



# 中华人民共和国国家标准

GB/T×××××-××××

## 信息技术 学习、教育和培训 学习者模型

Information technology - Learning, education and training -  
Learner model

(报批稿)

200X-××-××发布

200X-××-××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 .....	III
引言 .....	IV
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语和定义 .....	1
4 一致性 .....	5
4.1 基本原理 .....	5
4.2 一致性级别 .....	6
4.2.1 子集 .....	6
4.2.2 严格一致的实现 .....	6
4.2.3 一致的实现 .....	6
4.2.4 非一致的实现 .....	6
4.2.5 数据元素的约束 .....	7
4.2.6 数据元素的生命期 .....	8
4.2.7 约束/生命期的递归/前后关系 .....	9
4.3 一致性标签 .....	9
4.4 编码一致性 .....	10
4.4.1 一致性的定义 .....	10
4.4.2 数据集一致性 .....	10
4.4.3 数据实例一致性 .....	11
4.5 API 一致性 .....	11
4.6 协议一致性 .....	12
4.7 数据应用一致性 .....	12
4.7.1 严格一致的数据应用 .....	12
4.7.2 一致的数据应用 .....	12
4.7.3 数据应用的多样性 .....	13
4.7.4 数据信息库 .....	13
4.7.5 数据读取器 .....	13
4.7.6 数据书写器 .....	14
5 学习者概念模型介绍 .....	14
5.1 学习者信息框架 .....	14
5.2 信息类型与数据信息库 .....	15
5.3 数据访问模型 .....	16
5.4 远程，分布和漫游系统 .....	17
5.5 粒度水平的相互关系 .....	17
5.6 安全模型 .....	17
5.7 学习者数据集 .....	<b>Error! Bookmark not defined.</b>
5.8 粒度水平的相互关系 .....	<b>Error! Bookmark not defined.</b>
6 语义 .....	18
6.1 一般的数据操作 .....	18

6.2 具体应用数据操作.....	19
6.3 数据兼容性.....	20
6.4 数据类型.....	20
6.4.1 基本数据类型的定义.....	20
6.4.2 复杂数据类型的定义.....	22
6.4.3 未定义的数据类型.....	27
7 学习者信息定义.....	27
7.1 个人信息.....	27
7.2 学业信息.....	31
7.3 管理信息.....	33
7.4 关系信息.....	36
7.5 安全信息.....	37
7.6 偏好信息.....	39
7.7 绩效信息.....	40
7.8 作品集信息.....	43

## 前 言

本标准由教育部提出。

本标准由全国信息技术标准化委员会归口。

本标准起草单位：华南理工大学、清华大学。

本标准主要起草人：奚建清、陈天、文志康、张杨、郑莉、李秀、安颖莲。

## 引 言

本标准规定了学习者信息的语义和语法。学习者信息是与学习者相关并由学习系统使用的信息，可以被学习系统、个人（如教师和学生等）以及其它实体创建。在本标准中，学习者信息被划分为八类：个人信息、学业信息、管理信息、关系信息、安全信息、偏好信息、绩效信息和作品集信息。

本标准的目的是：

- 1) 使任何年龄、背景、地域、学习方式、学校或工作环境的学习者，能够创建基于标准的个人学习者信息库。这个信息库可以在他们的整个教育、学习经历中以及工作生涯中使用。
- 2) 提高学习者信息数据的可移植性。
- 3) 使内容管理和学习管理系统能为学习者提供更为个性化和有效的学习体验。
- 4) 为数据的安全、私密和完整性提供一个框架。

# 信息技术 学习、教育和培训

## 学习者模型

### 1 范围

本标准规定了学习者模型的语法和语义、学习者个人信息、学业信息、管理信息、关系信息、安全信息、偏好信息、绩效信息和作品集信息模型。

本标准适用于为不同特征的学习者信息库的建设并且在不同系统之间对学习者的信息迁移和共享，提高学习者信息数据的可移植性，同时保证用户数据安全、隐私和完整。

### 2 规范性引用文件

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

GB 2261-1980 人的性别代码

GB 2312-1980 信息交换用汉字编码字符集

GB/T 2659-2000 世界各国和地区名称代码 (eqv ISO 3166-1:1997)

GB 3304-1991 中国各民族名称的罗马字母拼写法和代码

GB 4762-1984 政治面貌代码

GB 4767-1984 健康状况代码

GB/T 7408—2005 数据元和交换格式 信息交换 日期和时间表示法 (ISO 8601:2000, IDT)

GB/T 8562-1988 纪律处分代码

GB/T 8563-1988 奖励代码

GB/T 15424-1994 电子数据交换用支付方式代码

GB/T 16835-1997 高等学校本科、专科专业名称代码

GB/T 18221—2000 信息技术 程序设计语言、环境与系统软件接口 独立于语言的数据类型 (idt ISO/IEC 11404: 1996)

GB/T 21365-2008 信息技术 学习、教育和培训 学习对象元数据

GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符

### 3 术语和定义

下列术语和定义适用于本标准。

#### 3.1

**安全边界 security perimeter**

隔离“内部”与“外部”的连续、封闭的部分。安全边界的“内部”必需是安全的。而“外部”则不一定是安全的。

#### 3.2

**安全边界完整性 security perimeter integrity**

使内向和向外安全威胁均保持在可接受的范围内的安全水平。

示例1：安全边界完整性的缺失表现在：(1)安全保护水平低于可接受的水平；(2)内向安全威胁增加到不可接受的水平；或者(3)外向安全威胁增加到不可接受的水平。

示例2：用户的权限和管理是安全边界完整性的一方面。

#### 3.3

**安全管理员 security administrator**

安全边界内负责安全管理的人。

**3.4**

**安全强度 security strength**

描述安全边界实现情况的安全度量。

注：下面几点特征可描述安全强度：

——质量标准：提供了什么样的安全标准？安全标准的例子：最低安全，可审计，可验证的设计。

□ ——偶然性：系统受侵犯时会发生什么？哪些保护将生效？哪些回退点是可用的？受侵害的程度如何？

□ ——惩罚：对安全机构的入侵者将发生什么？

示例：安全强度也关心加密密钥使用的位数。

**3.5**

**绑定 binding**

从一个框架或标准到另一个框架或标准的应用或映射。

**3.6**

**编码 coding**

1) 在信息交换中，信息的格式化或结构化表示。

另见：信息编码（3.7）

2) 按某一结构表示信息的过程。

**3.7**

**信息编码 encoding**

信息的位或字节格式及其表示。

注：信息编码也可以简称为“编码”。

**3.8**

**学习者 learner**

依靠学习技术系统获取知识或技能的个体。

**3.9**

**学习者记录 learner record**

本标准定义的学习者信息集合。

**3.10**

**学习者记录引用 learner record reference**

指向学习者记录的标识符。

**3.11**

**学习者信息 learner information**

学习者或学习者实体的学习技术信息和人的信息的交集。

**3.12**

**学习者应用 learner application**

运用与本标准一致的API、编码或协议的信息技术应用。

**3.13**

**代理 agent**

能够代表人的行为和动作的自动化系统。

**3.14**

**访问 access**

“访问”某个特性是指应用程序读写数据的过程。

**3.15**

**访问者 accessor**



试图访问安全边界内资源的用户、系统、代理或实体。

### 3.16

**分布式(访问、系统) distributed(access、system)**

以互联网或广域网作为子系统之间和其他相关系统之间的主要通信方式的技术系统。

### 3.17

**分解操作 decomposition operation**

从容器中把信息类型实例分离。

### 3.18

**互操作性 interoperability**

两个或多个系统之间交换信息或功能的能力。

### 3.19

**解释(数据) interpret (data)**

依据标准处理数据以得到其语义。

### 3.20

**基于角色的访问控制 role-based access control**

基于用户身份和他与其他用户的关系,对是否授权操作或允许其访问资源进行验证的安全技术。

### 3.21

**聚合操作 aggregation operation**

合并一个或更多数据类型的几个实例到单个容器中的操作。

### 3.22

**聚合(数据类型, 数值) aggregate (datatype, value)**

一个生成的数据类型,其数值由其他构件数据类型的数值组成,在这一意义上对所有组件值的操作都是有意义的。

### 3.23

**漫游(访问、系统) nomadic (access、system )**

(1)服务的一种连续性形式,通过这种形式,服务可以跨越不同的地点和时间段。

(2)部分时间与子系统或相关系统的通讯网络断开。

注:也被称为“随时连通”“随时漫游”。

### 3.24

**粒度 granularity**

有关对象的大小、尺度及其详细程度等特征的描述。

### 3.25

**命名空间 namespace**

规定各种标记含义的协议。

### 3.26

**对内安全威胁 inbound security threat**

破坏系统安全并影响安全边界内部信息的外部威胁。

示例:将数据注入通信流中;改变信息;破坏信息。

### 3.27

**对外安全威胁 outbound security threat**

对信息进行偷窃或未经授权复制,而使信息在安全边界外可用或在边界内不可用。

示例:侦听网络包;采集信息。

### 3.28

**人的信息 human information**

一个自然人所有信息的集合。

3.29

**生成 generate**

把数据按其语义转换成适合数据交换的形式。

示例：按照概念模型，对数据结构进行序列化，而不按照特定编码展现数据。

3.30

**实现 implementation**

——抽象的工作实例。

——低层次的抽象。

——生成实现的过程。

3.31

**实现定义行为（值） implementation-defined behavior (value)**

未指定的行为或未指定的值，这些行为或值需要在实现时做出选择。

3.32

**数据对象 data object**

在数据访问的概念模型中的数据处理单位。

3.33

**数据集 data set**

数据结构的实例。

另见：数据结构（3.34）

注：数据集是独立绑定的。

3.34

**数据结构 data structure**

有下列两个定义：

——零个或多个数据元素聚集而成的数据类型。

——零个或多个数据元素聚集而成的实例。

示例：一条记录；一个集合；一个序列；一个列表；一个数组。

注：一个数据结构可以是更高水平的数据结构中的一个元素。

3.35

**数据类型 data type**

由不同的值组成的集合的一个属性，该属性描述这些不同的值所具有的共同特点以及所能施加的操作。

3.36

**数据实例 data instance**

以某种绑定呈现的数据集。

3.37

**数据应用 data application**

数据互操作性标准内功能模块的实现。

示例：数据信息库；数据读取；数据写入。

注1：在这个定义下，一个功能模块也可以叫做“信息技术系统”。

注2：给出一个数据互操作性标准称为D，一个标准的D数据应用和标准的D应用是不同的。后者是包括标准D的任何信息技术的实现，而前者是一个有限定的信息技术系统。

3.38

**体验 experience**

学习者在学习活动中对事物的认识和亲身经历。

3.39

**信息库 repository**

——信息集合。

——信息与数据存取控制机制（如检索、索引、存储、恢复和安全）集的实现。

**3.40****信息类型 information type**

在具体应用系统领域中的信息种类。

**3.41****学习管理系统 learning management system**

一个系统，用于：

——调度学习资源；

——辅助、控制或者指引学习过程；

——分析和报告学习者成绩。

**3.42****学习者标识 learner identifier**

与学习者相关的标志。

注：学习者可以拥有多于一个学习者标识符——多个标识符。单个或多个标识符的方案不在本标准说明。见 GB/T 21366-2008。

**3.43****学习者绩效粒度 learner performance granularity**

学习者绩效信息的相对大小、范围或细节。

注：学习者绩效记录可以有不同粒度。

示例：记录按键次数（例如粒度可以取学习的秒数）；学习者进度的每分钟数据样本（如学习的分钟数）；课的得分（如学习的日数）；学期成绩（如学习的月数）；专业认证（如学习的年数）。

**3.44****学习者实体 learner entity**

与学习系统进行交互的单个的学习者，或者一组视做单个实体的学习者。

另见：学习者（3.42）。

**3.45****用户 user**

人、其代理人或与信息技术系统进行交互的代理。

**3.46****语义 semantics**

对应的解释和含义。

**3.47****远程（访问、系统） distance (access、system)**

通过网络的应用，受通讯系统的带宽或延迟所限制。

**3.48****最低峰值（SPM） smallest permitted maximum**

用于由实现定义的值，指最大值的可允许最小数。

示例：X元素字符串长度的最低峰值是17。

**4 一致性****4.1 基本原理**

一致性，一方面与数据集的结构有关，另一方面与系统的行为有关。也就是说，一致性既有非行为性的范围也有行为性的，两者都很重要。“行为”是一般意义上的，例如，一个数据集不一定有“行为”，

但是在数据集的周边，存储、提取、生成和解释数据集的实现都出现“行为”——并注明“未定义”、“实现定义”和“未指定”。

注：在本标准中，“应”解释为对一个实现的要求；反过来，“不应”表示禁止。不属于要求或被禁止的行为就是未定义的。未定义行为在本标准中以“未定义行为”表示或以行为的具体定义表示。

#### 4.2 一致性级别

以下的小节定义严格一致的实现和一致的实现。在一致性的上下文中，术语“支持”、“使用”、“测试”、“访问”、和“探测”将在编码一致性（4.3）、API一致性（4.4）、协议一致性（4.5）和数据应用一致性（4.6）中进行定义。

##### 基本原理

区分“严格一致”和“一致”的实现，对于定位互操作性和扩展的需求是十分必要。本标准给出提高互操作性的说明。扩展是应用户、供应商、院校和行业的需要而产生，并且扩展：

- 不是由本标准直接指定的；
- 遵循其他标准的规定；
- 可能在本标准未来的版本中试用。

##### 4.2.1 子集

一个实现必需支持起码一个学习者信息类型（个人、学业、管理、关系、安全、偏好、绩效、作品集）。不支持全部学习者信息类型的实现必须在实现一致性声明和一致性标签中简要说明支持哪一个子集。

参阅一致性标签（4.3）。

注：实现必须使用自动化技术来传送子集信息，以避免互操作性问题。

##### 4.2.2 严格一致的实现

严格一致的实现应至少符合其中一种情况：严格一致的编码、严格一致的API、严格一致的协议，或严格一致的数据应用。

一个严格一致的实现：

- 必须支持所有必备和可选数据元素；
- 不应使用、测试、访问或探测任何扩展性特性或扩展数据元素；
- 不应超出本标准指定的界限或最低峰值；
- 不应解释和生成依赖于任何未规定、未定义、由实现定义或本地规定行为的数据元素。

注：扩展特性和扩展数据元素的使用是未定义行为。

##### 4.2.3 一致的实现

一致的实现应至少是其中之一：一致的编码、一致的应用编程接口、一致的协议或一致的数据应用。

一致的实现：

- 必须支持所有必要和可选数据元素；
- 如经实现和数据交换双方的批准，只要严格一致的实现的意义和行为保持不变，可以使用、测试、访问或探测扩展特性或扩展数据元素；
- 不应支持或使用那些改变严格一致的实现的含义或行为的扩展特性和扩展数据元素；
- 如实现允许，可以超出规定指定的界限或最低峰值；
- 可以解释和生成依赖于任何实现定义、本地规定或未规定行为的数据元素。

注 1：扩展特性和扩展数据元素的使用是未定义行为。

注 2：所有严格一致的实现也都是是一致的实现。

注 3：如果一个实现，用扩展的方法重新定义标准的特性时，且这些特性改变了严格一致的实现的含义或行为，则该实现与本标准不一致。

##### 4.2.4 非一致的实现

该小节是信息性表述而不是规范性表述。

与本标准不一致（无论严格一致或一致）的实现是非一致的实现。

#### 4.2.5 数据元素的约束

该小节是信息性表述而不是规范性表述。

数据元素有4种约束属性：必需、可选、条件和扩展。约束属性与数据结构的有效性相关。

##### 4.2.5.1 必需数据元素

数据结构要求必需数据元素必须是有效的。所有数据集和数据实例都必须包含这些元素。所有数据应用都必须支持这些元素。

不支持或不包含一个或多个必需元素的实现是非一致性实现。

##### 4.2.5.2 可选数据元素

可选数据元素对于有效的数据结构来说，允许使用但非必需的。数据集和数据实例允许并非必须包含这些数据元素。因为所有数据信息库和数据读取器都必须支持所有有效的数据集，相应地，数据信息库和数据读取器也必须支持所有的可选数据元素。这有点含糊不清，因为“可选”并不是对数据信息库和数据读取器而言——约束属性“可选”是对于数据结构的有效性而言（此处的“可选”是对于数据结构的实例可选）。数据书写器生成和产生每个数据实例时，必须能够支持可选元素。

不支持一个或多个可选数据元素的应用是非一致性实现。

如果实现包含或支持可选数据元素，则它们的使用是由本标准指定。

包含或支持可选数据元素的实现，若其包含或支持这些可选数据元素的方法与标准不一致，也是非一致的实现。“可选”并不意味着实现可用任何方法处理数据元素（“在任何实现的选择”）；如果数据元素已实现，则本标准也详细说明其要求。

##### 4.2.5.3 条件数据元素

只在特定条件下被定义并包含在一个数据结构实例中的数据元素。

##### 4.2.5.4 扩展数据元素

扩展数据元素不能在严格一致的实现中出现。

只要该实现支持每个扩展数据元素，扩展数据元素可在一致的实现中出现。

示例1：实现允许和使用特定扩展数据元素。

示例2：数据交换双方允许使用特定数据元素。

示例3：没有使用其他扩展数据元素。

对于支持扩展数据的一致实现而言，这些元素各自都有约束属性。

示例1：必需扩展数据元素。

示例2：可选扩展数据元素。

示例3：条件扩展数据元素。

扩展数据元素范围内，约束属性决定的数据结构的合法性，例如，可选扩展数据元素：

——数据元素不是使数据结构合法的必要条件；

——是用于支持扩展数据元素的一致实现。

注：必需扩展数据元素可以导致互操作性问题，因为必需扩展数据元素

1) 要使数据结构有效必须有该数据元素；

2) 用于支持该扩展数据元素的一致实现。换言之，

——只有支持这种扩展数据元素的实现才可互操作；

——严格一致的实现不可互操作，因为对于互操作性是必须有扩展元素。

不存在既支持扩展数据元素或扩展特性，又支持所有语义互操作性的一般技术和方法；仅存在支持扩展数据元素的具体技术和方法，如，在上述范围内。

在上述情况外（不支持的环境中），使用扩展数据元素将导致未定义行为，可能会是：

——适当的。例如，如果该数据元素并不重要，则忽略它。

——不适当的。例如，即使是一个重要的数据元素，如安全分类，也忽略它；

——无害的。例如，错误信息；

——中断的。例如，许多错误信息；

——可预测的。例如，一个程序出错、非正常退出，或无缘故“挂起”；

——不可预测的。例如，一个程序出错、非正常退出，突然退出，或无缘故“挂起”。

处理未定义行为并没有普遍适用的方法。处理未定义行为的具体方法，可能有好的效果，也可能有副作用，也可能两者都有。

一些绑定“放松”对无法识别的数据元素的处理。通常，扩展数据元素引致未定义行为，但有些绑定“放松”对实现定义行为的要求，或忽略未知扩展数据元素——这两种“放松”处理要求（实现定义行为，忽略未知或扩展数据元素）可减少中断。

扩展数据元素既是约束（数据模型）特性，也是一致性水平特性（严格一致的与一致的）。

#### 4.2.6 数据元素的生命期

该小节是信息性表述而不是规范性表述。

以下生命期属性表示该标准过去、现在和将来的版本的协作目标。

注：生命期属性是与约束属性无关的。

##### 4.2.6.1 过时数据元素

过时数据元素在现今或以前版本的标准中定义。“过时”表示数据元素的定义在将来的版本中会被删除。实现不应该使用过时数据元素。使用过时数据元素的实现必须为将来的版本计划。

使用过时数据元素的实现不意味着该实现不一致的。在该版本的标准中，严格一致的实现和一致的实现仍可以使用过时数据元素。

“过时”特性是独立于约束属性的，因此可能存在过时必需数据元素、过时可选数据元素、过时条件数据元素和过时扩展数据元素。

##### 4.2.6.2 保留数据元素

保留数据元素是在标准的这个版本中没有定义的元素。保留数据元素是因为(1)在之前的版本定义过，或(2)将在以后的版本中定义。

严格一致的实现不可有保留数据元素。

保留数据元素可用于一致的实现中，当：

——保留数据元素曾经定义过；

——被定义为扩展数据元素；且

——实现和数据交换双方支持该扩展数据元素（看以下）。换言之，特定的实现可通过定义扩展实现，来扩展和覆盖“保留数据元素”（的非定义）。

虽然“保留”特性是独立于约束属性，但保留数据元素没有定义。因此，不存在保留必需数据元素、保留可选数据元素、保留条件数据元素和保留扩展数据元素，因为必需数据元素、可选数据元素、条件数据元素和扩展数据元素都意味定义数据元素，与“保留”的未定义特性有冲突。

#### 4.2.7 约束/生命期的递归/前后关系

该小节是信息性表述而不是规范性表述。

数据元素的约束属性和生命期属性适用于数据元素本身，但仅间接适用于它的子元素。在一个数据元素和其子元素的存在范围内，每个子元素都有它各自的约束和生命期属性（处于其他属性之中）。前后关系和约束/生命期属性的确定是递归的，可用于所有数据元素。

例如，一个数据元素 X 是可选的，且 X 有两个子元素：Y 是必需的而 Z 是可选的。设 P、Q 代表子元素 P 的子元素 Q，那么：

- 如果 X 不存在，那么 X.Y 和 X.Z 不可能存在；反之亦然，如果 X.Y 和 X.Z 存在，则 X 存在。
- 如果 X 存在，那么一致的数据实现就必须有 X.Y。
- 如果 X 存在，那么所有一致的实现可存在 X.Z。
- 如果 X 存在且 X.Y 不存在，那么实现就是不一致的。

因此，如果 X 存在则 Y 仅变为必需的。

#### 4.3 一致性标签

一致性标签是实现一致性声明的概括。通过手工、半自动和自动的方法，一致性标签可用于传递实现一致性声明的信息。关联或附加一致性标签的方法和技术都不在本标准之内。

如果一个实现并不支持所有信息类型，一致性标签必须用“单字母列表子集”（在 5.1 小节，“学习者信息”中的缩写列表）的符号来说明支持哪一个子集。例如，仅支持学习者偏好信息 (M) 和学习者绩效信息 (G) 的实现，可用“MG 子集”表示该子集。

一致性标签中绑定的描述，必须按照编码、API 和协议的顺序列出各绑定。列出的编码、API 和协议，每一个编码绑定必须与每一个 API 绑定同时有效；每一个 API 绑定必须与每一个协议绑定同时有效；每一个编码绑定必须与每一个协议绑定同时有效。

注：如果所有列出的编码、API 和协议的组合不是同时有效的，那么要用复合一致性标签指出一致性声明。

示例 1：“XML 和 DNVP 编码/ Java 和 JavaScript API”表示该实现与所有四个的任意组合一致：XML-Java, DNVP-Java, XML-JavaScript, DNVP-JavaScript。

示例 2：在之前的例子中，如果该实现不支持 XML-JavaScript 的组合，可以用两个一致性标签“XML 和 DNVP 编码/ Java API”和“DNVP 编码/ JavaScript API”描述实现的一致性声明。

以下是各种实现的汇总，可用于实现一致性声明及其一致性标签中：

——严格一致的学习者[子集]数据集：不依赖于绑定；所有必需数据元素必须存在；一些可选数据元素可以存在；扩展数据元素不可以存在。

——一致的学习者 [子集]数据集：不依赖于绑定；所有必需数据元素必须存在；一些可选数据元素可以存在；一些扩展数据元素可以存在。

——严格一致的学习者[子集]绑定数据实例：绑定必须指定；所有必需数据元素必须存在；一些可选数据元素可以存在；扩展数据元素不可以存在。

——一致的学习者[子集]绑定数据实例：绑定必须指定；所有必需数据元素必须存在；一些可选数据元素可以存在；可以存在一些扩展数据元素。

——严格一致的学习者[子集]绑定数据信息库：绑定必须指定；必须支持存储/提取所有必需数据元素属性，必需支持存储/提取所有可选数据元素；数据交换应用不可以存储/提取扩展数据元素，

——一致的学习者[子集]绑定数据信息库：绑定必须指定；必须支持存储/提取所有必需数据元素属性，必需支持存储/提取所有可选数据元素；数据交换应用可以存储/提取一些扩展数据元素。

——严格一致的学习者[子集]绑定数据读取器：绑定必须指定；仅可解释必需和可选数据元素，但是不可以解释扩展数据元素。

——一致的学习者[子集]绑定数据读取器：绑定必须指定；可解释必需和可选数据元素，也可以解释一些扩展数据元素。

——严格一致的学习者[子集]绑定数据书写器：绑定必须指定；必需生成所有必需数据元素；可以生成可选数据元素；不可以生成扩展数据元素。

——一致的学习者[子集]绑定数据书写器：绑定必须指定；必需生成所有必需数据元素；可以生成可选数据元素；可以生成扩展数据元素。

——严格一致的学习者[子集]绑定 API 环境：绑定必须指定；必须支持所有必需性和可选性数据元素，API 绑定的应用不可以探测扩展数据元素和扩展服务。

——一致的学习者[子集]绑定 API 环境：绑定必须指定；必须支持所有必需性和可选性数据元素。

——严格一致的学习者[子集]绑定 API 应用：绑定必须指定（按编码、API 和协议的顺序）；必须支持所有必需和可选数据元素；不可以探测扩展数据元素和扩展服务。

——一致的学习者[子集]绑定 API 应用：绑定必须指定（按编码、API 和协议的顺序）；必须支持所有必需性和可选性数据元素；可以在数据交换双方和本标准外的规范的允许范围内使用扩展数据元素和扩展服务。

——严格一致的学习者[子集]绑定协议：绑定必须指定（按编码、API 和协议的顺序）；必须支持所有必需和可选数据元素；协议绑定的应用不可以探查扩展数据元素和扩展服务。

——一致的学习者[子集]绑定协议：绑定必须指定（按编码、API 和协议的顺序）；必须支持所有必需和可选数据元素；可以在数据交换双方和本标准外的规范的允许范围内使用扩展数据元素和扩展服务。

注：一个实现可能会在它的实现的一致性声明(ICS)中作多于一种的一致性声明。

## 4.4 编码一致性

### 4.4.1 一致性的定义

严格一致的学习者编码必须至少是以下的一种：严格一致的数据集，或严格一致的数据实例。

一致的学习者编码必须至少是以下的一种：一致的数据集，或一致的数据实例。

### 4.4.2 数据集一致性

数据集一致性是独立于绑定的。

严格一致的数据集必须是：

- 结构上独立于绑定；
- 严格与本标准的功能性、概念模型和语义一致的；
- 必需包含所有必需数据元素；
- 可以包含可选数据元素；
- 不包含扩展数据元素。



一致的数据集必须是：

- 结构上独立于绑定；
- 与本标准的概念模型和语义一致的；
- 必需包含所有必需数据元素；
- 可以包含可选数据元素；
- 可包含扩展数据元素。

数据集的一致性评估必须通过以下途径进行：

- 以 GB/T 18221—2000 符号描述数据集；
- 校验是否符合本标准的要求。

#### 4.4.3 数据实例一致性

严格一致的数据实例必须：

- 是严格一致的数据集；
- 起码与一种学习者编码严格一致。

一致的数据实例必须：

- 是一致的数据集；
- 起码与一种学习者编码一致。

注1：术语“学习者编码”在以上两段中被使用，而它的要求在4.3小节“编码一致性”的第三段中指定。

注2：严格一致/一致的数据集、严格一致/一致的编码，和严格一致/一致的数据实例的区别在于：

- 1) 数据集是独立于绑定的数据实例；
- 2) 编码可以引用一个数据实例、一个数据实例集合或一个数据实例语法；
- 3) 一个严格一致/一致的数据实例与特定的绑定相关。

定义：支持，使用

在一致性范围内，术语“支持”和“使用”分别在每个学习者编码绑定中定义。

定义：测试，访问，探测

在一致性范围内，术语“测试”、“访问”和“探测”都定义为空操作，就是说，对于数据实例一致性，“测试”、“访问”、和“探测”不进行任何操作也没有任何效果。

基本原理

除了三个应用一致性（信息库、数据读取器、数据书写器），还有第四种一致性：数据实例。用户要为特定的数据实例声明一致性。

#### 4.5 API 一致性

严格一致的学习者 API 必须起码与一种学习者 API 绑定严格一致。

一致的学习者 API 必须起码与一种学习者 API 绑定一致。

注：API 绑定的一致性范式有两部分组成：API 应用和 API 环境。

学习者 API 必须与第5节：“功能性”、第6节：“概念模型”和第7节：“语义”的数据类型一致。

定义：支持，使用，测试，访问，探测

对于数据交换的参与者，以下术语在 API 一致性范围中定义：

——一个“支持”特性是一个可被 API 环境实现并可被任何学习者 API 的应用使用的特性；

——一个特性若被学习者 API 的应用读、写或操作，则称其被“使用”；

——如果一个学习者 API 的应用询问所说特性的是否存在，则该特性被“测试”；

——如果一个学习者 API 的应用试图读取或写入与特性相关的数据，则称该特性被“访问”；

——如果一个学习者 API 的应用，试图通过在“安全”环境下使用某特性（参阅以上“使用”），暗中测试它是否存在，而没有引起未定义行为，则该特性被“探测”。

注：API 一致性对 API 绑定和使用 API 绑定的应用都有要求，就是说，基于 API 一致性而对实现的一致性评估是由适当的 API 定义和用途确定。

#### 4.6 协议一致性

严格一致的学习者协议必须与至少一个学习者协议绑定严格一致。

一致的学习者协议必须与至少一个学习者协议绑定一致。

学习者 API 必须第 5 节“概念模型”和第 6 节“语义”的数据类型一致，并且应定义方法，建立和解除对通讯网络的支持。

定义：支持，使用，测试，访问，探测

关于学习者一致性，以下术语在协议一致性范围内中定义：

如果一个特性可被任何学习者协议的应用使用，则称这个特性被“支持”；

如果一个特性被学习者协议的应用读、写或其他操作，则称其被“使用”；

如果一个学习者协议的应用查询关于所说特性的是否存在，则该特性被“测试”；

如果一个学习者协议的应用试图读取或写入与特性相关的数据，则该特性被“访问”；

如果一个学习者协议的应用，通过试图在“安全”环境下使用某特性（参阅以上“使用”），暗中测试它是否存在，而没有引起未定义行为，则该特性被“探测”。

注：协议一致性对协议绑定和使用协议绑定的应用都有要求，就是说，基于协议一致性而对实现的一致性评估是由适当的协议定义和用途确定。

#### 4.7 数据应用一致性

##### 4.7.1 严格一致的数据应用

对于所有严格一致的应用，

——必需特性必须存在（或必须可用），而且必须与本标准一致。

——可选特性可以存在（或可用），如果存在（或可用）的话，必须与该标准一致。

——扩展的特性不可以直接被使用或不能被测试其存在性或可用性。

注 1：由于特性隐藏在实现中的，严格一致的应用可能会间接使用扩展特性。

注 2：一个严格一致的应用可以是最少一致，对于本标准来说却能得到最好的互操作性。严格一致的涉及到以下：

- 1) 特性的最小集合的评估、量度或可用性；
- 2) 数据应用中未使用特性探测；
- 3) 数据应用中未使用扩展特性集合。

##### 4.7.2 一致的数据应用

对于所有一致的数据应用，

——必需特性应该存在（或应该可用），且应该与本标准一致。

——可选特性可以存在（或可用），如果存在（或可用）的话，应该与本标准一致。

——扩展特性可以存在（或可用），可以对其存在性（或可用性）而进行测试，它们的使用和行为必须是实现定义的。

注：一致的数据应用可能更有用，但对于本标准来说却减弱了互操作的可能。一致性涉及到以下：

- 1) 特性的最小集合的评估，量度，或可用性；
- 2) 在实现允许的前提下，扩展特性的存在（或可用性）的特性探测，且/或之前与之（扩展特性）一致；
- 3) 在本标准以外指定的扩展特性。

#### 4.7.3 数据应用的多样性

共有三类严格一致/一致的应用：数据信息库，数据读取器和数据书写器。

基本原理

应用一致性共有三种不同的场合：数据信息库，数据读取器和数据书写器。供应商和管理人员或信息库要给出一致性声明（“本学习者信息库与规范一致”）。供应商要为工具提供一致性声明（数据读取器，“这是一致的学习者数据读取器，用于导入数据”；数据书写器，“这是一致的学习者数据书写器，用于导出数据”，或兼而有之）。

#### 4.7.4 数据信息库

数据信息库是存储和提取数据对象的数据应用。

严格一致的数据信息库必须：

- 1) 能接收数据集以备提取；
- 2) 对所接收的数据集使用严格一致的数据解释；
- 3) 把数据集存放到持久的存储设备中，并舍弃扩展数据；
- 4) 根据请求，发送以前保存的数据集；
- 5) 生成严格一致的数据集，发送出去；
- 6) 且与至少一个学习者编码绑定和至少一个学习者 API 或学习者协议绑定严格一致；

注 1：一个严格一致的数据信息库并不要求“保留”扩展数据元素，就是说，数据交换不应该依赖于扩展元素持久化到严格一致的数据信息库，但是也不禁止这种情况。

一致的数据信息库必须：

- 1) 能接收数据集以备提取；
- 2) 对所接收的数据集使用一致的数据解释；
- 3) 把数据集存放到持久的存储设备中，保留扩展数据；
- 4) 根据请求，发送以前保存过的数据集；
- 5) 生成一致的数据集，发送出去；
- 6) 且与至少一个学习者编码绑定和至少一个学习者 API 或学习者协议绑定一致。

注 2：一致的数据信息库可以添加、删除或修改被存储的扩展数据元素以备以后提取。

注 3：一致的数据信息库可以存储部分扩展数据，但并不要求存取所有扩展数据。

注 4：一致的数据信息库可以存取数据集以外的数据对象。

#### 4.7.5 数据读取器

数据读取器是一种数据的应用，它处理数据的过程为：

- 1) 消费数据，这些数据基于外部数据源；
- 2) 解释阶段 1 得到的数据，继而产生数据集，置于数据应用程序内部。

注 1：理论上，数据读取器分两个阶段处理信息（消费和解释），但是实现的设计并不受限制，可以分任意个阶段进行。

严格一致的数据读取器所解释的数据必须与本标准和至少一个本标准的绑定严格一致。

注 2：严格一致的数据读取器不解释扩展数据元素。

注 3：根据本标准的绑定，严格一致的数据读取器可以“忽略”数据扩展，例如，严格一致的数据读取器可以消费扩展数据却忽略（不解释）它。

一致的数据读取器应用解释的数据必须与本标准和至少一个本标准的绑定一致。

注 4：一致的数据读取器可解释扩展数据元素。

#### 4.7.6 数据书写器

数据书写器是一种数据应用，包括：从数据集中生成数据并产生数据。

注 1：理论上，数据书写器分两个阶段处理信息（生成和产生），但是实现的设计并不受限制，可以分任意个阶段进行。

严格一致的数据书写器生成的数据必须与本标准和至少一个本标准的绑定严格一致。

注 2：严格一致的数据书写器不生成扩展数据元素。

一致的数据书写器应生成的数据必须与本标准和至少一个本标准的绑定一致。

注 3：一致的数据书写器可生成扩展数据元素。

### 5 学习者概念模型介绍

#### 5.1 学习者信息框架

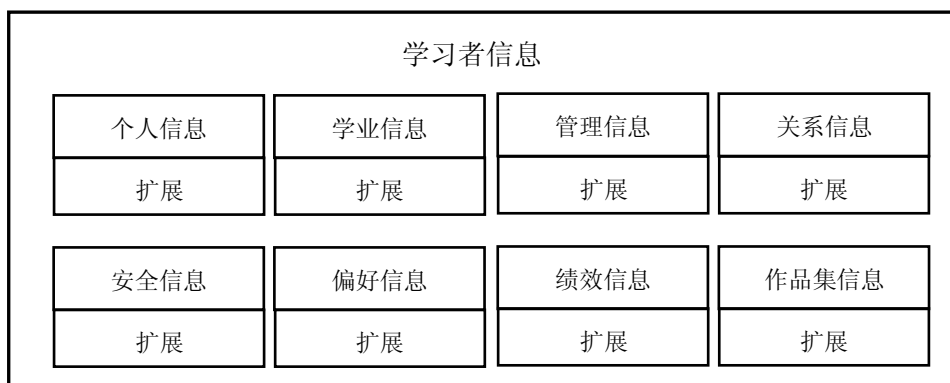


图1 学习者信息框架

如图1所示，本标准是学习者信息的一个子集，共分为八类信息。每类信息包含最基本的必要信息，和应用系统自定义的扩展信息。这八类信息为：

——个人信息

是基本的个人信息，这类信息一般与学习系统或者学习过程没有直接联系，部分信息属于个人隐私，是保密的。

——学业信息

是与学习者的学习相关的一些信息。

——管理信息

是学习者在学习系统中的简要管理信息。

——关系信息

是描述学习者与其他学习技术系统用户（如教师、代理或其他学习者）之间关系的信息。

——安全信息

是有关学习者安全凭证的信息。

——偏好信息

是学习环境或者学习者个人的一些可促进人机交互或者个性化服务的信息。

——绩效信息

学习者的学习经历、学习结果等的信息。

——作品集信息

是学习者的代表性作品及其引用的集合，用于描述和证明学习者的能力和成就。

注 1：应用可以扩展或合并这八类信息。应用可以利用数据库的键，把分散的信息类型的信息库联系在一起。例如，一个学习者个人信息的信息库利用学习者标识与学习者绩效信息的信息库联系在一起。

注 2：本标准并不要求这八种信息类型分散，但是许多实现都为了满足安全、管理、调整和系统性能的要求而进行分散存储。

单字母列表：

——N = 学习者个人信息，助记字母是名字“Name”的 N。

——L = 学习者学业信息，助记字母是学习“Learning”的 L。

——A = 学习者管理信息，助记字母是管理“Administration”的 A。

——R = 学习者关系信息，助记字母是关系“Relations”的 R，

——S = 学习者安全信息，助记字母是安全“Security”的 S。

——M = 学习者偏好信息，助记字母是我的配置“My configuration”的 M。

——G = 学习者绩效信息，助记字母是成绩“Grades”的 G。

——W = 学习者作品集信息，助记字母是作品“Works”的 W。

信息类型的原理概况

以下的每条提出一系列问题，例如，不仅有“本地”和“远程”的信息，还有许多级别和变化。

——本地 和 远程

学习者信息到学习者有不同的距离。本地信息，表现为在线可用、高性能访问和安全限制较少。远程信息，则常以间或可用、较低性能访问和安全限制多为特征。可试把本地校园服务器和远程校园服务器作比较。

——私有 和 公开

学习者信息有不同的隐私和公开级别。本标准很大程度受用户、机构和互联网对保密问题的关注所影响。保密性问题是：提供什么信息给谁？学习者规范的安全范例会以〈须知〉作指导。可试比较学习者个人信息（保密信息）和学习者偏好信息（公用信息）；或比较（有限制）公开的和（有限制）私有的绩效信息。

——学习过程 和 其他用途

学习者信息可以提取对学习技术系统有用和必需的信息及其他信息，以便优化学习经历。调整学习内容的顺序可改善学习过程。试比较绩效信息（必要的或有助于改善学习过程）和个人信息（改善学习过程不必要的）。

——内容生成 和 学习者生成

学习者信息可以由学习技术系统生成、学习者自己生成，或两者共同生成，或由其他生成，或兼而有之。试比较学习者绩效信息（利用学习内容生成）和学习者作品集信息（由学习者创建和维护）。

## 5.2 信息类型与数据信息库

一个信息库可以存储一种或多种信息类型。一个数据集可以存储一种或多种数据类型。

示例 1：分离的数据信息库：一致性实现对每一种信息类型有一个数据信息库：一个学习者个人信息库，一个学业信息库，一个管理信息库，一个关系信息库，一个安全信息库，一个学习者偏好信息库，一个学习者绩效信息库、一个学习者作品集信息库。

示例 2：组合的数据信息库：一致性实现有一个信息库存放所有类型学习者信息。在以下例子中，安全管理员不限制对“学习者偏好方面”的访问（例如，学习者偏好信息），而限制对“学习者个人方面”（例如，学习者个人信息）访问。由此，学习者个人信息可“保密”而学习者偏好信息则“公开”，都可在同一个信息库中进行访问。用户或应用程序并没有意识到是在同一个信息库中访问的所有信息类型。

### 5.3 数据访问模型

学习者数据集是学习技术系统中相关学习者信息的集合。这些信息包括一种或多种类型的信息：个人、学业、管理、关系、安全、偏好、绩效和作品集。

学习者数据集的数据交换受以下因素影响：

——学习者编码

学习者数据集可以根据学习者编码绑定进行编码。通过学习者编码规范以外的方法达成共识，数据交换各方更方便地进行数据交换。

——学习者应用编程接口 (API)

应用编程接口绑定是影响数据交换的控制传输机制（控制从主叫传到被叫）。

——学习者协议

协议绑定既是控制传输机制又是数据传输机制。

这些绑定合起来称为“学习者编码、API 和协议”。

注：学习者数据应用可以使用多于一种学习者编码、API 和协议。

以下是数据访问的概念模型：

——数据对象模型

一个数据对象应至少是一个数据元素或一个由实现定义的对象。

——数据存储模型

数据，包括数据集，可以存储在一个数据对象里，通过一个或多个标识符进行引用。

——数据提取模型

数据，包括数据集，可以从数据对象中提取，通过一个或多个标识符进行引用。

——数据归类模型

作为数据元素的数据对象必须有一种数据类型。数据类型可以指定值域、表达方式、编码、存储、梗概、向其他类型的转换、方法和操作。学习者数据元素的数据类型的定义由本标准根据 GB/T 18221—2000 的符号和语义给出。

——数据属性模型

数据属性是实现定义的对象，与数据对象相关。如同数据对象，属性也可被访问。

注：属性(“attributes”)也被称为“特性”(“properties”)。

——数据信息库访问模型

学习者绑定定义了所有对数据信息库的访问。

——数据信息库安全模型

参阅安全模型(5.6)。

——数据持续模型

数据对象的生存期必须是由实现定义。

——数据导向模型

数据结构导向的技术在学习者绑定中定义。

——数据识别模型

识别、加注标签、命名空间及其相关技术必须是实现定义。

——数据引用模型

信息库为以后的去引用，可以创建对数据对象的引用。引用的命名规则、生存期和引用范围必须是实现定义的。

——数据去引用模型

信息库可以根据引用访问数据对象，例如，去除一个引用。去引用的方法是实现定义的。

——数据索引模型

信息库的索引方法是实现定义的。

注：术语“索引”用于数据库系统范围中，例如，组织数据库记录的方法。

——数据搜索模型

信息库的搜索方法是实现定义的。

#### 5.4 远程，分布和漫游系统

作为在全局命名空间内查找、存储和提取信息的基础，远程特性需要全局命名空间。命名空间的规则和分辨方法必须是实现定义的。

分布的方式有两种：单一信息库存储一种或多种信息类型，多个信息库的操作同单一信息库。同步、复制、提交、回滚和倒退的方法必须是实现定义的。

间断连接的游离用户允许漫游特性。保持服务的动态质量和连接的持续性的方法必须是实现定义的。

#### 5.5 粒度水平的相互关系

学习者绩效信息不同粒度水平的相互关系必须是实现定义的。

示例：如果一个老师改变了一门课的成绩（较小粒度/密度），什么时候平均成绩（较大粒度/密度）会自动变化呢？

#### 5.6 安全模型

安全性是由安全边界定义和界定的。以下特性必须是实现定义的：

- 安全边界的范围。
- 内向安全威胁的属性、类型和可接受程度。
- 外向安全威胁的属性、类型和可接受程度。
- 安全强度。
- 安全特性的参数化、设置、协商和降低。
- 安全边界一体化的管理。

互操作性约定在本标准以外定义。

注：安全模型与 ISO/IEC 17799-1 “信息安全管理实际代码”协调。

以下是在概念模型中定义的安全特性：

——基于会话视图的安全模型。安全特性是基于每个会话、每个视图。每次安全会话由访问者（要求访问的用户或代理）发起。访问者输入安全凭证，用于验证或授权访问者或两者。一个视图代表学习者信息的一部分（这里的“视图”与数据库中的“视图”相似）。每一个已建立的视图代表一次会话。就是说，“会话”表示访问过程，而“视图”表示访问的范围。

——安全参数协商模型。会话前、会话期间和会话后的安全参数，由数据交换双方商定。安全参数在学习者绑定中定义。

——安全扩展模型。无法预知是否会使用附加的安全特性。安全扩展的合并方法在学习者绑定中定义。

——访问控制模型。访问者可以（单独或集体地）读、写、新建、删除数据元素或改变数据元素的属性。如果有其他方法，必须是实现定义的。

——识别模型。识别学习者的方法必须是实现定义的。

——验证模型。验证用户的方法不在本标准范围内。

——去标识模型。除了学习者个人信息，所有信息都要去除标识。用户和他们的信息去除标识的方法不在本标准范围内。注意：管理员，或者学习者，有责任选择适当的学习者标识，不泄漏学习者的身份。例如，选择学习者的姓名作为标识，以去除标识观点，不是个好办法。有很多好方法可用于去标识，包括短时间使用随机数。

——授权模型

授权操作的方法必须是实现定义的。

——委托模型

委托管理、授权或凭证的方法必须是实现时定义的。

——接受模型

接受的方法必须是实现定义的。

——拒绝模型

拒绝数据、用户和凭证的方法必须是实现时定义的

——保密模型。注意：本标准未指定保密模型和保密要求，但支持安全的架构和途径，以实现各种各样的保密架构。

——机密模型。本标准未指定机密模型和机密要求，但支持访问控制和划分信息类型，以实现各种各样的机密架构。

——加密模型。本标准未指定加密模型和加密要求，但支持安全的架构和技术，以整合多种加密模型和技术。

——数据完整性模型。本标准未指定数据完整性模型和数据质量要求，但支持保障信息的架构和途径，以实现各种各样数据完整性架构。

——证书的认可。本标准不要求确认学习者绩效信息或作品集信息，但支持参数设定，以自动确认两种学习者信息。

——数字签名模型。本标准未指定数据签名模型和数据签名要求，但支持多个签名架构和技术，以整合各种各样的数字签名模型、政策和技术。数字签名模型与 ISO/IEC 15945 “支持数据签名应用的受信第三方服务标准”协调。

## 6 语义

无论选择哪一个学习者绑定和学习者编码，学习者信息的含意都是一样的。

注1:绑定和编码的选择是为了方便数据交换的参与者。

注2:学习者编码绑定定义了本节中每个数据类型的映射。

注3:学习者API绑定定义了本节中每个操作和数据类型的映射。

注4:学习者协议绑定定义了本节中每个操作和数据类型的映射，而且为建立和拆除通信支撑网络定义了各种方法。

注5:在整节中都会使用简写“SPM”，其意思是“最低峰值”。SPM给一致的实现设定下限，除非预先安排，应用不能让实现的设定超出SPM范围。

### 6.1 一般的数据操作

学习者信息必须支持以下对数据集的数据管理操作：

——创建操作

新建某个数据类型的实例，如个人信息。

注：比较在应用的内存空间中，暂存中和数据库中创建的记录的生存期。

——删除操作

把一个数据类型的实例从储存器中丢弃。

注：比较删除一个记录的各种技术：擦除、懒惰存储管理、垃圾回收。

——擦除操作

擦除一个数据类型的内容，且不可恢复。

注：删除操作只实现了数据对应用程序的不可用，并没有实现数据的不可恢复性。



## ——复制操作

以同样内容新建一个数据类型的实例。

## ——转移操作

通过改变信息的存储(隐式标签转换)或改变标签(显式标签转换),改变与数据类型实例相关的标签。

示例:隐式标签的转换可能会受将数据从源复制到目标并删除源数据的影响(如将第15个元素复制到第3个)。显式标签转换会受改变某信息目录中标签的影响。

## ——标签操作

创建(或删除)与信息实例相关的名字,名字由“调用方”指定。

## ——导向操作

用一种命名方法(绝对的,相对的,完全的,渐进的)为数据类型的实例定位。

## ——查找操作

查找与搜索条件相匹配的数据类型实例,并通过引用、标签或拷贝返回找到的信息。

## ——引用操作

为一个数据类型实例创建句柄。

注:标签操作和引用操作的差别调用程序指定标签的名字,而被调用者指定引用的名字。

## ——去引用操作

以句柄(引用操作创建)访问数据类型实例。

## ——集合操作

连结若干实例(一种或多种数据类型)到一个容器内。

## ——分解操作

从一个容器中提取出数据类型的实例。

学习者模型绑定定义了管理访问数据的操作方法及该操作的有效性。

注:有些操作可能无法执行,例如:在只读媒介上创建或删除数据。在一些情况下,这些操作需要存在,但是使用它们会导致运行错误,例如:一个API必须支持创建操作,但是当创建操作被尝试执行时会导致错误。

## 6.2 具体应用数据操作

学习者信息必须支持以下对数据集的具体应用数据操作:

## ——查找操作

按照一定规则,学习者记录可被查找。例如,“哪些学习者成绩在60分以下?”,“哪些学习者的作品集集中包含视频文件?”

## ——累积操作

学习者记录可被累积、聚集或分析。例如,“三年级学生成绩的平均分是多少?”,“个别学习者的平均分是多少?”

## ——时间压缩和扩展操作

学习者记录可以是不同的粒度水平。时间压缩将记录缩为较大粒度水平。时间扩展通过插补,创建更细粒度水平的记录。例如,通过期中成绩和期末成绩可以计算出一个最终成绩,经过压缩后,数据集缩减到只剩最终成绩。

## ——排序操作

对学习记录可根据某种准则进行排序。例如,可按用户名字的字母顺序排序。

## ——连接操作

多个表可以通过指定列连接起来。

学习者绑定定义了访问应用具体数据的操作方法及该操作的有效性。

注：应用程序如果使用这些操作，必须先检测实现是否支持这些操作。参见 API 绑定、协议绑定、信息库说明，以获得更多信息。

### 6.3 数据兼容性

学习者信息可以支持以下对数据集的数据兼容性操作：

——数据类型的升级

将低级别的值，转换为高级别。例如，整数转换为实数；长度为 10 的字符数组转换为长度为 20 的字符数组；记录 {姓名，地址} 转换为超集记录 {姓名，地址，电话}。

——数据类型的降级

将高级别的值，转换为低级别。例如，实数转换为整数；长度为 20 的字符数组转换为长度为 10 的字符数组；记录 {姓名，地址，电话} 转换为子集记录 {姓名，地址}。

——数据类型的转换

把值从一种数据类型转换为另一种数据类型。例如，整数转换为字符串；{姓，名} 转换为 {姓名}。

——文本格式化

数值转换为文本表达或文本表达转换为数值。

学习者绑定定义了访问数据兼容性的操作方法及该操作的有效性。

注：应用程序如果使用这些操作，必须先检测实现是否支持这些操作。参见 API 绑定、协议绑定、信息库说明，以获得更多信息

### 6.4 数据类型

本节介绍的数据类型是用于举例解释本标准所定义的信息，注意不是规定了一个信息在应用实现中的取值类型。

#### 6.4.1 基本数据类型的定义

以下是被各类学习者信息引用的基本数据类型及其解释。

表1 基本数据类型及其定义

编号	类型名称	解 释	类型定义
1	字符串	以两个字节为单位	characterstring(ISO-10646-1)
2	字节串	以一个字节为单位	octetstring
3	情境标签	其元素可以描述数据元素被引用的情境值的意义必须在实现时进行定义	context_label_type = characterstring(iso-10646-1) // SPM: 500
4	数组列表	多个数组元素的简写。	arraylist(type_spec, size)= array(0..size-1)of(type_spec)  //N 个数据类型，N=size //type_spec:某种数据类型

表 1 (续)

编号	类型名称	解 释	类型定义
5	多语言字符串	其元素是一种标明所用语言的字符串。其中 locale 指明字符串所使用的语种  参阅 GB/T 2695-2000 和 ISO3166 世界各国和地区名称代码	mlstring_type = record ( string: characterstring(iso-10646-1), // SPM: 1000 locale: characterstring(iso-10646-1) // SPM: 255 )
6	多语言字符串数组	由多语言字符串组成的数组	mlstring_array_type(limit) = array(0..limit-1)of(mlstring_type)
7	名值对	由属性名和取值构成的偶对 1) bucket_name: 名字-值对的名字部分 2) bucket_value: 名字-值对的取值部分	bucket_type = record ( bucket_name: characterstring(iso-10646-1), // SPM: 200 bucket_value: octetstring // SPM: 4096 )
8	标识符	连结数据库数据的内部标识符。标识符的含意、名域、范围和识别方法都必须在实现时进行定义 1) context_label: 引用这种数据元素的情境 2) identifier_type: 标识符的类型 3) identifier_value: 标识符的值	learner_identifier_type = record ( context_label : context_label_type, identifier_type : identifier_type_type, identifier_value : octetstring // SPM: 1024 )
9	外标识符	使学习者信息和信息库相关联的外部标识符。外标识符的含意、名域、范围和识别方法都必须在实现时进行定义	hid_type = record ( context_label : context_label_type, identifier_type : identifier_type_type, identifier_value : octetstring // SPM: 1024 )
10	标识符类型	标识符类型的类型, 反映了学习者标识符的多样性。其取值不在本标准中说明	identifier_type_type = octetstring // SPM: 256

表 1 (续)

编号	类型名称	解 释	类型定义
11	数据认证	<p>数据认证的元素将说明绩效信息或作品集等信息中的记录的有效性</p> <p>1) source: 谁认证这条记录。</p> <p>2) method: 用什么方法认证。</p> <p>3) parameter: 使认证有效的必要选项和参数。</p> <p>注:有效的 source, method 和 parameter 是 identifier 有效的必要条件。</p> <p>4) subset: 在这个记录中已被认证的元素列表。</p> <p>5) identifier: 传送给认证者的标识符, 例如一个证书 ID。</p>	<pre>data_certification_type = record ( certification_source : octetstring, // SPM: 2048 certification_method : octetstring, // SPM: 1024 certification_parameter : octetstring, // SPM: 16384 certification_subset : octetstring, // SPM: 1024 certification_identifier : octetstring, // SPM: 2048 certification_bucket : arraylist(bucket_type, 100) )</pre>

#### 6.4.2 复杂数据类型的定义

表2定义了各类学习者信息所引用的复杂数据类型。

表2 复杂类型的定义

编号	类型名称	解 释	类型定义
1.	名字类型	<p>1) official_name: 学习者的合法名字。如, 学习者护照上的名字</p> <p>2) full_informal_name: 在非正式场合中所用的非正式全名</p> <p>注: 在个人信息的“名字”中引用该类型。</p>	<pre>name_type= record ( context_label : context_label_type, official_name : arraylist(formal_name_type, 5), full_informal_name : arraylist(full_name_type, 5), )</pre>
2.	正式名字类型	<p>姓/名</p> <p>primary/secondary</p> <p>注: 在‘名字类型’中引用该类型。</p>	<pre>formal_name_type = record ( primary : characterstring(iso-10646-1), // SPM: 70 secondary : characterstring(iso-10646-1) // SPM: 70 )</pre>

表 2 (续)

编号	类型名称	解 释	类型定义
3.	全名类型	该数据类型可以广泛地兼容各种形式的名字。 注：在‘名字类型’中引用该类型。	full_name_type= characterstring(iso-10646-1) // SPM: 140
4.	电话类型	1)context_label : 适用范围。 2)identifier_type: 电话号码的类型。 3)phone_number: 电话号码。 注：在个人信息的“电话”中引用该类型。	telephone_type = record ( context_label : context_label_type, identifier_type : identifier_type_type, phone_number : octetstring // SPM: 50 )
5.	电子邮件类型	学习者的联络电子邮件，包括： 1)email_address_type: 电子邮件地址的类型。 2)email_address: 电子邮件地址。 注：在个人信息的“电子邮件”中引用该类型。	email_contact_type = record ( context_label : context_label_type, email_address_type : identifier_type_type, email_address : octetstring // SPM: 255 )
6.	邮寄地址类型	1)devliery_* 邮寄地址，包括： ——country 国家 ——territory 地区 ——city 城市 ——street_name 街道 ——street_type 街道类型 ——street_id_number 街道编号 ——office 邮局 ——routing 支局 ——po_box 邮箱号 ——postcode 邮政编码 ——supplementary_address 附加地址  2)organization_*: 收件单位，包括： ——name 名称 ——activity 部门 ——division 分支	Postal_address_type = record ( context_label : context_label_type, delivery_country : characterstring(iso-10646-1) // SPM: 35 delivery_territory : characterstring(iso-10646-1), // SPM: 35 delivery_city : characterstring(iso-10646-1), // SPM: 35 delivery_street_name : characterstring(iso-10646-1), // SPM: 70 delivery_street_type : characterstring(iso-10646-1), // SPM: 35 delivery_street_id_number : characterstring(iso-10646-1), // SPM: 20 delivery_office : characterstring(iso-10646-1), // SPM: 35

表 2 (续)

编号	类型名称	解 释	类型定义
	邮寄地址类型	3)addressee_* 收件人, 包括: ——care_of_address 代理 ——suffix 称谓 ——name_family 姓 ——name_given 名 ——title 称号 ——function 职务 ——occupation 职位  <b>注:</b> 在个人信息的“邮寄地址”中引用该类型。 <b>注:</b> 在邮寄申请表中, 收件人的名称、称谓等都要与学习者名字信息中的值相一致。	<pre> delivery_routing :   characterstring(iso-10646-1), // SPM: 20 delivery_po_box :   characterstring(iso-10646-1), // SPM: 20 delivery_postcode :   characterstring(iso-10646-1), // SPM: 20 delivery_supplementary_address :   characterstring(iso-10646-1), // SPM: 70  organization_name :   characterstring(iso-10646-1), // SPM: 70 organization_activity :   characterstring(iso-10646-1), // SPM: 70 organization_division :   characterstring(iso-10646-1), // SPM: 70  addressee_care_of_address :   characterstring(iso-10646-1), // SPM: 70 addressee_occupation :   characterstring(iso-10646-1), // SPM: 70 addressee_function :   characterstring(iso-10646-1), // SPM: 70 addressee_title :   characterstring(iso-10646-1), // SPM: 35 addressee_name_family :   characterstring(iso-10646-1), // SPM: 70 addressee_name_given :   characterstring(iso-10646-1), // SPM: 70 addressee_name_suffix :   characterstring(iso-10646-1), // SPM: 35           )         </pre>

表 2 (续)

7.	关系类型	<p>1)others_identifier_list 与学习者相关的其他用户的标识符。</p> <p>2)relations_label_list 描述关系属性的标签。</p> <p>3)relation_to_them_list 学习者和相关用户的关系。</p> <p>4)relation_to_me_list 相关用户和学习者的关系。</p> <p>注：在关系信息的“关系”中引用该类型。</p>	<pre>Relationship_type = record ( others_identifier_list :   arraylist(identifier_type, 200),   relations_label_list :     arraylist(mlstring_type, 200),   relation_to_them_list :     arraylist(relationship_type_type, 200),   relation_to_me_list :     arraylist(relationship_type_type, 200) )</pre>
8.	关系类型的类型	<p>1)同学: 与学习者分享学习体验的其他学习者。</p> <p>2)属于: 学习者与所在团体的关系。</p> <p>3)包含: 与学习者有联系的团体。</p> <p>4)教师: 学习者的老师。</p> <p>5)教员: 参与管理学习者的人员, 包括教务人员、指导员等。</p> <p>6)学生: 该老师的学生。</p> <p>注：在‘关系类型’中引用该类型。</p>	<pre>Relationship_type_type = enumerated ( “同学”,   “学生”,   “教师”,   “教员”,   “属于”,   “包含” )</pre>
9.	安全凭证类型	<p>1)context_label: 引用这种数据元素的情境。</p> <p>2)credential_type: 凭证的类型。</p> <p>3)credential_value: 凭证的值。</p> <p>注：在安全信息的“安全凭证”中引用该类型。</p>	<pre>Security_credential_type = record ( context_label :   context_label_type,   credential_type :     identifier_type_type,   credential_value :     octetstring     // SPM: 8192 )</pre>
10.	设备类型	<p>1)security: 保密;</p> <p>2)text: 文本;</p> <p>3)speech: 语音;</p> <p>4)graphic: 图象;</p> <p>5)audio: 音频;</p> <p>6)video: 视频;</p> <p>7)tactile: 触觉;</p> <p>8)session: 会话;</p> <p>9)other: 其他未指定的设备。</p>	<pre>Device_preference_type = record ( security_list :   arraylist(device_io_preference_type, 20),   text_list :     arraylist(device_io_preference_type, 20),   speech_list :     arraylist(device_io_preference_type, 20),   graphics_list :     arraylist(device_io_preference_type, 20), )</pre>

表 2 (续)

	设备类型	注：在偏好信息的“设备偏好”中引用该类型。	<pre>audio_list :     arraylist(device_io_preference_type, 20), video_list :     arraylist(device_io_preference_type, 20), tactile_list :     arraylist(device_io_preference_type, 20), session_choosing :     arraylist(device_io_preference_type, 20), other :     arraylist(device_io_preference_type, 100) )</pre>
11.	设备输入输出类型	<p>1)input: 相应的输入设备。</p> <p>2)output: 相应的输出设备。</p> <p>注：在“设备类型”中引用该类型。</p>	<pre>Device_io_preference = ( input :     device_parameter_type, output :     device_parameter_type )</pre>
12.	设备参数类型	<p>1)name: 国际统一的设备名称。</p> <p>2)rating: 设备的有用程度。</p> <p>3)priority: 和相似设备相比的优先级（或重要性）。</p> <p>4)device_name: 机器可识别的设备名字。</p> <p>5)device_type: 机器可识别的设备类型描述，其值不在本标准规定。</p> <p>6)method: 访问的方法，其值不在本标准规定。</p> <p>7)protocol: 用于访问设备的协议，其值不在本标准规定。</p> <p>8)coding: 所用的编码技术，其值不在本标准规定。</p> <p>9)encoding: 所用的编码技术，其值不在本标准规定。</p> <p>10)other: 其他信息。</p> <p>注：在‘设备输入输出类型’中引用该类型。</p>	<pre>Device_parameter_type = ( context :     context_label_type, preference_name :     characterstring(iso-10646-1), preference_rating :     integer, preference_priority :     integer, device_name : characterstring(iso-10646-1), device_type :     characterstring(iso-10646-1), method :     characterstring(iso-10646-1), protocol :     characterstring(iso-10646-1), coding :     characterstring(iso-10646-1), encoding :     characterstring(iso-10646-1), other :     characterstring(iso-10646-1) )</pre>



### 6.4.3 未定义的数据类型

以下是本标准未定义的，但引用的数据类型，这些类型可以在其他标准或标准中找到。

- MIME\_type 多用途网际邮件扩展类型。
- URI\_type 统一资源标识类型。
- LOM\_reference\_type 学习对象元数据引用类型，见 GB/T 21365-2008 中的学习对象元数据。
- competency\_definition\_reference\_type 学力引用类型，暂时用学力标识类型代替。
- performance\_reference\_type 绩效信息引用类型，暂时用绩效信息的外标识类型代替。

## 7 学习者信息定义

### 7.1 个人信息

以下信息是学习者个人信息。个人信息是学习者在学习技术系统外就有的信息，与绩效度量没有直接联系。

表3 个人信息

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
1.	个人信息	学习者的个人信息	M	集合	personal_info_type	
1.1	标识	连结信息的内部数据库键	M	标识符	<pre>personal_identifier_list: arraylist( learner_identifier            _type,200)  // learner_identifier_type: // 标识符类型。</pre>	<pre>Personal_identifier_ list= ( identifier_type= "pointer", identifier_value= "0x12345678" )</pre>

表 3 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
1.2	外标识	连接不同信息库中信息的外部标识符。	M	外标识符	personal_hid_list: arraylist(hid_type,200)  // hid_type: 外标识符类型。	personal_hid_list = ( identifier_type= " GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符", identifier_value= "00112233" )
1.3	名字	学习者的名字	M	名字	name_list : arraylist(name_type,40)  // name_type: 名字类型。 // 参阅: 表 3	name_list= ( official_name = ( primary="阳", secondary="昊", ), full_informal_name = "小昊" )
1.4	电话	学习者的电话	O	电话	telephone_list : arraylist(telephone_type,15)  // telephone_type: 电话类型。 // 参阅: 表 3	telephone_list= ( context_label = "家", identifier_type= "通话", phone_number= "+86 20 8711 2800", ), ( context_label= "办公室", identifier_type = "通话", phone_number = "+86 20 8711 2800" ), ( context_label = "办公室", identifier_type = "传真", phone_number = "+86 20 8711 2800" )

表 3 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
	电话	学习者的电话	O	电话		( context_label = “紧急情况”, identifier_type= “通话”, phone_number = “+86 20 8711 1234” )
1.5	电子邮件地址	学习者的电子邮件地址。	M	电子邮件	email_contact_list : arraylist(email_contact_type,25)  // email_contact_type: // 电子邮件类型。 // 参阅：表 3	email_contact_list = ( ( context_label = "家", email_address_type = "rfc822", email_address = "foobar@mail.com" ), ( context_label = "办公室", email_address_type = "rfc822", email_address = "foo@bar.com" ))
1.6	邮寄地址	学习者的邮寄地址	O	邮寄地址	postal_address_list: arraylist(postal_address_type,10)  // postal_address_type: // 邮寄地址类型。 // 参阅：表 3	postal_address_list= (context_label = “学校”, addressee_name _given = “昊”, addressee_name _family = “阳”, delivery_street_type = “街”, delivery_street_name = “五山”, delivery_street_id_number = “123”, delivery_city = “广州”, delivery_territory = “广东”,

表 3 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
	邮 寄 地 址	学习者的邮寄地址	O	邮寄地址		delivery_routing = “510640”, delivery_country = “中国” )
1.7	个人档案	学习者的相关档案	O	集合	demography	
1.7.1	国籍	学习者的现有国籍	M	字符串	nationality: characterstring(iso-10646-1) //SPM:50 // 参阅 GB/T 2695-2000 和 //ISO3166 世界各国和地区名称 //代码	nationality="cn"
1.7.2	民族	学习者所属民族	O	字符串	folk: characterstring(iso-10646-1) //SPM:50 //参阅 GB 3304-1991 中国各民族//名称的罗马字母拼写法和代码	folk="01"
1.7.3	籍贯	学习者的祖父辈的家乡	O	字符串	native_place: characterstring(iso-10646-1) //SPM:100	native_place="中国广东省湛江市"
1.7.4	出生地	学习者的出生地点	O	字符串	birth_place: characterstring(iso-10646-1) //SPM:100	birth_place="中国广东省广州市"
1.7.5	出生日期	学习者的出生日期	O	时间	birth_date: time(second,10,0) // yyyymmddThhmmss //参阅 GB/T 7408—2005	birth_date="19791001"
1.7.6	性别	学习者的性别	M	整数	gender: integer, //参阅 GB 2261-1980 人的性别代//码	gender=1
1.7.7	身份证号	学习者的身份证号	O	字符串	id_number: characterstring(iso-10646-1) //SPM:64 ( 国家统一编号 )	id_number="010101020202030303"
1.7.8	政治面貌	学习者的政治身份	O	字符串	political_affiliation: integer, //参阅 GB 4762-1984 政治面貌代//码	political_affiliation =01

表3 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
1.7.9	语言能力	学习者擅长的语言, 可为学习时提供参考	M	字符串	language: characterstring(iso-639) //SPM:32 //参阅 GB 4880—1991 和 // ISO—639 语种名称代码	language="zh"
1.7.10	健康状况	学习者的健康状况	O	字符串	health: integer, //参阅 GB 4767-1984 健康状况代// 码	health=03
1.8	扩展元	提供了有限度的扩展能力	O	名值对	personal_bucket: arraylist(bucket_type,100)  // bucket_type: 名值对类型	personal_bucket = ( ( name= "社会保险号", value= "123-45-6789" ), ( name="付款方式", value="现金" ) )
注 1: “约束” 栏内 M 表示必备数据元素; O 表示可选数据元素; C 表示条件元素, 其存在依赖于其他元素的值。 注 2: 本标准中“类型定义”用 GB/T 18221—2000 格式 (数组列表) 对相应的数据元素的类型进行说明。下同。						

## 7.2 学业信息

以下信息是学习者学业信息。学业信息是在教育过程中与学习者的学习密切相关的一些信息, 对学习者的学习有较大影响。

表4 学业信息

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
2	学业信息	学习者的学业信息	M	集合	learning_info_type	
2.1	标识	连结信息的内部数据库键	M	标识符	learning_identifier_list : arraylist( learner_identifier _type,200)  // learner_identifier_type: // 标识符类型	learning_ identifier_list = ( identifier_type= "pointer", identifier_value= "0x12345678" )

表 4 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
2.2	外标识	连接不同信息库中信息的外部标识符	M	外标识符	learning_hid_list: arraylist(hid_type,200)  // hid_type: 外标识符类型	learning_ hid_list= ( identifier_type= "GB/T 21366-2008 信息技术 学习、教育 和培训 参与者标识符 ", identifier_value= "44556677" )
2.3	学校	学习者就读的学校	M	字符串	institute: characterstring(iso-10646-1) //SPM:64	institute= "华南理工大学"
2.4	专业	学习者所学的专业	O	字符串	major: characterstring(iso-10646-1) //SPM:64 //参阅 GB/T 16835-1997 高等学校// 本科、专科专业名称代码	major= "信息工程"
2.5	班级	学习者所在的班级	O	字符串	class: characterstring(iso-10646-1) //SPM:64	class="199804"
2.7	学习计划	描述学习者的学习计划。如, 课程编码词汇表/相应的学分/修读时间	O	字符串	learning_scheme: characterstring(iso-10646-1) //SPM:1024	learning_scheme = " 第一年在华南理工大学学习基础课, 第二第三年在 MIT 学习, 第四年在 IBM 实习"
2.8	学习完成情况	描述学习者当前已经完成的课程或学分等	M	字符串	credit: characterstring(iso-10646-1) //SPM: 256	credit="56 学分"
2.9	扩展元	提供了有限度的扩展能力	O	名值对	learning_bucket: arraylist(bucket_type,100)  // bucket_type: 名值对类型	

## 7.3 管理信息

以下信息是学习者管理信息。管理信息是教育管理系统中与学习者相关的信息，有利于学习者学习的正常进行。

表5 管理信息

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
3	管理信息	学习者在学习技术系统中的管理信息	O	集合	administration_info_type	
3.1	标识	连结信息的内部数据库键	M	标识符	administration_identifier_list : arraylist( learner_identifier_type,200) // learner_identifier_type: // 标识符类型	administration_identifier_list = ( identifier_type=“pointer”, identifier_value=“0x12345678” )
3.2	外标识	连接不同信息库中信息的外部标识符	M	外标识符	administration_hid_list: arraylist(hid_type,200) // hid_type: 外标识符类型	administration_hid_list = ( identifier_type=“GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符”, identifier_value=“88990011” )
3.3	费用信息	收费项与收费方式的集合	O	集合	fare_list: arraylist(fare_type,200); fare_type: record: ( fee: characterstring(iso-10646-1), amount: float, pay_method: integer )	fare_list = (fee= “学费”, amount =4500, pay_method=1 )
3.3.1	收费项	学校要求的各种费用的名称	M	字符串	fee: characterstring(iso-10646-1) //SPM:50	fee= “学费”
3.3.2	收费金额	交缴学费的数量	M	浮点数	amount: float	amount =4500

表 5 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
3.3.3	收费方式	交缴学费的方式	O	字符串	pay_method: integer //参阅 GB/T 15424-1994 电子数 //据交换用支付方式代码	pay_method=01
3.4	奖罚	奖励与处罚的集合	O	集合	award_penalty_list: arraylist(award_penalty_type,50)  award_penalty_type: record ( award: award_type, penalty: penalty_type )	award_penalty_list = ( award= ( content= “三好学生”, date_time= 20001008, ), penalty= ( penalty_cause= “经常旷课”, penalty_level=1, date_time= 20010115 ) )
3.4.1	奖励	记录关于学校对学习者的奖励	O	集合	award_type: record (content: characterstring(iso-10646-1), //SPM:100 //参阅 GB/T 8563-1988 奖励代码 date_time: time(second,10,0) // yyyyymmddThhmmss )	award= ( content= “三好学生”, date_time= 20001008 );



表 5 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
3.4.2	处罚	记录对学习者的违纪处罚	O	集合	<pre> penalty_type: record (penalty_cause:   characterstring(iso-10646-1); //SPM: 100 //参阅 GB/T 8562-1988 纪律处分// 代码 penalty_level:   integer; date_time:   time(second,10,0) // yyyyymmddThhmmss ) </pre>	<pre> penalty= ( penalty_cause= “经常旷课”, penalty_level=1, date_time= 20010115 ) </pre>
3.5	异动	事件和时间的集合	M	集合	<pre> event_log_list: arraylist(event_log_type,200);  event_log_type: record: ( event:   characterstring(iso-10646-1); date_time:   time(second,10,0) ) </pre>	<pre> event_log_list= ( event= “从华师附中升学到 华南理工大学”, date_time= 19980901 ) </pre>
3.5.1	事件	记录学习者与学习经历有关的事情。如,工作变迁、转学、休学、复学、保留学籍。	M	字符串	<pre> event:   characterstring(iso-10646-1) //SPM: 1024 </pre>	<pre> event= “从华师附中升学到华 南理工大学” </pre>
3.5.2	时间	注明相应事件发生的时间	M	时间类型	<pre> date_time:   time(second,10,0) // yyyyymmddThhmmss </pre>	<pre> date_time= 19980901 </pre>
3.6	学业证书	毕业证、结业证或其他证书	O	字符串	<pre> graduate_certificate:   characterstring(iso-10646-1) //SPM:1024 </pre>	
3.7	扩展元	提供了有限度的扩展能力	O	名值对	<pre> administration_bucket:   arraylist(bucket_type,100) // bucket_type: 名值对类型 </pre>	

## 7.4 关系信息

以下信息是学习者关系信息。关系信息可以描述学习者与其他不同性质用户的关系。

表6 关系信息

编号	名称	解释	约束性	类型	类型定义	取值举例
4.	关系信息	学习者与其他用户的关系信息	O	集合	relations_info_type	
4.1	标识	连结信息的内部数据库键	M	标识符	relations_identifier_list : arraylist( learner_identifier_type,200)  // learner_identifier_type: // 标识符类型	relations_identifier_list = ( identifier_type= "pointer", identifier_value= "0x12345678" )
4.2	外标识	连接不同信息库中信息的外部标识符	M	外标识符	relations_hid_list: arraylist(hid_type,200)  // hid_type: 外标识符类型	relations_hid_list = ( identifier_type= "GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符", identifier_value= "22334455" )
4.3	目录项	根据目录项可以将不同性质的关系分类,便于外部检索和定位	O	集合	relations_catalog_entry_list: arraylist(catalog_entry_type,50)  catalog_entry_type: record ( catalog , entry )	relations_catalog_entry_list= ( catalog ="学业关系", entry="同学" )
4.3.1	目录名	目录的名称	O	字符串	catalog: characterstring(iso-10646-1) //SPM: 255	catalog ="学业关系"
4.3.2	项	在目录中目录项的实际值	O	字符串	entry: characterstring(iso-10646-1) //SPM: 255	entry="同学"

表 6 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
4.4	关系	具有某种共性的人的列表	O	关系类型	relationship_list: arraylist(relationship_type,200)  // relationship_type: 关系类型。 //参阅: 表 3	relationship_list = ( others_identifier_list = (identifier_type= "GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符", identifier_value= " 44556677" ), relations_label_list = ( string ="学业关系", locale = "zh" ), relation_to_them = "同学" )
4.5	扩展元	提供了有限度的扩展能力	O	名值对	relations_bucket : arraylist(bucket_type,200)  // bucket_type: 名值对类型	

## 7.5 安全信息

以下信息是学习者安全信息。安全信息是与学习者的安全凭证相关的信息。

表7 安全信息

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
5	安全信息	与学习者相关的安全凭证信息	O	集合	security_info_type	
5.1	标识	连结信息的内部数据库键	M	标识符	security_identifier_list: arraylist( learner_identifier_type,200)  // learner_identifier_type: // 标识符类型	security_identifier_list = ( identifier_type= "pointer", identifier_value= "0x12345678" )

表 7 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
5.2	外标识	连接不同信息库中信息的外部标识符	M	外标识符	security_hid_list: arraylist(hid_type,200)  // hid_type: 外标识符类型	security_hid_list = ( identifier_type= "GB/T 21366-2008 信息 技术 学习、教育和培训 参 与者标识符", identifier_value= "66778899" )
5.3	目录项	根据目录项可以将不同性质的安全凭证分类, 便于外部检索和定位	O	集合	security_catalog_entry_list: arraylist(catalog_entry_type,50),  catalog_entry_type: record ( catalog , entry )	security_catalog_ entry_list= ( catalog ="私钥", entry="生物测度" )
5.3.1	目录名	目录的名称	O	字符串	catalog: characterstring(iso-10646-1) //SPM: 255	catalog ="私钥"
5.3.2	项	在目录中目录项的实际值	O	字符串	entry: characterstring(iso-10646-1) //SPM: 255	entry="生物测度"
5.4	安全凭证	安全凭证列表  注: 安全凭证的含意、名字空间、范围和识别方法必须在实现时进行定义。	O	安全凭证类型	credential_list : arraylist(security_credentia _type, 500)  // security_credentia_type: // 安全凭证类型。 // 参阅: 表 3 //取值例子: 口令、借阅证、银行 //卡、饭卡、考试证、学生证、医 //疗证等。	credential_list = ( ( context_label= "家", credential_type= "口令", credential_value="剑鱼" ), ( context_label="办公室", credential_type ="生物测度" ", credential_value= "120398123b10931203123" )
5.5	扩展元	提供了有限度的扩展能力。	O	名值对	security_bucket : arraylist(bucket_type,300)  // bucket_type: 名值对类型	

## 7.6 偏好信息

以下信息是学习者偏好信息。偏好信息是与学习者的学习设备参数、学习风格、学习模式相关的信息。

表8 偏好信息

编号	名称	解释	约束性	类型	类型定义	取值举例
6.	偏好信息	与学习者学习的风格、模式相关的信息	O	集合	preference_info_type	
6.1	标识	连结信息的内部数据库键	M	标识符	preference_identifier_list : arraylist( learner_identifier_type,200)  // learner_identifier_type: // 标识符类型	preference_identifier_list= ( identifier_type="pointer", identifier_value="0x12345678" ),
6.2	外标识	连接不同信息库中信息的外部标识符	M	外标识符	preference_hid_list : arraylist(hid_type,200)  // hid_type: 外标识符类型	preference_hid_list = ( identifier_type="GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符", identifier_value="99887766" )
6.3	先前偏好集	在当前偏好集前被包括的偏好	O	外标识符	pre_include_preference_hid_list: arraylist(hid_type,100)  // hid_type: 外标识符类型	pre_include_preference_hid_list = ( identifier_type = "GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符", identifier_value = "先自学后上课" )
6.4	后继偏好集	在当前偏好集后被包括的偏好	O	外标识符	post_include_preference_hid_list: arraylist(hid_type,100)  // hid_type: 外标识符类型	post_include_preference_hid_list= ( identifier_type = "GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符", identifier_value = "有声音的课件" )

表 8 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
6.5	设备偏好	对某个 I/O 设备类型的设备设定  注：描述设备偏好，有助于信息技术系统和学习技术系统适应学习者的需要及优化学习。	O	设备类型	device_preference_list : arraylist(device_preference_type,50)  // device_preference_type: // 设备类型。 // 参阅：表 3	device_preference_list= (audio_list = ( output = (context_label = "所有情况", preference_rating = 10, preference_priority = 10000 ) ) )
6.7	兴趣	为选取学习教材提供参考	O	字符串	personal_interests_list: arraylist(personal_interests_type,200) personal_interests_type: character(iso-10646-1): //兴趣类型 //SPM: 50	personal_interests_list="化学"
6.8	扩展元	提供了有限度的扩展能力	O	名值对	preference_bucket : arraylist(bucket_type,100)  // bucket_type: 名值对类型	

## 7.7 绩效信息

以下信息是学习者绩效信息。绩效信息是学习者的学习经历、学习结果等的信息。

表9 绩效信息

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
7.	绩效信息	与学习者的学习成绩和学习效果相关的信息	O	集合	performance_info_type	
7.1	标识	连结信息的内部数据库键	M	标识符	performance_identifier_list: arraylist( learner_identifier_type,200)  // learner_identifier_type: // 标识符类型	performance_identifier_list = ( identifier_type= "pointer", identifier_value= "0x12345678" )

表 9 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
7.2	外标识	连接不同信息库中信息的外部标识符	M	外标识符	performance_hid_list : arraylist(hid_type,200)  // hid_type: 外标识符类型	performance_hid_list = ( identifier_type= “GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符”, identifier_value= “55443322” )
7.3	所有者	学习者标识符。 见 GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符	O	字符串	owner_identifier : characterstring(iso-10646-1) // SPM: 1024	owner_identifier= “webmaster@scut.edu.cn”
7.4	记录日期	绩效记录被记下的日期-时间	M	时间类型	recording_date_time : time(second,10,0) // yyyyymmddThhmmss ( 用 GB/T 7408—2005 表示的时间和日期 )	recording_date_time = 20001122T001122
7.5	生效日期	学习者绩效记录开始有效的日期-时间 注: 绩效记录可能只在一段确定的时间内是有效	O	时间类型	valid_date_time_begin : time(second,10,0) //yyyyymmddThhmmss	valid_date_time_begin=20001123T001122
7.6	失效日期	学习者绩效记录开始无效(过期)的日期-时间。	O	时间类型	valid_date_time_end : time(second,10,0) //yyyyymmddThhmmss	valid_date_time_end= 20031123T001122
7.7	颁布者	学习者绩效记录的发布者标识符	O	字符串	issue_from_identifier : characterstring(iso-10646-1) // SPM: 1024 ( 任课老师、教务员、实验老师、指导老师、主审老师等的标识 )	issue_from_identifier = “registrar@xyz.edu”
7.8	颁布日期	学习者绩效记录被发布的日期-时间。这不同于记录时间(记录存入数据信息库的时间)	O	时间类型	issue_date_time : time(second,10,0) //yyyyymmddThhmmss	issue_date_time = 20001122T001122

表 9 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
7.9	颁布对象	学习者绩效记录的发布对象标识符。可以是某个或一组学生的标识	O	字符串	issue_to_identifier : characterstring(iso-10646-1) // SPM: 1024	issue_to_identifier="华南理工大学电信 2001 硕士 1 班"
7.10	学习体验	与"内容"相关的标识符。注: 这可能是一个 URL, 工具名等。这些名字的标准超出本规范的范围	M	字符串	learning_experience_identifier : characterstring(iso-10646-1) // SPM: 1024	learning_experience_identifier = "五山学院/历史 101"
7.11	学力	与绩效信息有关的学力标识符, 用于刻画或表征学习绩效	O	字符串	competency_identifier: characterstring(iso-10646-1) // SPM: 1024	competency_identifier = "CET-6" //英语六级水平。 //见学力定义规范
7.12	粒度	内容的相对"大小"。数据元素是字符串。其取值不在本标准中说明	O	字符串	granularity : characterstring(iso-10646-1) // SPM: 300 可能取值例子: 入学考试、期中考试、期末考试、平时成绩、学期总评、学年总评、会考、毕业考试、其他考试。	Granularity="期中考试"
7.13	绩效编码方案	分级、编码、度量等的各种方案	M	字符串	performance_coding : characterstring(iso-10646-1) // SPM: 1024 ( 国际标准规定的各种编码方案的名称/编号代码 )	performance_coding = "US-NY-K12-LETTER-GRADE"
7.14	绩效度量	许可值的值制	M	字符串	performance_metric : characterstring(iso-10646-1) // SPM: 300	performance_metric = "A B C D F", (5 分制, 100 分制, 150 分制, 900 分制)
7.15	绩效值	被记录的"成绩"等结果信息	M	字符串	performance_value : characterstring(iso-10646-1) // SPM: 2048	performance_value = "A", 其他可能的值如: "A+", "97", "Pass"。



表 9 (续)

编号	名称	解释	约束性	类型	类型定义	取值举例
7.16	认证信息	与学习者绩效信息相关的数据认证信息	O	数据认证类型	certificate_list : arraylist(data_certification_type,200)	certificate_list = ( certification_source = "certs.org", certification_method =http://certs.org:1234/validate certification_parameter_list = "\$id \$data \$fields" certification_subset = "issue_date_time " + "issue_from_identifier " + "issue_to_identifier " + "learning_experience_identifier " + "performance_coding " + "performance_metric " + "performance_value", certification_identifier = "123abchd1kl2jdljsjew43i5" )
7.17	扩展元	提供了有限度的扩展能力	O	名值对	performance_bucket: arraylist(bucket_type,100)  // bucket_type: 名值对类型	performance_bucket=( name= "time_on_task", value = "1h25m32s" )

## 7.8 作品集信息

以下信息是学习者作品集信息。作品集信息是学习者的代表性作品及相关证明的集合，用于描述和证明学习者能力和成就。

表10 作品集信息

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
8.	作品集信息	与学习者的作品或成就相关的信息	O	集合	portfolio_info_type	
8.1	标识	连结信息的内部数据库键	M	标识符	portfolio_identifier_list: arraylist(learner_identifier_type,200)  // learner_identifier_type: // 标识符类型	portfolio_identifier_list = ( identifier_type= " pointer", identifier_value= "0x12345678" )
8.2	外标识	连接不同信息库中信息的外部标识符	M	外标识符	portfolio_hid_list: arraylist(hid_type,200)  // hid_type: 外标识符类型	portfolio_hid_list = ( identifier_type= "GB/T 21366-2008 信息技术 学习、教育和培训 参与者标识符", identifier_value= "11009988" )
8.3	媒体标识类型	媒体标识的类型	O	字符串	media_id_type: arraylist(MIME_type,200)	media_id_type = "video/mpeg",
8.4	媒体标识	与学习者的成就或作品相关的媒体的标识	O	外标识符	media_id: arraylist(URI_type,200)	media_id = "http://mystuff.org/ my_paintings/ tour.mpeg"
8.5	学习对象元数据引用	与学习者的成就(或作品)相关的学习对象元数据的引用列表	O	标识符	media_lom_list : arraylist(LOM_reference_type, 200)	media_lom_list= (( identifier="GB/T 21365-2008"), (identifier=" LOM ") )
8.6	绩效信息引用	与学习者的成就(或作品)相关的学习者绩效信息记录的引用列表	O	标识符	media_performance_list: arraylist(performance_ reference_type,200)	media_performance_ list = ( identifier_type= " GB/T XXXXX 学力 定义", identifier_value= "55443322" )

表 10 (续)

编号	名称	解 释	约 束 性	类 型	类型定义	取值举例
8.7	学力引用	与学习者的成就（或作品）相关的学力定义的引用列表	O	标识符	media_competency_definition_list: arraylist(competency_definition_reference_type,200)	media_competency_definition_list = "CET-6"  //英语六级水平。 //见学力定义规范
8.8	认证信息	与学习者绩效信息相关的数据认证信息	O	数据认证类型	certificate_list: arraylist(data_certification_type, 200)	certificate_list = ( certification_source = "certs.org", certification_method =http://certs.org:1234/validate, certification_parameter_list = "\$id \$data \$fields", certification_subset = "media_type " + "media_id " + "media_id/* ", certification_identifier="asdk12s782349238s43i5x" )
8.9	扩展元	提供了有限度的扩展能力	O	名值对	portfolio_bucket: arraylist(bucket_type,100)  // bucket_type: 名值对类型	