionTakenIterator 接口定义 import: java.io.Serializable cn.edu.celtsc.intf.common.assessment.SectionTaken type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextSectionTaken	布尔值	无	是否还有下一个 SectionTaken
nextSectionTaken	SectionTaken	无	取下一个 SectionTaken

A. 2. 1. 11 Assessment 接口

Assessment 包括 0 个或多个 Section。Section 按照先入先出（FIFO）方式加入一个 Assessment。AssessmentType 对一个应用程序来说是有意义的，但在接口规范中没有确切定义。Assessment 还包括一个 Topic。一个 Assessment 与一个 AssessmentPublished 相关联，该 AssessmentPublished 包括一个 Assessment 和与这个 Assessment 的有效性相关的数据集。一个 Assessment 还与一个 AssessmentTaken 相关联，该 AssessmentTaken 包括一个 Assessment 和与该 Assessment 集成的学生相关的数据集。

cn.edu.celtsc.intf.common.assessment.Assessment 接口定义			
import: java.io.Serializable			
cn.edu.celtsc.intf.common.base.CeltsType			
cn.edu.celtsc.intf.common.id.Id			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
addSection	void	section : Section	向该 Assessment 中添加一个 Section
getAssessmentType	CeltsType	无	获取该 Assessment 的类型
getData	Serializable	无	获取该 Assessment 的 Data
getDescription	String	无	获取该 Assessment 的描述
getDisplayName	String	无	获取该 Assessment 的显示名字
getId	Id	无	获取该 Assessment 的 Id
getItems	ItemIterator	无	获取该 Section 中所有 Item
getProperties	PropertiesIterator	无	获取该 Assessment 的所有属性配置
getPropertiesByType	Properties	propertiesType: CeltsType	获取该 Assessment 的某种类型的属性配置
getPropertyTypes	TypeIterator	无	获取该 Assessment 的所有属性类型
getSections	SectionIterator	无	获取该 Assessment 中所有 Section
getTopic	String	无	获取该 Assessment 的 Topic
orderSections	void	sections : Section[]	改变该 Assessment 中 Section 的顺序
removeSection	void	sectionId: Id	从该 Assessment 中删除一个 Section
updateData	void	data : Serializable	更新该 Assessment 的 Data
updateDescription	void	description : String	更新该 Assessment 的描述
updateDisplayName	void	displayName: String	更新该 Assessment 的显示名字
updateTopic	void	topic: String	更新该 Assessment 的 Topic

A. 2. 1. 12 AssessmentIterator 接口

AssessmentIterator 为一组 Assessment 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.assessment.AssessmentIterator 接口定义
import: java.io.Serializable
cn.edu.celtsc.intf.common.assessment.Assessment
type: public

extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextAssessment	boolean	无	是否还有下一个 Assessment
nextAssessment	Assessment	无	取下一个 Assessment

A. 2. 1. 13 AssessmentPublished 接口

cn.edu.celtsc.intf.common.assessment.AssessmentPublished 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.base.CeltsType cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.assessment.Assessment			
type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
createAssessmentTaken	AssessmentTaken	agentId: Id	创建一个 AssessmentPublished
deleteAssessmentTaken	void	assessmentTakenId: Id	删除一个 AssessmentPublished
getAssessment	Assessment	无	获取要发布的 Assessment
getAssessmentsTaken	AssessmentTakenIterator	无	获取所有的 AssessmentTaken
getAssessmentsTakenBy	AssessmentTakenIterator	agentId: Id	获取所有与某个学生的 agentId 相关联的 Assessment
getCourseSectionId	Id	无	获取与该 AssessmentPublished 相关联的 CourseSection 的 Id
getData	Serializable	无	获取该 AssessmentPublished 的 Data
getDate	long	无	获取该 AssessmentPublished 的发布日期
getDescription	String	无	获取该 AssessmentPublished 的描述
getDisplayName	String	无	获取该 AssessmentPublished 的显示名字
getId	Id	无	获取该 AssessmentPublished 的 Id
updateCourseSectionId	void	courseSectionId: Id	更新该 AssessmentPublished 的 CourseSectionId
updateData	void	data : Serializable	更新该 AssessmentPublished 的 Data
updateDescription	void	description: String	更新该 AssessmentPublished 的描述
updateDisplayName	void	displayName: String	更新该 AssessmentPublished 的显示名字

A. 2. 1. 14 AssessmentPublishedIterator 接口

AssessmentPublishedIterator 为一组 AssessmentPublished 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.assessment.AssessmentPublishedIterator 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.assessment.AssessmentPublished			
type: public extends: Serializable			

成员	返回值/类型	参数说明	说明
hasNextAssessmentPublished	boolean	无	是否还有下一个 AssessmentPublished
nextAssessmentPublished	AssessmentPublished	无	取下一个 AssessmentPublished

A. 2. 1. 15 AssessmentTaken 接口

cn.edu.celtsc.intf.common.assessment.AssessmentTaken 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.base.CeltsType cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.assessment.Section			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
createEvaluation	Evaluation	evaluationType: CeltsType	为该 Assessment 创建一个特定类型的评估报告
createSectionTaken	SectionTaken	section : Section	在一个 AssessmentPublished 和学生的 AgentId 的基础上创建一个 SectionTaken
deleteEvaluation	void	evaluationId: Id	删除一个评估报告
deleteSectionTaken	void	sectionTakenId: Id	删除一个 SectionTaken
getAgentId	Id	无	获取接受评价的学生 agentId
getAssessmentPublished	AssessmentPublished	无	获取采用的 AssessmentPublished
getData	Serializable	无	获取该 AssessmentTaken 的 Data
getDate	long	无	获取该 AssessmentTaken 的执行日期
getDisplayName	String	无	获取该 AssessmentTaken 的显示名字
getEvaluations	EvaluationIterator	无	获取该 AssessmentTaken 的所有评估报告
getEvaluationsByType	EvaluationIterator	evaluationType: CeltsType	获取该 AssessmentTaken 的某种类型的评估报告
getId	Id	无	获取该 AssessmentTaken 的 Id
getSectionsTaken	SectionTakenIterator	无	获取该 AssessmentTaken 的所有 SectionsTaken
updateData	void	data : Serializable	更新该 AssessmentTaken 的 Data
updateDisplayName	void	displayName: String	更新该 AssessmentTaken 的显示名字

A. 2. 1. 16 AssessmentTakenIterator 接口

AssessmentTakenIterator 为一组 AssessmentTaken 提供顺序访问，每次只能访问一个。

cn.edu.celtsc.intf.common.assessment.AssessmentTakenIterator 接口定义

import: java.io.Serializable cn.edu.celtsc.intf.common.assessment.AssessmentTaken			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextAssessmentTaken	boolean	无	是否还有下一个 AssessmentTaken
nextAssessmentTaken	AssessmentTaken	无	取下一个 AssessmentTaken

A. 2. 1. 17 AssessmentManager 接口

AssessmentManager 主要负责创建和删除 Assessment、AssessmentPublished、Item 和 Section。以及获取 Assessment、AssessmentPublished、Item、Section 和各种实现类型的值。

cn.edu.celtsc.intf.common.assessment.AssessmentManager 接口定义			
import: java.io.Serializable cn.edu.celtsc.intf.common.assessment.Assessment cn.edu.celtsc.intf.common.assessment.AssessmentPublished cn.edu.celtsc.intf.common.assessment.Item cn.edu.celtsc.intf.common.assessment.Section cn.edu.celtsc.intf.common.base.CeltsType cn.edu.celtsc.intf.common.id.Id			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
createAssessment	Assessment	name: String, description : String, assessmentType: CeltsType	创建一个新的 Assessment 并将其加入 Assessment 库
createAssessmentPublished	AssessmentPublished	assessment : Assessment	根据 Assessment 创建一个 AssessmentPublished
createItem	Item	name: String, description : String, itemType : CeltsType	创建一个 Item
createSection	Section	name: String, description : String, sectionType : CeltsType	创建一个 Section
deleteAssessment	void	assessmentId: Id	从 Assessment 库中删除一个 Assessment
deleteAssessmentPublished	void	assessmentPublishedId: Id	删除一个 AssessmentPublished
deleteItem	void	itemId: Id	删除一个 Item
deleteSection	void	sectionId: Id	删除一个 Section

getAssessment	Assessment	assessmentId: Id	根据 Id 获取一个 Assessment
getAssessmentPublished	AssessmentPublished	assessmentPublishedId: Id	根据 Id 获取一个 AssessmentPublished
getAssessments	AssessmentIterator	无	获取所有的 Assessment
getAssessmentsByType	AssessmentIterator	assessmentType: CeltsType	获取特定类型的 Assessment
getAssessmentsPublished	AssessmentPublishedIterator	无	获取所有的 AssessmentPublished
getAssessmentTypes	TypeIterator	无	获取所有的 Assessment 类型
getItem	Item	itemId: Id	根据 Id 获取一个 Item
getItems	ItemIterator	无	获取所有的 Item
getItemsByType	ItemIterator	itemType : CeltsType	获取特定类型的 Item
getItemTypes	TypeIterator	无	获取所有的 Item 类型
getSection	Section	sectionId: Id	根据 Id 获取一个 Section
getSections	SectionIterator	无	获取所有的 Section
getSectionsByType	SectionIterator	sectionType : CeltsType	获取特定类型的 Section
getSectionTypes	TypeIterator	无	获取所有的 Section 类型

A. 2. 2 课程管理 (coursemanagement) 服务

A. 2. 2. 1 Course 接口

Course 接口是课程模型的基本接口，按照不同的标准需求可以以此接口进行扩展，Course 接口所提供的是课程最一般化的信息，更具体化的课程在 CourseOffering 接口中提供。

cn.edu.celtsc.intf.education.coursemanagement.Course 接口定义			
import: cn.edu.celtsc.intf.common.base.CeltsType cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.base.PropertiesIterator			
type: public			
成员	返回值/类型	参数说明	说明
getCredits	float	无	获得课程学分数
setCredits	void	credits: float	设置课程学分
getCourseType	CeltsType	无	获得课程类型
setCourseType	void	courseType: CeltsType	设置课程类型
getDescription	String	无	获得课程的基本描述
setDescription	void	description: String	设置课程的描述信息
getDisplayName	String	无	获得课程的显示名称
setDisplayName	void	displayName: String	设置课程的显示名称
getId	Id	无	获得课程 ID
getProperties	PropertiesIterator	无	获得课程相关的配置属性
getStatus	CeltsType	无	获得课程的状态
setStatus	void	statusType: CeltsType	设置课程状态

getTitle	String	无	获得课程标题
setTitle	void	title: String	设置课程标题

A. 2. 2. 2 CourseOffering 接口

Course 所提供的课程是十分泛化的课程，而 CourseOffering 则是针对某个具体学期开设的具体一门课程，对于一个 Course 可以有若干个 CourseOffering，每个 CourseOffering 有自己的名册等信息并有与之相关联的课程章节内容，课程的片断模型由 CourseSection 接口定义。

cn.edu.celtsc.intf.education.coursemanagement.CourseOffering 接口定义			
import: cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.base.IdIterator cn.edu.celtsc.intf.common.base.PropertiesIterator cn.edu.celtsc.intf.common.base.Iterator cn.edu.celtsc.intf.common.base.CeltsType java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
addAsset	void	assetId: Id	向一门课程添加素材
removeAsset	void	assetId: Id	从课程中删除一个素材
getAssets	IdIterator	无	获得课程的全部素材
addStudent	void	studentId: Id regType: CeltsType	向名册中添加一个学生，并为其注册类型赋值。
removeStudent	void	studentId: Id	从该课程注册的学生名册中删除一个学生
changeRegType	void	studentId: Id regType: CeltsType	变更一个学生的注册类型
getRoster	Iterator	无	得到课程的学生名册
getCourse	Course	无	得到包括此 CourseOffering 的课程实例
getId	Id	无	得到当前 CourseOffering 的 ID
setDescription	void	description: String	设置描述信息
getDescription	String	无	获得描述信息
getDisplayName	String	无	获得显示名称
setDisplayName	void	displayName: String	设置显示名称
getProperties	PropertiesIterator	无	获得配置信息
getStatus	CeltsType	无	获得 CourseOffering 的状态
setStatus	void	statusType: CeltsType	设置 CourseOffering 状态
getTitle	String	无	获得 CourseOffering 标题
setTitle	void	title: String	设置 CourseOffering 标题

A. 2. 2. 3 CourseManager 接口

CourseManager 负责创建删除课程，并且可获得课程成绩的管理及分组管理。

cn.edu.celtsc.intf.education.coursemanagement.CourseManager 接口定义			
import: cn.edu.celtsc.intf.common.base.Manager cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.base.CeltsType			

type: public extends: Manager			
成员	返回值/类型	参数说明	说明
createCourse	Course	title: String number: String description: String courseType: CeltsType courseStatusType: CeltsType credits: float	创建一门课程，包括课程标题，课容量，描述，课程和状态类型，学分数。返回一个课程实例，拥有课程名、描述和统一的 ID。
deleteCourse	void	courseId: Id	根据 ID 删除一门课程

A. 2. 3 成员管理（membershipmanagement）服务

membershipmanagement 服务负责系统中所有成员关系的管理，包括成员的创建、删除、分组、查询等。

A. 2. 3. 1 Member 接口

Member 是数字教育公共服务平台中虚拟成员的抽象，包括的属性有 DisplayName, Id, CeltsType 和 Properties, Member 通过 MemberManager 统一创建。

cn.edu.celtsc.intf.education.membershipmanagement.Member 接口定义			
import: cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.base.Properties cn.edu.celtsc.intf.common.base.TypeIterator cn.edu.celtsc.intf.common.base.PropertiesIterator cn.edu.celtsc.intf.common.base.CeltsType java.io.Serializable			
type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
getDisplayName	String	无	获得成员显示的名称
getId	Id	无	获得成员 ID
getType	CeltsType	无	获得成员类型
getPropertiesByType	Properties	propertiesType: CeltsType	获得同类型的所有属性
getPropertyTypes	TypeIterator	无	获得属性的类型
getProperties	PropertiesIterator	无	获得属性

A. 2. 3. 2 MemberIterator 接口

MemberIterator 接口用于 Member 的查询等操作，是获取一组成员对象的迭代器接口。

cn.edu.celtsc.intf.education.membershipmanagement.MemberIterator 接口定义			
import: java.io.Serializable			
type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextMember	boolean	无	判断是否集合中是否还有成员
nextMember	Member	无	获得下一个成员实例

A. 2. 3. 3 MemberManager 接口

MemberManager 接口提供创建，删除成员，以及得到成员实例的功能。

cn.edu.celtsc.intf.education.membershipmanagement.MemberManager 接口定义			
<pre>import: cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.base.Manager cn.edu.celtsc.intf.common.base.Properties cn.edu.celtsc.intf.common.base.TypeIterator cn.edu.celtsc.intf.common.base.CeltsType java.io.Serializable type: public extends: Serializable</pre>			
成员	返回值/类型	参数说明	说明
createMember	Member	displayName: String memberType: CeltsType properties: Properties	通过 display name, type, properties 创建一个成员
deleteMember	void	id: Id	通过 ID 删除指定成员
getMember	Member	id: Id	根据唯一的 id 获得相应的成员实例, 由于其它的属性并不唯一, 本标准不提供其它获取实例的方式
getMembers	MemberIterator	无	通过迭代器获得全部成员
getMemberTypes	TypeIterator	无	获得成员的所有类型
getPropertyTypes	TypeIterator	无	获得所有属性类型
getMembersByType	MemberIterator	memberType: CeltsType	获得指定类型的所有成员实例
getMemberSearchTypes	TypeIterator	无	获得全部成员的查询类型
getMembersBySearch	MemberIterator	searchCriteria: Serializable memberSearchType: CeltsType	按查询类型和查询标准检索成员

A. 2. 4 组管理 (groupmanagement) 服务

groupmanagement 服务负责系统中组的管理, 分组既可以用于成员, 也可以用于学习资源, 因此将组作为一种独立的服务进行管理而不是作为成员的特例, 可以让分组行为在学习过程中有更广的使用空间。

A. 2. 4. 1 Group 接口

组严格意义上说也可以看作一种成员, 在 OSID 的早期版本内, 组合成员是相同对待的, 但 v3 的新近更新意见是将组与成员分离开来, 以便于在其它的地方使用分组。组可以包括成员, 也可以有子组, 因此组接口中包括一些测试方法, 验证成员和组的包含关系。

cn.edu.celtsc.intf.education.groupmanagement.Group 接口定义
<pre>import: cn.edu.celtsc.intf.common.id.Id cn.edu.celtsc.intf.common.base.Properties cn.edu.celtsc.intf.common.base.TypeIterator cn.edu.celtsc.intf.common.base.PropertiesIterator cn.edu.celtsc.intf.education.membershipmanagement.Member cn.edu.celtsc.intf.education.membershipmanagement.MemberIterator</pre>

cn.edu.celtsc.intf.common.base.CeltsType			
type: public			
extends: Member			
成员	返回值/类型	参数说明	说明
setDescription	void	description: String	设置组描述, description是组特有的描述信息
getDescription	String	无	获得组描述信息
getId	Id	无	获得唯一的组id
getDisplayName	String	无	获得显示名称
getType	CeltsType	无	获得类型
add	void	memberOrGroup: Member	添加一个结点,可以是成员或者子组
remove	void	memberOrGroup: Member	删除一个结点
getMembers	MemberIterator	includeSubgroups: boolean	获得组的全部成员实例,输入参数可以确定是否递归查找子组
getGroups	GroupIterator	includeSubgroups: boolean	获得全部子组,输入参数可以确定是否递归查找子组
contains	boolean	memberOrGroup: Member searchSubgroups: boolean	验证一个组或成员是否属于当前的组
getPropertiesByType	Properties	propertiesType: CeltsType	获得同类型的所有属性
getPropertyTypes	TypeIterator	无	获得所有类型
getProperties	PropertiesIterator	无	获得所有属性

A.2.4.2 GroupIterator 接口

GroupIterator 接口用于 Member 的查询等操作,是获取一组成员对象的迭代器接口。

cn.edu.celtsc.intf.education.groupmanagement.GroupIterator 接口定义			
import: java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
hasNextGroup	boolean	无	判断是否集合中是否还有组
nextGroup	Group	无	获得下一个组实例

A.2.4.3 GroupManager 接口

GroupManager 接口提供创建,删除组,以及得到组实例的功能。

cn.edu.celtsc.intf.education.groupmanagement.GroupManager 接口定义			
import: cn.edu.celtsc.intf.common.id.Id			
cn.edu.celtsc.intf.common.base.Manager			
cn.edu.celtsc.intf.common.base.Properties			
cn.edu.celtsc.intf.common.base.TypeIterator			
cn.edu.celtsc.intf.common.base.CeltsType			

java.io.Serializable type: public extends: Manager			
成员	返回值/类型	参数说明	说明
createGroup	Group	displayName: String groupType: CeltsType description: String properties: Properties	通过 display name, type, description, properties 创建一个组
deleteGroup	void	id: Id	通过 ID 删除指定组
getGroup	Group	id: Id	根据唯一的 id 获得相应的组实例
getGroups	GroupIterator	无	通过迭代器获得全部组
getGroupTypes	TypeIterator	无	获得组的所有类型
getPropertyTypes	TypeIterator	无	获得所有属性类型
getGroupsByType	GroupIterator	groupType: CeltsType	获得指定类型的所有组实例
getGroupSearchTypes	TypeIterator	无	获得全部组的查询类型
getGroupsBySearch	GroupIterator	searchCriteria: Serializable groupSearchType: CeltsType	按查询类型和查询标准检索组

A. 2.5 编列 (sequencing) 服务

sequencing 服务提供的是与编列活动相关的操作及数据模型。在 CELTS, IMS 及 SCORM 等规范中均有编列规范部分。本标准给出的编列服务试图不将其接口绑定到某个具体规范, 而仅描述数字教育公共服务平台系统中编列活动所需要涉及的行为, 并为之设计接口。这些接口包括如编列规则模型, 编列活动及活动树模型, 也包括编列引擎, 编列解析器等。

A. 2.5.1 SequenceRule 接口

SequenceRule 接口是编列规则的基本数据模型, 其中给出的是对一个编列活动的评价函数, 具体的编列规则是由规范所确定的, 此处编列规则的模型仅仅是对上层开发的应用接口。

cn.edu.celtsc.intf.education.sequencing.SequenceRule 接口定义 import: cn.edu.celtsc. cn.edu. celtsc.intf.common java.io.Serializable type: public extends: Serializable			
成员	返回值/类型	参数说明	说明
evaluate	String	iType: CeltsType iThisActivity: SequenceActivity iRetry: boolean	根据编列活动, 类型参数, 获得一个编列规则的评价结果。

A. 2.5.2 SequenceActivity 接口

SequenceActivity 是编列活动接口, 也是编列活动的基本数据模型, 此接口提供类似活动树结点结构的管理操作。

cn.edu.celtsc.intf.education.sequencing.SequenceActivity 接口定义 import: cn.edu.celtsc.intf.common.id.Id java.util.Vector java.io.Serializable type: public			
---	--	--	--

extends: Serializable			
成员	返回值/类型	参数说明	说明
addChild	void	ioChild: SequenceActivity	添加一个孩子结点
setChildren	void	ioChildren: Vector iAll: boolean	为全部孩子结点设置
getChildren	Vector	iAll: boolean	得到当前结点的全部孩子结点
hasChildren	boolean	iAll: boolean	判断当前结点是否拥有子结点
getNextSibling	SequenceActivity	iAll: boolean	获得直接后继兄弟活动结点
getPrevSibling	SequenceActivity	iAll: boolean	获得直接前驱兄弟活动结点
getParent	SequenceActivity	无	获得父活动结点
getParentID	Id	无	获得父活动结点 ID
getActiveOrder	int	无	得到相关的兄弟结点的活动种类
getObjStatusSet	Vector	无	获得对象状态集
setDepth	void	iDepth: int	设置活动深度
getDepth	int	无	获得活动深度
setActivityCount	void	iCount: int	设置活动计数
getActivityCount	int	无	获得活动计数
setSelection	void	iSelection: boolean	设置当前活动的选中状态
isSelected	boolean	无	判断当前活动是否被选中

A. 2. 5. 3 SequenceActivityTree 接口

SequenceActivityTree 接口是编列活动基础上的树型结构模型，管理整个编列活动树。

cn.edu.celtsc.intf.education.sequencing.SequenceActivityTree 接口定义			
import: cn.edu.celtsc.intf.common.id.Id			
java.util.Vector			
java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
setRoot	void	iRoot: SequenceActivity	将一个活动设置为活动树的根结点
getRoot	SequenceActivity	无	获得活动树的根结点活动
setLastLeaf	void	iLastLeaf: String	设置最后一个叶结点
getLastLeaf	String	无	获得最后一个叶结点
setCurrentActivity	void	iCurrent: SequenceActivity	设置当前活动结点
getCurrentActivity	SequenceActivity	无	获得当前活动结点
setFirstCandidate	void	iFirst: SequenceActivity	设置第一个候选活动
getFirstCandidate	SequenceActivity	无	获得第一个候选活动

setSuspendAll	void	iSuspendTarget SequenceActivity	: 将活动设置为 SuspendAll 状态
getSuspendAll	SequenceActivity	无	获得挂起状态 SuspendAll 的活动结点
getActivity	SequenceActivity	iActivityID: Id	获得指定 ID 的结点活动
getObjMap	Vector	obj: String	获得指定学习目标的活动集
getGlobalObjectives	Vector	无	获得学习目标集
clearSessionState	void	无	清除目前活动树的状态
setDepths	void	无	计算当前树的深度
setTreeCount	void	无	设置当前活动树的活动计数

A. 2. 5. 4 Sequencer 接口

Sequencer 接口是编列引擎的基本操作，是联系用户与具体活动树的部件。

cn.edu.celtsc.intf.education.sequencing.Sequencer 接口定义			
import: cn.edu.celtsc.intf.common.base.IdIterator cn.edu.celtsc.intf.common.id.Id java.io.Serializable			
type: public			
extends: Serializable			
成员	返回值/类型	参数说明	说明
getObjStatusSet	IdIterator	activityId: Id	获得指定 ID 活动的学习目标状态集
setActivityTree	void	iTree SequenceActivityTree	: 设置编列活动树
getActivityTree	SequenceActivity Tree	无	获得编列活动树
clearSequenceState	void	无	清除编列引擎状态参数
reportSuspension	void	activityId: String iSuspended: boolean	告知编列引擎当前活动成为挂起状态, 当此活动结束后, 此状态生效
setAttemptDuration	void	activityId: Id iDur: String	告知编列引擎此活动的尝试持续时间改变
setAttemptObjSatisfied	void	activityId: Id objectiveId: Id iStatus: String	告知编列引擎一个活动目标的满足状态改变
setAttemptProgressStatus	void	activityId: Id iProgress: String	告知编列引擎编列活动的过程状态改变

A. 2. 5. 5 SequenceNavigation 接口

SequenceNavigation 是编列导航接口，对应于接收到 IMS SS 中的导航请求时所做的导航行为。

cn.edu.celtsc.intf.education.sequencing.SequenceNavigation 接口定义
--

type: public			
成员	返回值/类型	参数说明	说明
navigate	void	iRequest: int	处理导航请求
navigate	void	iTarget: String	处理"Choice"导航请求

A. 2. 5. 6 SequenceParser 接口

SequenceParser 是编列解析器接口，其类似 XML 的解析器，直接用于解析包含编列信息的 XML 文件，并提供获得相应结点的方法。现有的编列引擎应该支持 Celtsc 和 IMS Simple Sequence 两种规范。在此接口中使用了 DOM 中的 Node 类，在未来的版本中期望这一操作被隔离出去。

cn.edu.celtsc.intf.education.sequencing.SequenceParser 接口定义			
import: org.w3c.dom.Node			
cn.edu.celtsc.intf.common.id.Id			
type: public			
成员	返回值/类型	参数说明	说明
initParser	void	seqFileName: String	初始化编列信息解析器
parseFile	boolean	无	解析文件，解析出错则返回 false，正确解析则为 true
getCourseID	Id	无	获得 Course 的 ID
getOrganizationID	Id	无	获得 Organization 的 ID
getScopeID	Id	无	获得 Scope 的 ID
findDefaultOrganization	Node	无	找到 DefaultOrganization 的结点
getSequencingCollection	Node	无	获得编列集的结点
getAttribute	String	iNode: Node attributeName: String	得到 XML 结点的属性值

A. 2. 6 学习对象管理 (learning object management)

A. 2. 6. 1 LangString 接口

LangString 是多语言字符串接口。多语言字符串容器用于满足标准中多语言字符串可以一次出现多个的情况。如标准所示：一个多语言字符串的值可以包含多个语义相同的字符串，如不同的翻译或选择性的描述。

cn.edu.celtsc.intf.education.lom.common.LangString 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getLang	String	void	获得字符串语种
getContent	String	void	获得字符串内容
setLang	void	Lang: String	设置字符串语种
setContent	void	Content: String	设置字符串内容

A. 2. 6. 2 LangStringContainer 接口

LangStringContainer 是多语言字符串容器接口。多语言字符串容器应用在很多元素中，因此避免了大量的重复接口定义。很多元数据元素即可简单的以该容器子接口的形式出现即可。在应用中，只需要指定多语言容器的名称，即“NAME_”属性，就可以确定具体是哪个元素。

cn.edu.celtsc.intf.education.lom.common.LangStringContainer 接口定义

import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
常量	类型	说明	
NAME_TITLE	String	用于标题元素	
NAME_KEYWORD	String	用于关键字元素	
NAME_DESCRIPTION	String	用于描述元素	
NAME_COVERAGE	String	用于覆盖区元素	
NAME_TYPICAL_AGE_RANGE	String	用于典型年龄范围元素	
NAME_VERSION	String	用于版本元素	
NAME_INSTALLATIONREMARKS	String	用于安装描述元素	
NAME_OTHERPLATFORMREQUIREMENTS	String	用于其他平台要求元素	
NAME_TAXONENTRY	String	用于分类单元条目元素	
NAME_TAXONSOURCE	String	用于分类来源元素	
成员	返回值/类型	参数说明	说明
getName	String	void	获取元素类型名称
getLangStrings	LangString[]	void	获得所有字符串
setLangStrings	void	langStrs: LangString[]	设置所有字符串
getByLang	LangString	lang: String	获得规定语种的字符串
emptyLangString	void	void	清空所有字符串
getContentInString	String	void	获得所有字符串内容
contains	boolean	content: String	是否包含指定字符串

规范中表 5-1 包含了所有的继承了该接口的字符串容器，它们都是简单的继承该接口，没有增加其他成员、方法。

A. 2. 6. 3 SourceValue 接口

SourceValue 是词汇表接口。词汇表学习对象元数据标准中的基本数据类型，也是标准推荐使用的取值列表，以“词汇表”作为数据类型的数据元素的值是一个二元组（来源、值）。

cn.edu.celtsc.intf.education.lom.common.SourceValue 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
常量	类型	说明	
CELT_SOURCE	String[]	CELT 标准类型的词汇表	
成员	返回值/类型	参数说明	说明
getName	String	void	获取元素类型名称
getLangStrings	LangString[]	void	获得所有字符串
setLangStrings	void	langStrs: LangString[]	设置所有字符串
getByLang	LangString	lang: String	获得规定语种的字符串
emptyLangString	void	void	清空所有字符串
getContentInString	String	void	获得所有字符串内容
contains	boolean	content: String	是否包含指定字符串

由于在学习对象元数据标准中有很多的词汇表类型元素，因此与多语言字符串容器类似，词汇表作为学习对象元数据标准的一个基本数据结构，被很多元素所继承，用来代表具体的词汇表型数据元素。

通过调用 `getName` 方法来判断当前词汇表的具体类型。规范中表 5-2 展示了所有的词汇表型数据元素。

A. 2. 6. 4 DateTime 接口

`DateTime` 是日期时间的接口。日期时间是学习对象元数据标准的基本数据类型，用于标准的表示日期时间，是一个时间点。

cn.edu.celtsc.intf.education.lom.common.DateTime 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
<code>getDateTime</code>	String	void	获取日期时间
<code>setDateTime</code>	void	dateTime: String	设置日期时间
<code>setDescription</code>	void	desc: Description	设置描述
<code>getDescription</code>	Description	void	获得描述

A. 2. 6. 5 Duration 接口

`Duration` 是持续时间的接口。持续时间用于描述标准中设计到学习时长。

cn.edu.celtsc.intf.education.lom.common.Duration 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
<code>getDuration</code>	String	void	获取时长
<code>setDuration</code>	void	duration: String	设置时长
<code>setDescription</code>	void	desc: Description	设置描述
<code>getDescription</code>	Description	void	获得描述

规范中表 5-3 是该接口的子接口及其功能说明，其子接口只是简单的继承该接口，没有增加其他成员、方法。

A. 2. 6. 6 VCard 接口

`VCard` 数据类型被广泛的用于描述实体信息，在元数据标准中同样使用 `VCard` 作为实体信息描述的数据结构。

cn.edu.celtsc.intf.education.lom.common.VCard 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
<code>getName</code>	String	void	获取 FN
<code>setName</code>	void	name: String	设置 FN
<code>setOrganization</code>	void	org: String	设置单位信息
<code>getOrganization</code>	String	void	获得单位信息
<code>setTel</code>	void	tel: String	设置电话
<code>getTel</code>	String	void	获得电话
<code>setEmail</code>	void	email: String	设置电子邮箱
<code>getEmail</code>	String	void	获得电子邮箱
<code>setAddress</code>	void	addr: String	设置地址

getAddress	String	void	获得地址
setContent	void	content: String	输入整个 VCard 数据并解析其中内容
getContent	String	void	返回标准的 VCard 串

A. 2. 6. 7 Entity 接口

Entity 是实体的接口。元数据中通过实体作为 VCard 的载体来描述实体信息。

cn.edu.celtsc.intf.education.lom.common.Entity 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getVCard	String	void	获取 VCard 的字符串
setVCard	void	vcard: String	通过字符串方式设置 VCard
getVCardAsObject	void	vcard: VCard	设置 VCard 对象
setVCardAsObject	VCard	void	获得 VCard 对象

A. 2. 6. 8 Identifier 接口

Identifiers 是标识符接口。标识符用来对学习对象、元数据实例等内容进行标识。

cn.edu.celtsc.intf.education.lom.common.Identifier 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getCatalog	String	void	获得类别
setCatalog	void	catalog: String	设置类别
getEntry	String	void	获得表项
setEntry	void	entry: String	设置表项

A. 2. 6. 9 Contribute 接口

Contribute 是贡献的接口。贡献元素用于记录对学习对象、学习对象元数据做出贡献的相关记录。

cn.edu.celtsc.intf.education.lom.common.Contribute 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getRole	Role	void	获得角色
setRole	void	role: Role	设置角色
getEntities	Entity[]	void	获得所有实体
setEntities	void	ents: Entity[]	设置所有实体
addEntity	void	entity: Entity	添加一个实体
emptyEntity	void	void	清空所有实体

A. 2. 6. 10 OrComposite 接口

OrComposite 是或组合接口。或组合用于表示元数据技术类别中多个要求的技术组合方案。

cn.edu.celtsc.intf.education.lom.common.OrComposite 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			

成员	返回值/类型	参数说明	说明
getType	Type	void	获得技术类型
setType	void	type: Type	设置技术类型
getName	Name	void	获得技术名称
setName	void	name: Name	设置技术名称
setMinVersion	void	min: String	获得技术的最低版本
getMinVersion	String	void	设置技术的最低版本
setMaxVersion	void	max: String	获得技术的最高版本
getMaxVersion	String	void	设置技术的最高版本

A. 2. 6. 11 Requirement 接口

Requirement 是要求接口。表示使用学习对象所需要的技术上的要求。

cn.edu.celtsc.intf.education.lom.common.Requirement 接口定义

```
import: cn.edu.celtsc.intf.education.Serialize
```

```
extends: Serialize
```

```
type: public
```

成员	返回值/类型	参数说明	说明
getOrComposites	OrComposite[]	void	获得所有或组合
setOrComposites	void	ors: OrComposite[]	设置所有或组合
addOrComposite	void	or: OrComposite	添加一个或组合
emptyOrComposite	void	void	清空所有或组合

A. 2. 6. 12 Resource 接口

Resource 是资源接口。资源元素表示该学习对象对其他学习对象的引用记录。

cn.edu.celtsc.intf.education.lom.common.Resource 接口定义

```
import: cn.edu.celtsc.intf.education.Serialize
```

```
extends: Serialize
```

```
type: public
```

成员	返回值/类型	参数说明	说明
getIdentifiers	Identifier[]	void	获得所有资源标识符
setIdentifiers	void	ids: Identifier[]	设置所有资源标识符
addIdentifier	void	id: Identifier	添加一个资源标识符
getDescriptions	Description[]	void	获得所有描述信息
setDescriptions	void	descs: Description[]	设置所有描述信息
addDescription	void	desc: Description	添加一个描述信息
emptyDescriptions	void	void	清空所有描述信息

A. 2. 6. 13 Taxon 接口

Taxon 是分类单元接口。分类单元是具有确定标号或术语的结点。这是从通用条目到具体条目的一条细化路径。一般用<标识, 条目>二元组来表示。

cn.edu.celtsc.intf.education.lom.common.Taxon 接口定义

```
import: cn.edu.celtsc.intf.education.Serialize
```

```
extends: Serialize
```

type: public			
成员	返回值/类型	参数说明	说明
getId	String	void	获得分类单元分类标识
setId	void	id: String	设置分类单元分类标识
getEntry	TaxonEntry	void	获得分类条目
setEntry	void	entry: TaxonEntry	设置分类条目

A. 2. 6. 14 TaxonPath 接口

TaxonPath 是分类路径接口。表示学习对象在某一特定分类系统中的分类路径。

cn.edu.celtsc.intf.education.lom.common.TaxonPath 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getSource	TaxonSource	void	获得分类系统的名称
setSource	void	source: TaxonSource	设置分类系统的名称
getTaxons	Taxon[]	void	获得所有分类单元信息
setTaxons	void	taxons: Taxon[]	设置所有分类单元信息
addTaxon	void	taxon: Taxon	添加一个分类单元信息
emptyTaxons	void	void	清空所有分类单元信息

A. 2. 6. 15 General 接口

General 是通用类接口。该顶级元素对应的接口描述了学习对象的通用信息。

cn.edu.celtsc.intf.education.lom.General 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getIdentifiers	Identifier[]	void	获得所有标识符
setIdentifiers	void	ids: Identifier[]	设置所有标识符
addIdentifier	void	id: Identifier	添加一个标识符
emptyIdentifiers	void	void	清空所有标识符
getTitle	String	void	获得标题
setTitle	void	title: String	设置标题
getLanguages	String[]	void	获得所有语种
setLanguages	void	langs: String[]	设置所有语种
addLanguage	void	taxon: Taxon	添加一个语种
emptyLanguages	void	void	清空所有语种
getDescriptions	Description[]	void	获得所有描述信息
setDescriptions	void	descs: Description[]	设置所有描述信息
addDescription	void	desc: Description	添加一个描述信息
emptyDescriptions	void	void	清空所有描述信息
getKeywords	Keyword[]	void	获得所有关键字

setKeywords	void	keys: Keyword[]	设置所有关键字
addKeyword	void	key: Keyword	添加一个关键字
emptyKeywords	void	void	清空所有关键字
getCoverages	Coverage[]	void	获得所有覆盖区
setCoverages	void	covs: Coverage[]	设置所有覆盖区
addCoverage	void	cov: Coverage	添加一个覆盖区
emptyCoverages	void	void	清空所有覆盖区
getStructure	Structure	void	获得结构信息
setStructure	void	struct: Structure	设置结构信息
getAggregationLevel	AggregationLevel	void	获得聚合度信息
setAggregationLevel	void	aggr: AggregationLevel	设置聚合度信息

A. 2. 6. 16 LifeCycle 接口

LifeCycle 是生命期接口。生命期类描述了学习对象的历史和当前状态以及那些对学习对象的发展过程发生作用的实体。

cn.edu.celtsc.intf.education.lom.LifeCycle 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getVersion	Version	void	获得版本信息
setVersion	void	version: Version	设置版本信息
getStatus	Status	void	获得状态信息
setStatus	void	status: Status	设置状态信息
getContributes	Contribute[]	void	获得所有贡献信息
setContributes	void	cons: Contribute[]	设置所有贡献信息
addContribute	void	con: Contribute	添加一个贡献信息
emptyContributes	void	void	清空所有贡献信息

A. 2. 6. 17 Metametadata 接口

Metametadata 是元-元数据类接口。该类别描述了元数据实例自身的信息。

cn.edu.celtsc.intf.education.lom.Metametadata 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getIdentifiers	Identifier[]	void	获得所有标识符
setIdentifiers	void	ids: Identifier[]	设置所有标识符
addIdentifier	void	id: Identifier	添加一个标识符
emptyIdentifiers	void	void	清空所有标识符
getContributes	Contribute[]	void	获得所有贡献信息
setContributes	void	cons: Contribute[]	设置所有贡献信息

addContribute	void	con: Contribute	添加一个贡献信息
emptyContributes	void	void	清空所有贡献信息
getMetadataSchemas	String[]	void	获得所有元数据方案
setMetadataSchemas	void	schemas: String[]	设置所有元数据方案
addMetadataSchema	void	void	添加一个元数据方案
emptyMetadataSchemas	void	void	清空所有元数据方案
getLanguage	String	void	获得元数据语言
setLanguage	void	lang: String	设置元数据语言

A. 2. 6. 18 Technical 接口

Technical 是技术类接口。技术类别描述了学习对象的技术要求及其相关特征。

cn.edu.celtsc.intf.education.lom.Technical 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getFormats	String[]	void	获得所有资源格式
setFormats	void	formats: String[]	设置所有资源格式
addFormat	void	format: String	添加一个资源格式
emptyFormats	void	void	清空所有资源格式
getLocations	String[]	void	获得所有资源路径
setLocations	void	locs: String[]	设置所有资源路径
addLocation	void	loc: String	添加一个资源路径
emptyLocations	void	void	清空所有资源路径
getRequirements	Requirement[]	void	获得所有技术要求
setRequirements	void	reqs: Requirement[]	设置所有技术要求
addRequirement	void	req: Requirement	添加一个技术要求
emptyRequirements	void	void	清空所有技术要求
getSize	long	void	获得资源大小
setSize	void	size: long	设置资源大小
getInstallationRemarks	InstallationRemarks []	void	获得所有学习对象的安装描述信息
setInstallationRemarks	void	install: InstallationRemarks []	设置所有学习对象的安装描述信息
addInstallationRemark	void	install: InstallationRemarks	添加一个学习对象的安装描述信息
emptyInstallationRemarks	void	void	清空所有学习对象的安装描述信息

marks			
getOtherPlatformRequirements	OtherPlatformRequirements	void	获得其他平台需求
setOtherPlatformRequirements	void	other: OtherPlatformRequirements	设置其他平台需求
getDuration	Duration	void	获得持续时间
setDuration	void	duration: Duration	设置持续时间

A. 2. 6. 19 Educational 接口

Educational 是教育类接口。该类别描述了学习对象在教育 and 教学方面的一些关键特征。

cn.edu.celtsc.intf.education.lom.Educational 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getLearningResourceTypes	LearningResourceType[]	void	获得所有学习资源类型
setLearningResourceTypes	void	types: LearningResourceTypes []	设置所有学习资源类型
addLearningResourceType	void	type: LearningResourceType	添加一个学习资源类型
emptyLearningResourceTypes	void	void	清空所有学习资源类型
getIntendedEndUserRoleRoles	IntendedEndUserRole[]	void	获得所有终端用户类型
setIntendedEndUserRoleRoles	void	roles: IntendedEndUserRole []	设置所有终端用户类型
addIntendedEndUserRole	void	role: IntendedEndUserRole	添加一个终端用户类型
emptyIntendedEndUserRoleRoles	void	void	清空所有终端用户类型
getContexts	Context[]	void	获得所有语境
setContexts	void	conts: Context[]	设置所有语境
addContext	void	cont: Context	添加一个语境
emptyContexts	void	void	清空所有语境
getTypicalAgeRanges	TypicalAgeRange[]	void	获得所有语境
setTypicalAgeRanges	void	conts :	设置所有语境

es		TypicalAgeRange []	
addTypicalAgeRange	void	cont : TypicalAgeRange	添加一个语境
emptyTypicalAges	void	void	清空所有语境
getDescriptions	Description[]	void	获得所有描述信息
setDescriptions	void	descs: Description[]	设置所有描述信息
addDescription	void	desc: Description	添加一个描述信息
emptyDescriptions	void	void	清空所有描述信息
getLanguages	String[]	void	获得所有语种
setLanguages	void	langs: String[]	设置所有语种
addLanguage	void	taxon: Taxon	添加一个语种
emptyLanguages	void	void	清空所有语种
getInteractivityType	InteractivityType	void	获得交互类型
setInteractivityType	void	intType: InteractivityType	设置交互类型
getInteractivityLevel	InteractivityLevel	void	获得交互等级
setInteractivityLevel	void	intLevel: InteractivityLevel	设置交互等级
getSemanticDensity	SemanticDensity	void	获得语义密度
setSemanticDensity	void	semanticDens: SemanticDensity	设置语义密度
getDifficulty	Difficulty	void	获得难度
setDifficulty	void	difficulty: Difficulty	设置难度
getTypicalLearningTime	TypicalLearningTime	void	获得典型学习时间
setTypicalLearningTime	void	learningTime: TypicalLearningTime	设置典型学习时间

A. 2. 6. 20 Rights 接口

Rights 是权利类接口。该类别描述了学习对象的知识产权和使用条件等信息。

cn.edu.celtsc.intf.education.lom.Rights 接口定义

```
import: cn.edu.celtsc.intf.education.Serialize
extends: Serialize
type: public
```

成员	返回值/类型	参数说明	说明
getCost	Cost	void	获得是否收费信息
setCost	void	cost: Cost	设置是否收费信息

getCopyrightAndOtherRestrictions	CopyrightAndOtherRestriction[]	void	获得所有版权和其他限制信息
setCopyrightAndOtherRestrictions	void	copyrights: CopyrightAndOtherRestriction[]	设置所有版权和其他限制信息
addCopyrightAndOtherRestriction	void	copyright: CopyrightAndOtherRestriction	添加一条版权或使用限制信息
emptyCopyrightAndOtherRestrictions	void	void	清空所有版权限制信息

A. 2. 6. 21 Relation 接口

Relation 是关系类接口。该类别定义了学习对象与其他学习对象（目标学习对象）之间的关系。

cn.edu.celtsc.intf.education.lom.Relation 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getKind	Kind	void	获得学习对象与指定目标的关系种类
setKind	void	kind: Kind	设置学习对象与指定目标的关系种类
getResource	Resource	void	获得与本学习对象关联的资源信息
setResource	void	resource: Resource	设置与本学习对象关联的资源信息

A. 2. 6. 22 Annotation 接口

Annotation 是评注类接口。该类别提供了学习对象在教学使用方面的一些评价，以及这些评论的作者和创作时间。

cn.edu.celtsc.intf.education.lom.Annotation 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getEntity	Entity	void	获得学习对象评注信息对应的实体
setEntity	void	ent: Entity	设置学习对象评注信息对应的实体
getDate	DateTime	void	获得学习对象评注信息的时间
setDate	void	date: DateTime	设置学习对象评注信息的时间
getDescription	Description	void	获得评注信息
setDescription	void	desc: Description	设置评注信息

A. 2. 6. 23 Classification 接口

Classification 是分类类接口。

cn.edu.celtsc.intf.education.lom.Classification 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明

getKeywords	Keyword[]	void	获得所有关键字
setKeywords	void	keys: Keyword[]	设置所有关键字
addKeyword	void	key: Keyword	添加一个关键字
emptyKeywords	void	void	清空所有关键字
getTaxonPaths	TaxonPath[]	void	获得所有分类路径
setTaxonPaths	void	paths: TaxonPath[]	设置所有分类路径
addTaxonPath	void	path: TaxonPath	添加一个分类路径
emptyTaxonPaths	void	void	清空所有分类路径
getDescription	Description	void	获得分类描述信息
setDescription	void	desc: Description	设置分类描述信息
getPurpose	Purpose	void	获得分类目的
setPurpose	void	purpose: Purpose	设置分类目的

A. 2. 6. 24 LORoot 接口

LORoot 是学习对象根结点接口。该对象用于只是一个完整的学习对象的跟结点。该结点同时提供创建所有 9 个顶层元素的工厂方法。

cn.edu.celtsc.intf.education.lom.LORoot 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getGeneral	General	void	获得通用类
setGeneral	void	general: General	设置通用类
getLifeCycle	LifeCycle	void	获得生命期类
setLifeCycle	void	life: LifeCycle	设置生命期类
getMetametadata	Metametadata	void	获得元-元数据类
setMetametadata	void	metametadata: Metametadata	设置元-元数据类
getTechnical	Technical	void	获得技术类
setTechnical	void	tech: Technical	设置技术类
getEducationals	Educational[]	void	获得所有教育类
setEducacionals	void	keys: Educational[]	设置所有教育类
addEducational	void	key: Educational	添加一个教育类
emptyEducacionals	void	void	清空所有教育类
getRelations	Relation[]	void	获得所有关系类
setRelations	void	relations: Relation[]	设置所有关系类
addRelation	void	relation: Relation	添加一个关系类
emptyRelations	void	void	清空所有关系类
getRights	Rights	void	获得权利类
setRights	void	rights: Rights	设置权利类
getAnnotations	Annotation[]	void	获得所有评注类

setAnnotations	void	annos: Annotation[]	设置所有评注类
addAnnotation	void	anno: Annotation	添加一个评注类
emptyAnnotations	void	void	清空所有评注类
getClassifications	Classification[]	void	获得所有分类类
setClassifications	void	classes: Classification[]	设置所有分类类
addClassification	void	cla: Classification	添加一个分类类
emptyClassifications	void	void	清空所有分类类
createGeneral	General	void	生成一个通用类实例
createLifeCycle	LifeCycle	void	生成一个生命期类实例
createMetametadata	Metametadata	void	生成一个元-元数据类实例
createTechnical	Technical	void	生成一个技术类实例
createEducational	Educational	void	生成一个教育类实例
createRelation	Relation	void	生成一个关系类实例
createRights	Rights	void	生成一个权利类实例
createAnnotation	Annotation	void	生成一个评注类实例
createClassification	Classification	void	生成一个分类类实例

A. 2. 6. 25 LOFactory 接口

LOFactory 是学习对象工厂接口。该接口使用了设计模式中常用的工厂模式，该工厂用于创建一个完整的学习对象实例。

cn.edu.celtsc.intf.education.lom.LOFactory 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
createRoot	LORoot	void	创建学习对象元数据实例

A. 2. 6. 26 LOAction 接口

LOAction 是学习对象动作接口。该接口用于进行学习对象元数据相关的操作的父类接口。

cn.edu.celtsc.intf.education.lom.LOAction 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
execute	void	void	执行该动作对应的功能

在规范的正文中，表 5-4 中是该接口的子接口，它们规定了对于学习对象的各种操作，通过实现 execute 方法可以将操作具体化。

A. 2. 7 内容包装 (content packaging)

A. 2. 7. 1 HasMetadata 接口

HasMetadata 接口代表一段元数据，在学习内容包中的很多数据元素都包含元数据，因此有必要抽象出来单独的作为接口。

cn.edu.celtsc.intf.education.cp.HasMetadata 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getId	String	void	获得元数据标识符
setId	void	id: String	设置元数据标识符
getMetadata	String	void	获得元数据内容
setMetadata	void	metadata: String	设置元数据内容

A. 2. 7. 2 Level 接口

Level 是等级接口，代表学习内容包中结点的共性操作属性接口。

cn.edu.celtsc.intf.education.cp.Level 接口定义			
extends: HasMetadata			
type: public			
成员	返回值/类型	参数说明	说明
getTitle	String	void	获得元素标题
setTitle	void	title: String	设置元素标题
getItems	Item[]	void	获得当前结点所有子结点元素
setItems	void	items: Item[]	设置当前结点所有子结点元素
addItem	void	item: Item	顺序地添加一个子结点
emptyItems	void	void	清空所有子结点
addItem	void	item: Item, location: int	按照指定位置插入子结点
deleteItem	void	location: int	删除指定位置的子结点
deleteItem	void	content: String	删除包含指定内容的子结点
getItemById	Item	id: String	根据 ID 获得子结点
getChildren	Iterator	void	返回子结点迭代器
getSize	int	void	返回子结点数量
find	int	content: String	根据内容查找子结点位置

A. 2. 7. 3 Item 接口

Item 是结点接口，表示内容结构中的一个结点。结点中包括了对资源信息的引用，结点终包含的标题信息等属性。

cn.edu.celtsc.intf.education.cp.Item 接口定义			
extends: Level			
type: public			
成员	返回值/类型	参数说明	说明
isVisible	boolean	void	元素是否可见？
setVisible	void	visible: boolean	设置元素是否可见
getParameters	String	void	获得当前结点所有参数
setParameters	void	params: String	设置当前结点所有参数
getIDRef	Resource	void	获得结点关联资源
setIDRef	void	idref: Resource	设置结点关联资源
isLeaf	boolean	void	当前结点是否叶结点

A. 2. 7. 4 Organization 接口

Organization 是组织接口，表示对学习内容包清单中所有实际物理资源和所需媒体素材的引用。

cn.edu.celtsc.intf.education.cp.Organization 接口定义			
extends: Level			
type: public			
成员	返回值/类型	参数说明	说明
isDefault	boolean	void	是否为默认?
setDefault	void	default: boolean	设置为默认
getStructure	String	void	获得结构信息
setStructure	void	structure: String	设置结构信息

A. 2. 7. 5 Organizations 接口

Organizations 是组织集接口，表示学习内容包清单中一种或多种学习内容的组织方法。

cn.edu.celtsc.intf.education.cp.Organizations 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getDefaultID	String	void	获得默认结点标识符
setDefaultID	void	default: String	设置默认结点标识符
getOrganizations	Organization[]	void	获得所有组织结点
getOrganization	Organization	id: String	根据标识符获取组织结点
getOrganization	Organization	location: int	根据位置获取组织结点
addOrganization	void	org: Organization, location: int	将组织结点插入到指定位置
setOrganizations	void	orgs: Organization[]	设置所有组织结点
emptyOrganizations	void	void	清空所有组织结点
remove	void	location: int	删除指定位置的组织结点
find	int	id: String	根据标识符查找组织结点位置
iterator	Iterator	void	获得组织结点迭代器
size	int	void	获得组织结点个数
getAllItemID	String[]	void	获得所有 Item 结点标识符
getItemByID	Item	id: String	根据标识符获得 Item 结点

A. 2. 7. 6 ResourceFile 接口

ResourceFile 是资源文件接口，包含了对本地资源文件的引用信息。

cn.edu.celtsc.intf.education.cp.ResourceFile 接口定义			
extends: HasMetadata			
type: public			
成员	返回值/类型	参数说明	说明
getHref	String	void	获得资源文件路径
setHref	void	href: String	设置资源文件路径

A. 2. 7. 7 Dependency 接口

Dependency 是依赖接口，表示对引用资源所依赖的文件。

cn.edu.celtsc.intf.education.cp.Dependency 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getIDRef	String	void	获得对本地资源文件的引用
setIDRef	void	idref: String	设置对本地资源文件的引用

A. 2. 7. 8 Resource 接口

Resource 是资源接口，表示对学习内容有实际物理资源和所需媒体素材的引用。

cn.edu.celtsc.intf.education.cp.Resource 接口定义			
extends: HasMetadata			
type: public			
成员	返回值/类型	参数说明	说明
getType	String	void	获得资源 MIME 类型
setType	void	type: String	设置资源 MIME 类型
getXMLBase	String	void	获得 xml:base 属性
setXMLBase	void	base: String	设置 xml:base 属性
getHref	String	void	获得资源文件路径
setHref	void	href: String	设置资源文件路径
getResourceFiles	ResourceFile[]	void	获得所有资源文件
setResourceFiles	void	resources: ResourceFile[]	设置所有资源文件
addResourceFile	void	resources: ResourceFile	添加一个资源文件
emptyResourceFiles	void	void	清空所有资源文件
getDependencies	Dependency[]	void	获得所有引用文件列表
setDependencies	void	deps: Dependency[]	设置所有引用文件列表
addDependency	void	dep: Dependency	添加一个引用文件
emptyDependencies	void	void	清空所有引用文件

A. 2. 7. 9 Resources 接口

Resources 是资源列表接口，表示一个资源引用的集合。

cn.edu.celtsc.intf.education.cp.Resources 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getXMLBase	String	void	获得 xml:base 属性
setXMLBase	void	base: String	设置 xml:base 属性
getResources	Resource[]	void	获得所有资源结点
getResource	Resource	id: String	根据标识符获取资源结点

getResource	Resource	location: int	根据位置获取资源结点
addResource	void	org: Resource, location: int	将资源结点插入到指定位置
setResources	void	orgs: Resource[]	设置所有资源结点
emptyResources	void	void	清空所有资源结点
remove	void	id: String	删除指定标识符的资源结点
remove	void	location: int	删除指定位置的资源结点
find	int	id: String	根据标识符查找资源结点位置
iterator	Iterator	void	获得资源结点迭代器
size	int	void	获得资源结点个数
getAllResourceID	String[]	void	获得所有资源标识符

A. 2. 7. 10 ManifestMetadata 接口

ManifestMetadata 是内容清单数据接口，表示内容清单使用的元数据信息。

cn.edu.celtsc.intf.education.cp.ManifestMetadata 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getSchema	String	void	获得该学习内容包使用的规范
setSchema	void	schema: String	设置该学习内容包使用的规范
getSchemaVersion	String	void	获得该学习内容包使用规范的版本
setSchemaVersion	void	schemaVersion: String	设置该学习内容包使用规范的版本

A. 2. 7. 11 Manifest 接口

Manifest 是内容清单接口，表示学习内容包的根结点。

cn.edu.celtsc.intf.education.cp.Manifest 接口定义			
import: cn.edu.celtsc.intf.education.Serialize			
extends: Serialize			
type: public			
成员	返回值/类型	参数说明	说明
getManifests	Manifest[]	void	获得所有子内容清单
setManifests	void	subManifests: Manifest[]	设置所有子内容清单
addManifest	void	subManifest: Manifest	添加一个子内容清单
emptyManifests	void	void	清空所有子内容清单
getMetadata	ManifestMetada ta	void	获得内容清单元数据
setMetadata	void	metadata: ManifestMetadata	设置内容清单元数据
getVersion	String	void	获得版本属性

setVersion	void	version: String	设置版本属性
getXMLBase	String	void	获得 xml:base 属性
setXMLBase	void	base: String	设置 xml:base 属性
getId	String	void	获得元数据标识符
setId	void	id: String	设置元数据标识符
getOrganizations	Organizations	void	获得组织列表
setOrganizations	void	orgs: Organizations	设置组织列表
getDefaultOrg	String	void	获得默认组织
setDefaultOrg	void	id: String	设置默认组织
getResources	Resources	void	获得资源列表
setResources	void	resources: Resources	设置资源列表
getOrganization	Organization	id: String	根据标识符获取组织
getResource	Resource	id: String	根据标识符获取资源
addResource	void	resource: Resource	添加一个资源
deleteResource	void	id: String	根据标识符删除资源

A. 2. 7. 12 CPAction 接口

CPAction 是内容包装动作接口。该接口用于进行内容包装相关的操作的父类接口。

cn.edu.celtsc.intf.education.cp.CPAction 接口定义

```
import: cn.edu.celtsc.intf.education.Serialize
```

```
extends: Serialize
```

```
type: public
```

成员	返回值/类型	参数说明	说明
execute	void	void	执行该动作对应的功能

参考文献

- [1] IMS Global Learning Consortium, Inc. IMS Abstract Framework: White Paper, version 1.0, Jul 2003
 - [2] The Open Knowledge Initiative. Open Service Interface Definition, OSID Release: 1.0 rc6.1, 2003
-