

网络教育技术规范（CELTS）

学习者模型规范（CELTS-11/V1.0）

——实践指南

（ 试 行 规 范 ）

中国网络教育标准化委员会

目 次

1 概要	1
1.1 范围.....	1
1.2 目的.....	1
2 引用	3
3 定义及缩写	3
4 一致性	6
4.1 一致性级别.....	6
4.1.1 子集	6
4.1.2 严格一致的实现	6
4.1.3 一致的实现	7
4.1.4 非一致的实现	7
4.1.5 数据元素的约束	7
4.1.6 数据元素的生命期	9
4.1.7 约束/生命期的递归/前后关系	10
4.2 一致性标签.....	10
4.3 编码一致性.....	11
4.3.1 数据集一致性	12
4.3.2 数据实例一致性	12
4.4 API 一致性.....	12
4.5 协议一致性.....	13
4.6 数据应用一致性.....	13
4.6.1 严格一致的数据应用	13
4.6.2 一致的数据应用	14
4.6.3 数据应用的多样性	14
4.6.4 数据信息库	14
4.6.5 数据阅读器	15
4.6.6 数据书写器	15
5 功能性	15
5.1 系统功能要求.....	15
5.2 技术要求大纲.....	16
5.2.1 信息的访问控制	16
5.2.2 文化和惯例	16
5.2.3 信息的连通性	17
5.2.4 工程目标概要	17
6 学习者概念模型介绍	18
6.1 CELTS 学习者信息框架	18
6.2 信息类型 与 数据信息库.....	19
6.3 数据访问模型.....	19
6.4 远程, 分布和漫游系统.....	20
6.5 粒度水平的相互关系.....	20
6.6 安全模型.....	21

7 语义	22
7.1 一般的数据操作	22
7.2 具体应用数据操作	23
7.3 数据兼容性	23
7.4 数据类型	23
7.4.1 基本数据类型的定义方法及举例	23
7.4.2 复杂数据类型的定义方法及举例	25
7.4.3 未定义的数据类型	29
8 绑定	29
9 编码	30
10 附录 A: 参考文献	31
11 附录 B: ISO/IEC 11404 数据模型摘要 (信息性)	错误! 未定义书签。
11.1 基本数据类型	32
11.2 CELTS 学习者个人信息数据类型	34
11.3 CELTS 学习者学业信息数据类型	37
11.4 CELTS 学习者管理信息数据类型	38
11.5 CELTS 学习者关系信息数据类型	40
11.6 CELTS 学习者安全信息数据类型	41
11.7 CELTS 学习者偏好信息数据类型	42
11.8 CELTS 学习者绩效信息数据类型	44
11.9 CELTS 学习者作品集信息数据类型	45
12 附录 C: XML 编码绑定(条件性标准)	47
12.1 生成和产生 XML	47
12.2 XML 的解释与消费	51
12.3 基本数据类型的表示方式	52
12.3.1 字符和字符串	52
12.3.2 整数	52
12.3.3 实数	52
12.3.4 日期和时间值	52
12.3.5 空类型	54
12.3.6 字符表达式的编码	54
12.4 例外与扩展处理	54
12.4.1 实现定义行为	54
12.4.2 未指定的行为	54
12.4.3 未定义行为	55
13 附件 D: DNVP 编码绑定 (条件性标准)	56
13.1 带点名值对(DNVPs)	56
13.1.1 基本词汇	56
13.1.2 字段名和字段值	57
13.1.3 新行处理	57
13.1.4 语法摘要	57
13.2 生成和产生带点名值对	58
13.3 带点名值对的读取和解释	59
13.4 基本数据类型的表示法	60

13.4.1 字符和字符串	60
13.4.2 整数	60
13.4.3 实数	60
13.4.4 日期和时间的值	60
13.4.5 空类型	60
13.5 字符表达式的编码.....	60
13.6 例外和扩展处理.....	61
13.6.1 实现定义行为	61
13.6.2 未指定行为	61
13.6.3 未定义行为	61

CELT5-11 学习者模型规范实践指南

本文描述CELT5-11学习者模型规范在实际应用中的相关事项。

1 概要

CELT5-11学习者模型规范是一个关于学习者信息的数据交换规范。该规范中，学习者信息按其逻辑被划分为八类：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。

1.1 范围

本规范指定学习者模型的语法和语义，刻画了学习者(学生或学者)及其知识和能力的特性。本规范涉及到学习过程记录、技巧、能力、学习模式等元素。这些元素可以用各种方式描述，从粗略的摘要到最细微的层次。

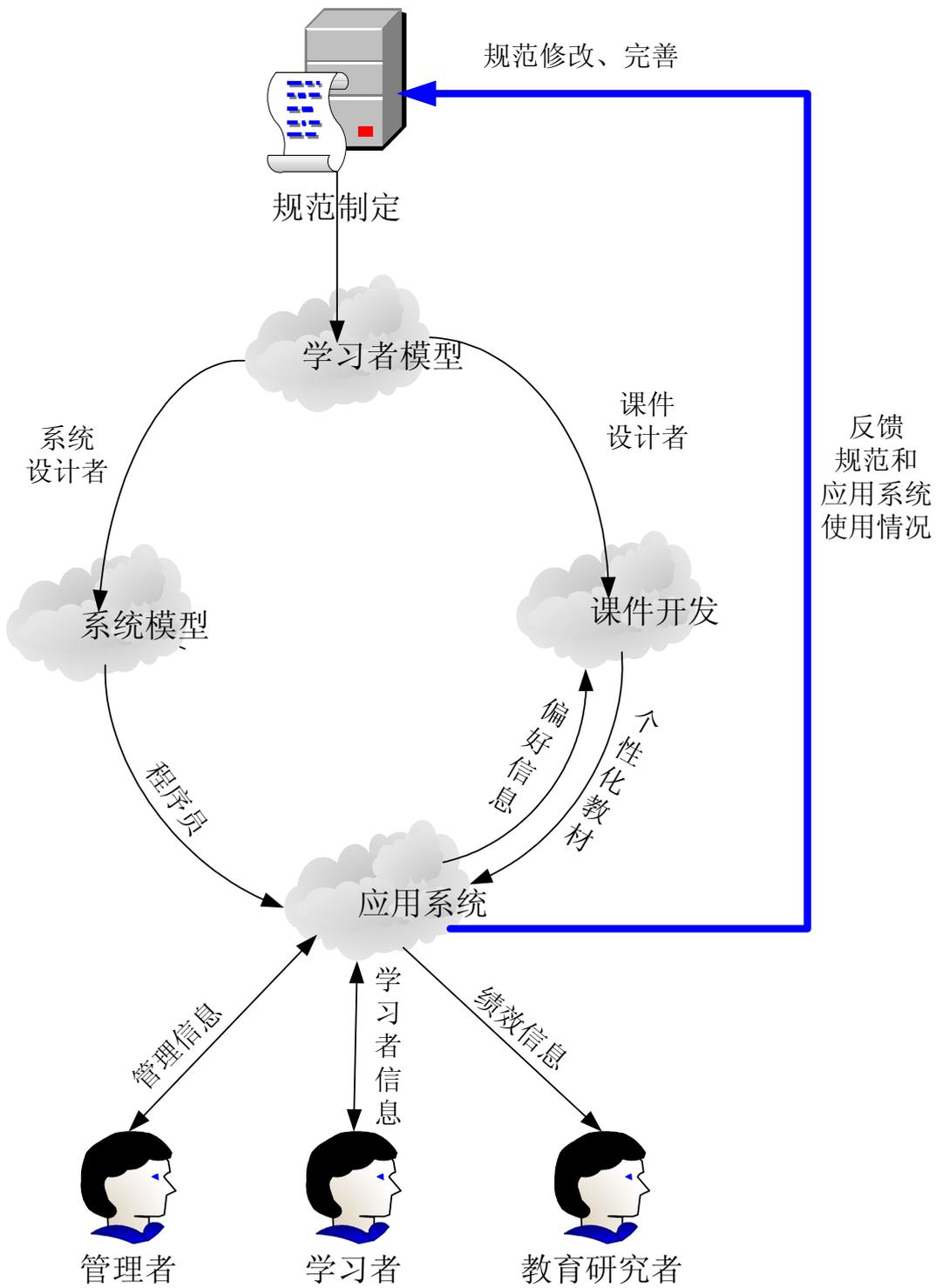
1.2 背景

本规范主要基于IEEE 1484学习技术标准化委员会(LTSC)的基础文档--公开与私人信息(PAPI)的7.0版，同时参考了国际上的同类规范，如美国全球学习联合公司的信息管理系统(IMS)中的学习者信息包(LIP)和高级远程教育(ADL)的共享课件对象参考模型(SCORM v1.0)等。其它相关的规范与标准还有：学校互操作框架(SIF)、美国国家标准机构ANSI TS 130 学生教育记录(Student Educational Record)、互联网 vCard 规范、航空工业CBT委员会(AICC)、HR-XML(人力资源-可扩展标记语言)规范、Internet2 eduPerson等。LTSC是专注教育的公开性正式团体，由各种参与者组成多个工作小组进行研究，并与其它标准组织有合作，如IMS、AICC等。IMS学习者信息包规范部分是源于PAPI(5.0和6.0版)。由于IEEE有广泛的代表性，1484.2学习者模型的成员及草案最近已经并入国际标准组织中的学习技术工作组(ISO/IEC JTC1/ SC36)，该工作组将在PAPI的基础上继续工作，建立信息技术的标准，支持自动化的学习者、教育机构和学习资源系统。

1.3 目的

本规范的目的是：

- 基于本规范能够为任何一个学习者创建一个个人学习者模型，使得应用系统能获取相关教育、学习和工作经历的信息；
- 课件开发者能够开发出更多高质量、个性化的教材；
- 为教育研究者提供规范化的数据；
- 为其他教育规范的发展以及以学生为中心的学习系统的设计提供工作基础；
- 为教育系统设计者提供指导。



2 引用

本规范主要借鉴以下文献：

IEEE 1484.2 “PAPI Learner Model” PAPI学习者模型

CELOTS-1 体系架构与参考模型规范

CELOTS-2 术语规范

3 定义及缩写

表1是本规范中用到的部分术语的中英文对照、术语的解释。

表1 术语表

术语名称	英文对照	解 释
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。
2. 安全边界完整性	Security perimeter integrity	使对内和对外安全威胁均保持在可接受的范围内的安全水平。
3. 安全管理员	Security administrator	安全边界内负责安全管理的人。
4. 安全强度	Security strength	安全程度，描述实现安全边界的特性。
5. 绑定	Binding	从一个框架或规范到另一个框架或规范的应用或映射。
6. 编码	Coding encoding	(A) 在信息交换中信息的形式化或结构化呈现。 (B) 以某种结构呈现信息的过程。
7. CELTS 学习者	CELTS learner	本规范定义的学习者公开与私人信息（人的信息）子集。
8. CELTS 学习者记录	CELTS learner record	本规范定义的学习者信息集合。
9. CELTS 学习者记录引用	CELTS learner record reference	指向 CELTS 学习者记录的标识符。
10. CELTS 学习者信息	CELTS Learner information	本规范中规定的学习者信息及其允许的扩展
11. CELTS 学习者应用	CELTS learner application	运用与本规范一致的 API、编码或协议的信息技术应用。
12. 代理	Agent	能够代表人的行为和动作的自动化系统。
13. 对内安全威胁	Inbound security threat	破坏系统安全和影响安全防线内部信息的外部威胁。
14. 对外安全威胁	Outbound security threat	对信息进行偷窃或未经授权复制，而使信息在安全防线外被截取得或在防线内被删除。
15. 访问	Access	“访问”某个特性是指应用程序读写数据的过程。
16. 访问者	Accessor	试图访问安全防线内资源的用户、系统、代理或实体。
17. 分布式(访问、系统)	Distributed(access、system)	在子系统和其他系统间以互联网或广域网为主要通信方式的技术系统。

术语名称	英文对照	解 释
18. 分解操作	Decomposition Operation	从容器中把信息类型实例分离。
19. 互操作性	Interoperability	两个或多个系统之间交换信息或功能的能力。
20. 解释(数据)	Interpret (data)	依据标准处理数据以得到其语义。
21. 基于角色的访问控制	Role-based access control	基于用户身份和他与其它用户的关系, 对其授权操作或允许访问资源进行验证的安全技术。
22. 聚合操作	Aggregation Operation	连结一个或更多数据类型的几个实例到单个容器中。
23. 聚合(数据类型, 数值)	Aggregate (datatype, value)	一个生成的数据类型或数值, 原则上由其它构件数据类型或数值组成。
24. 漫游(访问、系统)	Nomadic (access、system)	(A) 跨越不同地点或通信时段, 而能保持连续的服务。(B) 部分时间与子系统或相关系统的通讯网络断开。
25. 粒度	Granularity	有关对象的大小、尺度及其详细程度等特征的描述。
26. 名域	Namespace	规定各种标记含义的协议。
27. 人的信息	Human information	一个自然人所有信息的集合。
28. 生成	Generate	把数据按其语义转换成适合数据交换的形式。
29. 实现	Implementation	(A) 抽象的工作实例。(B) 低层次的抽象。(C) 生成实现的过程。
30. 实现定义的行为(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。
31. 数据对象	Data object	在数据访问的概念模型中的数据处理单位。
32. 数据集	Data set	数据结构的实例。
33. 数据结构	Data structure	(A) 零个或多个数据元素聚合体的数据类型;(B) 参见数据集。
34. 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。
35. 数据实例	Data instance	在一些绑定中被使用的数据集。
36. 数据应用	Data application	数据互操作性标准内, 在一致性或概念模型规定应用范围里的信息技术应用。。
37. 体验	Experience	学习者在学习活动中对事物的认识和亲身经历。
38. 信息库	Repository	(A) 信息集合。(B) 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。
39. 信息类型	Information type	在具体应用系统领域中的信息种类。
40. 学习管理系统	Learning management system	一个系统, 用于: (1)确定学习资源进度; (2) 辅助、控制或者指引学习过程; (3)分析和报告学习者成绩。
41. 学习者	Learner	依靠学习技术系统获取知识或技能的个体。
42. 学习者标识	Learner identifier	与学习者相关的标志。 注意 1: 学习者可以拥有多于一个学习者标识符

术语名称	英文对照	解 释
		——多个标识符。单个或多个标识符的方案不在本规范说明。 例如: CELTS-13 学生身份标识符, 护照号码, E-mail 地址, 身份号码, 社会保险号。
43. 学习者绩效粒度	Learner performance granularity	学习者绩效信息的相关尺寸、范围或细节。 注意: 学习者绩效记录可以有不同粒度。 例如: 记录按点数(例如: 学习的秒数); 学习者进度的每分钟数据样本(例如: 学习的分钟数); 课的得分(例如: 学习的日数); 学期级数(例如: 学习的月数); 专业认证(例如: 学习的年数)。
44. 学习者实体	Learner entity	学习者的集合, 例如学习团队。
45. 学习者信息	Learner information	综合学习者或学习者实体的学习技术信息和人的信息。
46. 用户	User	人、其代理人或与信息技术系统进行交互的代理。
47. 语义	Semantics	对应的解释和含义。
48. 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。
49. 最低峰值 (SPM)	Smallest Permitted Maximum	指应用程序至少能支持的列表项数或字符串长度等。

英文缩写及其解释

- **API** Application Programming Interface 应用编程接口
- **CE LTS** Chinese e-Learning Technology Standard 中国网络教育技术标准
- **IEC** International Electrotechnical Commission 国际电工委员会
- **IEEE** Institute of Electrical and Electronic Engineers 电气和电子工程师协会
- **ISO** International Organization for Standardization 国际标准化组织
- **SPM** Smallest Permitted Maximum 最低峰值

4 一致性

在本规范中，“必须”（“shall”）解释为对一个实现的要求；反过来，不可以（“shall not”）表示禁止。不属于要求或被禁止的行为就是未定义的。未定义行为在本规范中以“未定义行为”表示或以行为的具体定义表示。

注意：在实现一致性声明中会说明实现与本规范的哪一项特性一致。

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，一个数据集不显得有“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显现“行为”——并注明“未定义”、“实现定义”和“未指定”。

4.1 一致性级别

以下的小节定义严格一致的实现和一致性实现。在一致性的上下文中，术语“支持”、“使用”、“测试”、“访问”、和“探测”将在4.3小节“编码一致性”、4.4小节“API一致性”、4.5小节“协议一致性”和4.6小节“数据应用一致性”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。本规范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：（1）不是由本规范直接指定的，（2）遵循其它规范的规定，（3）可能在本规范未来的版本中试用。

4.1.1 子集

一个实现必需支持起码一个CELTs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CELTs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4.2小节，一致性标签。

注意：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。

4.1.2 严格一致的实现

严格一致的实现应是以下的一种或几种：严格一致的数据应用、严格一致的API、严格一致的编码或严格一致的协议。

严格一致的实现：

- 必须支持所有必要和可选数据元素；
- 不能使用、测试、访问或探测任何扩展性特性或扩展数据元素；
- 不能超出本规范指定的界限或最低峰值；
- 不能解释和生成依赖于任何未指定、未定义、实现定义或由场所指定的行为的数据元素。

注意：扩展和扩展数据元素的使用是未定义行为。

4.1.3 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；
- 不能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性特性和扩展数据元素；
- 如实现允许，可以超出规定指定的取值范围；
- 可以解释和生成依赖于任何未指定、实现定义或场所指定的行为的数据元素。

注意 1：扩展和扩展数据元素的使用是未定义行为。

注意 2：所有严格一致的实现也都是是一致的实现。

注意 3：如果一个实现，用扩展的方法重新定义规范的特性时，且这些特性改变了严格一致的实现的含义或行为，则该实现与本规范不一致。

4.1.4 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。

4.1.5 数据元素的约束

该小节是信息性表述而不是规范性表述。

数据元素的有 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。

4.1.5.1 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。

4.1.5.2 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阅读器都必须支持所有有效的数据集，相应地，数据信息库和数据阅读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阅读器而言——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，必须包括可选元素。

不支持或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。

4.1.5.3 条件数据元素

条件数据元素是必需的，但是否必需视乎特定条件，如本标准其它地方所定义。每个条件数据元素可以具有一系列的条件。如果条件满足，要使结构有效，数据结构就必须包含这些数据元素。如果每个条件都满足，数据集（和数据实例）必需包含这些元素。与以上可选数据元素同理，所有数据信息库和阅读器都必须支持条件数据元素。同理，数据书写器是必须支持被生成和产生的数据实例中的每个条件元素。

不支持一个或多个条件数据元素的应用是非一致的实现。

包含或支持一个条件数据元素，但包含或支持的方法和本标准不一致，也是非一致的实现。

4.1.5.4 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，例如：（1）实现允许和使用特定扩展数据元素，（2）数据交换双方允许使用特定数据元素，（3）没有使用其它扩展数据元素。

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，例如，可能有必需扩展数据元素，可选扩展数据元素，和条件扩展数据元素。扩展数据元素范围内，这些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（1）数据元素不是使数据结构合法的必要条件，（2）是用于支持扩展数据元素的一致实现。

注意：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，（1）只有支持这种扩展数据元素的实现才可互操作；且（2）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。

一些捆绑“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）。

4.1.6 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。

注意：生命期属性是与约束属性无关的。

4.1.6.1 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。

4.1.6.2 保留数据元素

保留数据元素在标准的这个版本中没有定义。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(1)保留数据元素曾经定义过，(2)被定义为扩展数据元素，且(3)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。

已定义，但将结合未来版本的数据元素是扩展数据元素，（就是说，它们不是保留数据元素）。当扩展性数据元素加入新版本的标准，就会成为必需数据元素、可选数据元素或条件数据元素。

扩展数据元素可在(1)本标准的信息表述性附录中，(2)条件规范性表述的附录中，(3)本标准以外的规范中定义。

扩展数据元素对于本版本并不必需，例如(1)扩展数据元素对于严格一致的系統不是必需的，(2)扩展数据元素对于一致的系統不是必需，(3)扩展数据元素，如果定义了，就不在规范中。

一些捆绑“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。

一致的实现可以使用由实现和数据交换双方允许的扩展元素。详情参阅 4.1.4.4 小节，“扩展数据元素”。

通过定义一个可选数据元素为 ISO/IEC 11404 空类型，可保存数据元素使它不被扩展数据元素覆盖。

4.1.7 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或生命期属性也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（处于其它属性之中）。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。

4.2 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。关联或粘贴一致性标签的方法和技术都不在本规范之内。

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 6.1 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 CELTS 学习者偏好信息(M)和 CELTS 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。注意：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。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。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 CELTS 学习者[子集]数据集：不带绑定；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 CELTS 学习者 [子集]数据集：不带绑定；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 CELTS 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。例如：“celts.xml”文件是一个严格一致的 CELTS 学习者 XML 数据实例。
- 一致的 CELTS 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。例如：“celts.txt”文件是一个一致的 CELTS 学习者 DNVP 数据实例。

- 严格一致的 CELTS 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，例如：服务器 XYZ 是一个严格一致的 CELTS 学习者 XML 编码 / SOAP 协议数据信息库。
- 一致的 CELTS 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，例如：服务器 XYZ 是一个一致的 CELTS 学习者 DNVP 编码 / DCTP 协议数据信息库。
- 严格一致的 CELTS 学习者[子集]绑定数据阅读器：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。例如：导入工具 XYZ 是一个严格一致的 CELTS 学习者 XML 编码 / Java-API 数据阅读器。
- 一致的 CELTS 学习者[子集]绑定数据阅读器：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。例如：导入工具 XYZ 是一个一致的 CELTS 学习者 DNVP 编码 / HTTP 隧道协议数据阅读器。
- 严格一致的 CELTS 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。例如：导出工具 XYZ 是一个严格一致的 CELTS 学习者 DNVP 编码 / JavaScript-API 数据书写器。
- 一致的 CELTS 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。例如：导入工具 XYZ 是一个一致的 CELTS 学习者 XML 编码 / SOAP 协议数据书写器。
- 严格一致的 CELTS 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 例如：XYZ 软件开发包是一个严格一致的 CELTS 学习者 Java API 环境。
- 一致的 CELTS 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。例如：XYZ 软件开发包是一个一致的 CELTS 学习者 JavaScript API 环境。
- 严格一致的 CELTS 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。例如：应用 XYZ 是一个严格一致的 CELTS 学习者 C++-API 应用。
- 一致的 CELTS 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规范外的规范的允许范围内使用扩展数据元素和扩展。例如：应用 XYZ 是一个一致的 CELTS 学习者 Perl 应用。
- 严格一致的 CELTS 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。例如：back office 网关是一个严格一致的 CELTS 学习者 XML 编码/ SOAP 协议。
- 一致的 CELTS 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规范外的规范的允许范围内使用扩展数据元素和扩展。例如：back office 网关是一个一致的 CELTS 学习者 DNVP 编码/ DCTP 协议。

注意：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。

4.3 编码一致性

严格一致的 CELTS 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 CELTS 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。

CELTs 学习者编码必须与第 5 节：“功能性”、第 6 节：“概念模型”和第 7 节：“语义”中的指定数据类型一致。

4.3.1 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(1) 结构上独立于绑定, (2) 严格与本规范的功能性、概念模型和语义一致的, (3) 必需包含所有必需数据元素, (4) 可以包含可选数据元素, (5) 不包含扩展数据元素。

一致的数据集必须是(1) 结构上独立于绑定, (2) 与本规范的功能性、概念模型和语义一致的, (3) 必需包含所有必需数据元素, (4) 可以包含可选数据元素, (5) 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 (1) 以 ISO/IEC 11404 符号表示法 符号描述数据集, (2) 校验是否符合本规范的要求。

4.3.2 数据实例一致性

严格一致的数据实例必须 (1) 是严格一致的数据集, (2) 起码与一种 CELTS 学习者编码严格一致。

一致的数据实例必须 (1) 是一致的数据集, (2) 起码与一种 CELTS 学习者编码一致。

注意 1: 术语“CELTS 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。

注意 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 CELTS 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、数据阅读器、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(“我的学习者信息与 CELTS-11 学习者模型规范一致”)。

4.4 API 一致性

严格一致的 CELTS 学习者 API 必须起码与一种 CELTS 学习者 API 绑定严格一致。

一致的 CELTS 学习者 API 必须起码与一种 CELTS 学习者 API 绑定一致。

CELTS 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

关于 CELTS 学习者模型一致性，以下术语在 API 一致性范围内中定义：一个“支持”特性是一个可被任何 CELTS 学习者 API 的应用使用的特性；一个特性若被 CELTS 学习者 API 的应用读、写和操作，则称其被“使用”；如果一个 CELTS 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 CELTS 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 CELTS 学习者 API 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”。

注意：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 CELTS 实现的一致性评估是由适当的 API 定义和用途确定。

4.5 协议一致性

严格一致的 CELTS 学习者协议必须与至少一个 CELTS 学习者协议绑定严格一致。

一致的 CELTS 学习者协议必须与至少一个 CELTS 学习者协议绑定一致。

CELTS 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，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 CELTS 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 CELTS 学习者协议的应用使用，则称这个特性被“支持”；一个特性如被 CELTS 学习者协议的应用读、写和操作，则称其被“使用”；如果一个 CELTS 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 CELTS 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 CELTS 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”。

注意：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 CELTS 实现的一致性评估是由适当的协议定义和用途确定。

4.6 数据应用一致性

数据应用一致性，是以该应用对本规范的符合程度来衡量。

有两种类型的数据应用一致性：严格一致的和一致性。

4.6.1 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），而且必须与本规范一致。
- 可选特性可以存在（或可用），如果存在（或可用）的话，必须与该规范一致。
- 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。注意：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；

注意：一个严格一致的应用可以是最少一致，对于本规范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）数据应用中没有使用特性探测；（3）数据应用中没有使用扩展特性集合。

4.6.2 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），且应该与本规范一致。
- 可选特性可以存在（或能用）且，如果存在（或可用）的话，应该与本规范一致。
- 扩展特性可以存在（或能用），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。

注意：一致的数据应用可能更有用，但对于本规范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）对实现允许的扩展特性的存在（或可用性）的特性探测或之前一致；（3）在本规范以外指定的扩展特性。

4.6.3 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，数据阅读器和数据书写器

基本原理

应用一致性共有三种不同的场合：数据信息库，数据阅读器和数据书写器。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（数据阅读器：“这是一致的学习者数据阅读器，用于导入数据”；数据书写器：“这是一致的学习者数据书写器本，用于导出数据”，或兼而有之）。

4.6.4 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 CELTS 学习者编码绑定和至少一个 CELTS 学习者 API 或 CELTS 学习者协议绑定严格一致；

注意 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，在严格一致的数据信息库中，数据交换不能保存扩展数据元素，但也不禁止此情况。

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，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 CELTS 学习者编码绑定和至少一个 CELTS 学习者 API 或 CELTS 学习者协议绑定一致。

注意 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。

注意 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。

注意 4：一致的数据信息库可以存取数据集以外的数据对象。

4.6.5 数据阅读器

数据阅读器是一种数据的应用，过程为：(1)消费数据，(2)解释数据，继而产生数据集。

注意 1：理论上，数据书写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。

严格一致的数据阅读器所解释的数据必须与(1)本规范和(2)至少一个本规范的绑定严格一致。

注意 2：严格一致的数据阅读器不解释扩展数据元素。

注意 3：根据本规范的绑定，严格一致的数据阅读器可以“忽略”数据扩展，例如，严格一致的数据阅读器可以消费扩展数据却忽略（不解释）它。

一致的数据阅读器应用解释的数据必须与(1)本规范和(2)至少一个本规范的绑定一致。

注意 4：一致的数据阅读器可解释扩展数据元素。

4.6.6 数据书写器

数据书写器是一种数据应用，包括：(1)从数据集中生成数据并(2)产生数据。

注意 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(1)本规范和(2)至少一个本规范的绑定严格一致。

注意 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(1)本规范和(2)至少一个本规范的绑定一致。

注意 3：一致的数据书写器可生成扩展数据元素。

5 功能性

5.1 系统功能要求

CELTS学习者实现必须(1)有人的信息以供学习技术系统使用；或(2)在CELTS学习者应用间交流该信息。本标准描述了人类信息库的接口和框架，信息库概念上根据各种应用的用途和管理划分。

CELTS 学习者实现可通过以下途径影响数据传输：

- CELTS 学习者应用编程接口 (API)；
- CELTS 学习者协议；
- 或本标准以外的方法。

任何上述方法都可用 CELTS 学习者模型的编码和数据格式来编译数据。

注意：实现不需支持。

5.2 技术要求大纲

该小节是信息性表述而不是规范性表述。

以下是 CELTS 学习者模型标准的技术要求。

5.2.1 信息的访问控制

许多个人信息都进行了编码、存储、排序、筛选及最终发送到计算机程序、学习者、父母、管理员和院校。出于隐私、技术、法规等原因：

- **一些信息被保密。**如：学习者个人信息。
- **一些信息被限制访问。**如，学习者学业信息、学习者绩效信息、学习者学业信息、学习者管理信息、学习者关系信息、学习者作品集信息。
- **一些信息对公众有多种级别的访问权限。**如：学习者管理信息、学习者偏好信息、学习者关系信息、学习者安全信息。
- **一些信息必须对某些应用开放。**如：学习者绩效信息、学习者关系信息、学习者安全信息、学习者偏好信息。
- **一些信息对管理部门开放。**如，学习者个人信息、学习者管理信息、学习者学业信息、学习者绩效信息、学习者关系信息、学习者安全信息。
- **一些信息主要对人有用。**如：学习者作品集信息。

应用有更进一步的限制：

- **应用可以访问的信息太多。**只需访问学习者绩效信息的应用（例如，辅导与培训系统）并不需要访问学习者个人信息，例如家庭电话、父母姓名和种族背景。实际上，一些规章和商业惯例都要求所有记录都必须完全去除标识（没有姓名、院校、年龄等）而仅用雇员号（学号、病历号等）来跟踪过程。例如：一个学员在远程学习中心通过互联网使用课件；课件需要访问学员的历史信息来制定最佳的课程安排，但是这些课件需要知道学员的私人信息吗？就如学员父母的姓名。注意：其它行业例如远程治疗 and 电子商务的应用，可以访问太多的信息，而导致在统一开发的应用中存在很多潜在的业务冲突。（例如，在一个的应用中，如果只有一个合并的信息库存储了学员和病人的偏好，学习技术应用可以访问病人的信息吗？）
- **应用需要在自己的信息库中保存信息。**一些应用选择把信息存储到自己的信息库中。例如，一个学习者在一家院校里购买了远程学习课程，但是该院校要到该学员付清学费才允许纪录的传输。注意：在其它行业，医生、医院和金融机构都希望自己保存特定的某些信息。

即使人的信息存放在分散的信息库里，仍有需要象访问一体化的信息库或数据库。

5.2.2 文化和惯例

CELTS 学习者信息必须支持：

- **文化归属。**某些文化惯例之间可以自动转换（例如，英尺对米，华氏度对摄氏度），而某些可以自动转换却会因时间变化（如，美元对欧元）。有些数据，例如导师或医师用英语写的评语，在某些文化中有用但不能自动转换。CELTS 的工程目标是最大程度国际化（也就是，不同的实现可互操作

而不依赖于文化) 并支持对应各地文化习惯的本地化。例如: 数据和时间以 UTC (Coordinated Universal Time) 格式存储, 但可通过通用库函数 (例如 C 中的 `localtime()`) 转化为本地时间。因此, CELTS 的其中一个工程目标是支持文化习惯, 包括语言。

- **规章惯例。**也许有些数据不能在制度间自动转换, 例如成绩, 数字成绩和字母成绩之间的对应关系是什么? 要以简单、单一的编码方案, 建立跨制度规范的共识是很不现实的。所以, CELTS 的工程目标包括: (1) 支持惯例, 例如, 编码方案“US-NY-K12-LETTER-GRADE”表示纽约公校的字母等级评分制; (2) 支持命名规则减少冲突, 如 XYZ 大学拥有域名“xyz.edu”, 用“xyz.edu”为前缀、后缀; (3) 让机构减少编码方案, 使之得益于更高的互操作性, 同样, 医疗和财务信息也有类似的规章惯例。“让市场解决问题”也是 CELTS 的个工程目标之一。

5.2.3 信息的连通性

应用会以在线或离线的方式访问资源:

- **应用会与断开中心资源的连接。**远程学习并不意味着连续不断的在线访问。例如, 学习者可能仅在每天早上连接以交换信息, 其余时间与互联网和其它集中的资源断开。这些漫游的用户是时断时续。当一个学习者出行, 在线访问显得不可行, 连续的访问又十分昂贵, 此时就需要漫游。

5.2.4 工程目标概要

以下是 CELTS 工程要求和目标的概要:

- 对信息的访问, 要控制在必要的范围内。
- 提高访问数据的性能。
- 支持各种数据和信息结构。
- 支持各种文化习惯。
- 支持各种制度规章。
- 支持各种行业规范。
- 支持各种编码和扩展机制。
- 支持各种信息划分方案。
- 让市场选择编码方案。
- 支持各种类型的连接状态, 包括在线、间或在线和离线。
- 支持改变信息访问的地点 (漫游)。

6 学习者概念模型介绍

本节用图示方法介绍了学习者信息模型的构成，随后介绍了涉及到的数据类型。

6.1 CELTS 学习者信息框架

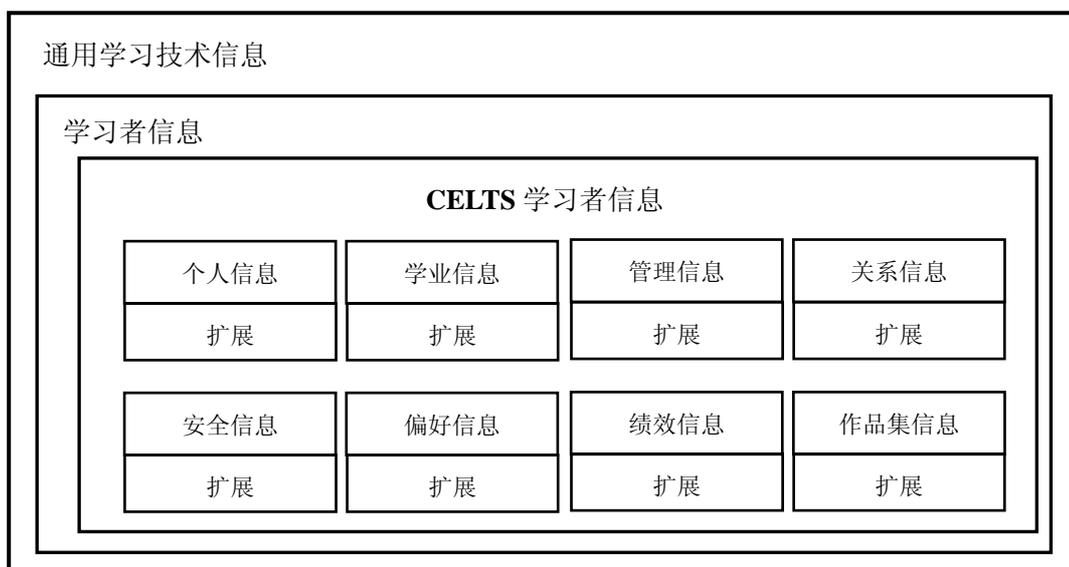


图1 通用学习技术信息、学习者信息和 CELTS 学习者信息之间的关系

从图1可见，学习者信息是通用学习技术信息的一个子集，CELTS学习者信息是学习者信息的一个子集，共包含了八个部分。这八类信息包含了最少的必要信息，并允许应用系统自行定义扩展信息。这八类信息为：

- **个人信息** 主要是学习者的个人信息，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** 是与学习相关的学习者信息。
- **管理信息** 是学习者在学习系统中与管理有关的简要信息。
- **关系信息** 是描述学习者与学习技术系统中其它用户（如教师、代理人或其它学习者）相互关系的信息。
- **安全信息** 是有关学习者安全凭证的信息，例如：密码、询问、回复、私钥、公钥等信息。
- **偏好信息** 是关于学习环境或学习者个人的可促进人机交互或个性化服务的信息。
- **绩效信息** 学习者学习经历、学习结果等相关信息。
- **作品集信息** 是学习者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。

注意 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。

注意 2：本规范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 and 系统性能的要求而进行分散存储。

单字母列表:

- N = CELTS 学习者个人信息, 助记字母是名字 “Name” 的 N。
- L = CELTS 学习者学业信息, 助记字母是学习 “Learning” 的 L。
- A = CELTS 学习者管理信息, 助记字母是管理 “Administration” 的 A。
- R = CELTS 学习者关系信息, 助记字母是关系 “Relations” 的 R,
- S = CELTS 学习者安全信息, 助记字母是安全 “Security” 的 S。
- M = CELTS 学习者偏好信息, 助记字母是我的配置 “My configuration” 的 M。
- G = CELTS 学习者绩效信息, 助记字母是成绩 “Grades” 的 G。
- W = CELTS 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。** 学习信息有不同的隐私和公开级别。本规范很大程度受用户、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 其它用途。** 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它信息, 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, 或由其它生成, 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)。

6.2 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。

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。

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。

6.3 数据访问模型

CELTS 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。

CELTS 学习者数据集的数据交换受以下因素影响:

- **CELTS 学习者编码。** CELTS 学习者数据集可以根据 CELTS 学习者编码绑定进行编码。通过 CELTS 学习者编码规范以外的方法达成共识, 各部分可更方便地进行数据交换。

- **CELT5 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)。
- **CELT5 学习者协议**。协议绑定既是控制传输又是数据传输机制。

这些绑定合起来称为“CELT5 学习者编码、API 和协议”。

注意: CELT5 学习者数据应用可以使用多于一种 CELT5 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实现定义对象。
- **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。数据类型可以预定值域、表达方式、编码、存储、梗概、向其它类型的转换、方法和操作。CELT5 学习者数据元素的数据类型的定义由本规范根据 ISO/IEC 11404 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。注意: 属性(“attributes”)也被称为“特性”(“properties”)。
- **数据信息库访问模型**。CELT5 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 6.8 小节“安全模型”。
- **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 CELT5 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。注意: 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。

6.4 远程, 分布和漫游系统

作为在全局名域内查找、存储和提取信息的基础, 远程特性需要全局名域。名域的规则和分辨方法必须是实现时定义的。

分布的方式有两种, 单一信息库存储一种或多种信息类型, 多个信息库如单一信息库般操作。同步、复制、提交、卷回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。

6.5 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。

例如: 如果一个教师改变了一门课的成绩(较小粒度/密度), 什么时候绩点(较大粒度/密度)会自动变化呢?

6.6 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。
- 对内安全威胁的属性、类型和可接受程度。
- 对外安全威胁的属性、类型和可接受程度。
- 安全强度。
- 安全特性的参数设置、建立、协商和拆除。
- 安全边界一体化的管理。

互操作性约定在本规范以外定义。

注意：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 CELTS 学习者绑定中定义。
- **安全扩展模型。**无法预知会否使用附加的安全特性。安全扩展的合并方法在 CELTS 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。如果有其它方法，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。注意：CELTS-13，学生身份标识，定义了与学习者标识相关的数据类型。
- **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。

- **证书的确认证。**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规范”协调。

7 语义

无论选择哪一个CELTs学习者绑定和CELTs学习者编码，CELTs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：CELTs学习者编码绑定定义了本节中每个数据类型的映射。

注3：CELTs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：CELTs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。

7.1 一般的数据操作

CELTs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。
- **删除操作。**把一个数据类型的实例从存储器中丢弃。（注意：比较在应用存储器中，暂存中和数据库中丢弃一个记录。）
- **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。例如：隐式标签转换会新建到新标签的连接而删除到旧标签的连接；显式标签转换会更改信息目录的标签。
- **标签操作。**创建（或删除）与信息实例相关的名字，名字由“主叫方”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。（注意：标签和引用的差别在于：“被叫方”名字被引用，“主叫方”以名字作标签）
- **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。**连结一个或多个数据类型的几个实例到一个容器中。
- **分解操作。**从一个容器中提取出数据类型的实例。

CELTs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。

7.2 具体应用数据操作

CE LTS 学习者信息必须支持以下对数据集的具体应用数据操作。

- **累积操作。**CE LTS 学习者记录可被累积、聚集或分析。例如：“三个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。**CE LTS 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：期中成绩和期末成绩累计成最终成绩，经过压缩后，只剩最终成绩，数据集缩简了。
- **排序操作。**对学习记录可根据标准进行排序。例如：可按用户名字的字母顺序排序。

CE LTS 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。

7.3 数据兼容性

CE LTS 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 (排序)}。
- **文本格式化。**数值转换为文本表达或文本表达转换为数值。

CE LTS 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。

7.4 数据类型

本节用数据类型举例解释本规范定义的信息。“最低峰值” (SPM) 是元素最大取值的下限，规定了 SPM 的元素，应用不能取小于 SPM 的最大值。

7.4.1 基本数据类型的定义方法及举例

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2 基本数据类型及其定义方法

编号	类型名称	解 释	类型定义 (ISO/IEC 11404)
1.	字符串	以两个字节为单位	<code>characterstring(iso-10646-1)</code>
2.	字节串	以一个字节为单位	<code>octetstring</code>
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。	<code>context_label_type = characterstring(iso-10646-1) // SPM: 500</code>
4.	数组列表	多个数组元素的简写。	<code>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</code>

编号	类型名称	解 释	类型定义 (ISO/IEC 11404)
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。	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。 • bucket_name: 名字-值对的名字部分。 • 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。其取值不在本规范中说明。	identifier_type_type = octetstring // SPM: 256
9.	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。 • context_label: 引用这种数据元素的情境。 • identifier_type: 标识符的类型。 • 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)
10.	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)
11.	数据认证	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。 • source: 谁认证这条记录。 • method: 用什么方法认证。	data_certification_type = record (certification_source : octetstring, // SPM: 2048)

编号	类型名称	解 释	类型定义 (ISO/IEC 11404)
		<ul style="list-style-type: none"> • parameter_list: 使认证有效的必要选项和参数。 注: 有效的 source, method 和 parameter_list 是 identifier 有效的必要条件。 • subset: 在这个记录中已被认证的元素列表。 • identifier: 传送给认证者的标识符, 例如一个证书 ID。 	certification_method : octetstring , // SPM: 1024 certification_parameter_list : octetstring , // SPM: 16384 certification_subset : octetstring , // SPM: 1024 certification_identifier : octetstring , // SPM: 2048 certification_bucket : arraylist(bucket_type,100))

7.4.2 复杂数据类型的定义方法及举例

表3定义了各类学习者信息所引用的复杂数据类型。

表3 复杂类型的定义及举例

编号	类型名称	解 释	类型定义(ISO/IEC 11404)
1.	名字类型	<ul style="list-style-type: none"> • official_name: 学习者的合法名字。 如: 学习者护照上的名字。 • full_informal_name: 在非正式场合中所用的非正式全名。 <p>注: 在个人信息的“名字”中引用该类型。</p>	name_type= record (context_label : context_label_type , official_name : arraylist(formal_name_type,5) , full_informal_name : arraylist(full_name_type,5) ,)
2.	正式名字类型	姓/名 primary/secondary 注: 在‘名字类型’中引用该类型。	formal_name_type = record (primary : characterstring(iso-10646-1) , // SPM: 70 secondary : characterstring(iso-10646-1) // SPM: 70)
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

编号	类型名称	解 释	类型定义(ISO/IEC 11404)
4.	电话类型	<ul style="list-style-type: none"> • context_label : 适用范围。 • identifier_type: 电话号码的类型。 • phone_number: 电话号码。 <p>注: 在个人信息的“电话”中引用该类型。</p>	<pre>telephone_type = record (context_label : context_label_type, identifier_type : identifier_type_type, phone_number : octetstring // SPM: 50)</pre>
5.	电子邮件类型	<p>学习者的联络电子邮件, 包括:</p> <ul style="list-style-type: none"> • email_address_type: 电子邮件地址的类型。 • email_address: 电子邮件地址。 <p>注: 在个人信息的“电子邮件”中引用该类型。</p>	<pre>email_contact_type = record (context_label : context_label_type, email_address_type : identifier_type_type, email_address : octetstring // SPM: 255)</pre>
6.	邮寄地址类型	<ul style="list-style-type: none"> • 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 • organization_*: 收件单位, 包括: name : 名称 activity : 部门 division : 分支 • addressee_*: 收件人, 包括: care_of_address : 代理 suffix : 称谓 name_family : 姓 name_given : 名 title : 称号 	<pre>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 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)</pre>

编号	类型名称	解释	类型定义(ISO/IEC 11404)
		<p>function : 职务</p> <p>occupation : 职位</p> <p>注: 在个人信息的“邮寄地址”中引用该类型。</p> <p>注: 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。</p>	<p>delivery_supplementary_address :</p> <p>characterstring(iso-10646-1), // SPM: 70</p> <p>organization_name :</p> <p>characterstring(iso-10646-1), // SPM: 70</p> <p>organization_activity :</p> <p>characterstring(iso-10646-1), // SPM: 70</p> <p>organization_division :</p> <p>characterstring(iso-10646-1), // SPM: 70</p> <p>addressee_care_of_address :</p> <p>characterstring(iso-10646-1), // SPM: 70</p> <p>addressee_occupation :</p> <p>characterstring(iso-10646-1), // SPM: 70</p> <p>addressee_function :</p> <p>characterstring(iso-10646-1), // SPM: 70</p> <p>addressee_title :</p> <p>characterstring(iso-10646-1), // SPM: 35</p> <p>addressee_name_family :</p> <p>characterstring(iso-10646-1), // SPM: 70</p> <p>addressee_name_given :</p> <p>characterstring(iso-10646-1), // SPM: 70</p> <p>addressee_name_suffix :</p> <p>characterstring(iso-10646-1), // SPM: 35</p> <p>)</p>
7.	关系类型	<ul style="list-style-type: none"> • others_identifier_list: 与学习者相关的其它用户的标识符。 • relations_label_list: 描述关系属性的标签。 • relation_to_them_list: 学习者和相关用户的关系。 • relation_to_me_list: 相关用户和学习者的关系。 <p>注: 在关系信息的“关系”中引用该类型。</p>	<p>relationship_type = record</p> <p>(others_identifier_list :</p> <p>arraylist(identifier_type,200),</p> <p>relations_label_list :</p> <p>arraylist(mstring_type,200),</p> <p>relation_to_them_list :</p> <p>arraylist(relationship_type_type,200),</p> <p>relation_to_me_list :</p> <p>arraylist(relationship_type_type,200)</p> <p>)</p>
8.	关系类型的类型	<ul style="list-style-type: none"> • 同学: 与学习者分享学习体验的其他学习者。 • 属于: 学习者与所在团体的关系。 • 包含: 与学习者有联系的团体。 • 教师: 学习者的老师。 • 教员: 参与管理学习者的人员, 包括 	<p>relationship_type_type = enumerated</p> <p>("同学",</p> <p>"学生",</p> <p>"教师",</p> <p>"教员",</p> <p>)</p>

编号	类型名称	解释	类型定义(ISO/IEC 11404)
		教务人员、指导员等。 <ul style="list-style-type: none"> • 学生: 该教师的学生。 注: 在‘关系类型’中引用该类型。	"属于", "包含")
9.	安全凭证类型	<ul style="list-style-type: none"> • context_label: 引用这种数据元素的情境。 • credential_type: 凭证的类型。 • credential_value: 凭证的值。 注: 在安全信息的“安全凭证”中引用该类型。	<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	<ul style="list-style-type: none"> • security: 加密; • text: 文本; • speech: 语音; • graphic: 图象; • audio: 音频; • video: 视频; • tactile: 触觉; • session: 会话; • other: 其它未指定的设备。 注: 在偏好信息的“设备偏好”中引用该类型。	<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), 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	<ul style="list-style-type: none"> • input: 相应的输入设备。 • output: 相应的输出设备。 注: 在‘设备类型’中引用该类型。	<pre> device_io_preference = (input : device_parameter_type, output : device_parameter_type) </pre>
12.	设备参	<ul style="list-style-type: none"> • name: 国际统一的设备名称。 	<pre> device_parameter_type = </pre>

编号	类型名称	解释	类型定义(ISO/IEC 11404)
	数类型	<ul style="list-style-type: none"> • rating: 设备的有用程度。 • priority: 和相似设备相比的优先级（或重要性）。 • device_name: 机器可识别的设备名字。 • device_type: 机器可识别的设备类型描述，其值不在本规范规定。 • method: 访问的方法，其值不在本规范规定。 • protocol: 用于访问设备的协议，其值不在本规范规定。 • coding: 所用的编码技术，其值不在本规范规定。 • encoding: 所用的译码技术，其值不在本规范规定。 • other: 其它信息。 <p>注：在‘设备输入输出类型’中引用该类型。</p>	<pre>(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>

7.4.3 未定义的数据类型

以下是本规范未定义的，但引用的数据类型，这些类型可以在其它标准或规范中找到。

- 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 CELTS-3 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。

8 绑定

CELTS 学习者信息与编码、API 和协议绑捆起来。

注意 1: 所有 CELTS 学习者绑定(编码、API、协议等)是“条件规范性的”：

- 实现一致性声明描述遵守哪些规范特性。
- 按本规范的定义，实现一致性声明说明所支持的环境。
- 对于实现，标准的相应部分是规范性的。

例如：声称满足 XML 编码绑定的实现，必须与“CELTS 学习者 XML 编码绑定”一致，而无须满足“DNVP 编码绑定”。如果实现声称与 XML 和 DNVP 编码绑定都一致，则必须两者都满足。

注意 2: 本规范现在或将来的版本将支持各种编码绑定(XML, ASN.1, C, LISP), API 绑定(C, C++, Java, ECMAscript, Perl, Tcl, Visual Basic, LISP), 和协议绑定(HTTP tunneling, DCTP, CORBA)

9 编码

按相应的规定, CELTS学习者信息被编码为各种数据格式和通讯层。

注1: 所有CELTS 学习者编码被他们各自的CELTS 学习者绑定所定义(编码, API, 协议等)。

注2: 所有CELTS 学习者绑定是“有条件地标准化的”。参阅第8节 绑定。

注3: 本规范的现在或将来的版本将支持各种应用环境(Unix, Linux, Win32), 数据格式(ASCII, ISO/IEC 646, ISO/IEC 8859-1, ISO/IEC 10646-1)和通讯层(HTTP, HTTPS, IIOP, SSL, FTP, TCP, IP)。

10 附录 A: 参考文献

- ISO JTC1 CAW (Cultural Adaptability Workshop)——文化适应性工作部
- ISO/IEC JTC1 SC22 WG11——与语言无关的绑定标准
- ISO/IEC JTC1 SC22 WG20——国际化标准
- ISO/IEC JTC1 SC32 WG2——元数据标准
- ISO/IEC JTC1 SC35——用户界面标准
- ISO/IEC JTC1 SC36 ——学习、教育、培训的信息技术标准
- GB 2261-1980——人的性别代码
- GB/T 2695-2000 和 ISO3166——世界各国和地区名称代码
- 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 4880-1991 和 ISO-639——语种名称代码
- GB 4767-1984——健康状况代码
- GB/T 8562-1988——纪律处分代码
- GB/T 8563-1988——奖励代码
- GB/T 16835-1997——高等学校本科、专科专业名称代码
- GB/T 15424-1994——电子数据交换用支付方式代码
- NCITS L8——美国国家信息技术标准委员会元数据标准
- IMS——美国全球联合学习公司的标准
- *** 有待提供 ***
- *** 有待更新 ***

11 附录 B: ISO/IEC 11404 数据模型摘要 (信息性)

本附录是信息性而不是规范性的。

以下信息是本标准中数据类型定义的摘要。

注 1: ISO/IEC 11404 与许多结构化编程语言相似。每个数据元素以“标识: 数据类型” (“**identifier : datatype**”) 作声明。数据类型可以是 ISO/IEC 11404 已有的数据类型, 或是生成的数据类型(如, 记录 “**record**”、数组 “**array**”), 或者创建的数据类型(如, “**type x = octetstring**” 创建了类型 “**x**”, 接着 “**y : x**” 声明 **y** 为实际类型 “**octetstring**”)。声明之间以逗号分开, 括号组合, 而数据类型是可以嵌套的。

11.1 基本数据类型

在 ISO/IEC 11404 中, 以下是基本数据类型的摘要。

```
////////////////////////////////////
//// 其它数据类型使用的基本数据类型
////////////////////////////////////

// 与数据元素相关的情境标签
// 例: "!(japan)" 意味着 "不在日本范围之内"
// 其意思由实现定义
type context_label_type =
    characterstring(iso-10646-1), // SPM: 500

//多个数组元素的简写
type arraylist(type_spec,size) =
    array (0..size-1) of (type_spec),

//多语言字符串
// 字符串:本地字符串
// 映射:对应的地区,也作 "地域"("locale").
type mlstring_type =
record
(
    string:
        characterstring(iso-10646-1), // SPM: 1000
    locale:
        characterstring(iso-646), // SPM: 255
),

//由多语言字符串组成的数组
type mlstring_array_type(limit) =
    array (0..limit-1) of (mlstring_type),
```

// CELTS 学习者信息类型中增加的名值对,该属性使信息获得有限的扩展能力,

```
type bucket_type =  
record  
(  
    bucket_name:  
        characterstring(iso-10646-1), // SPM: 200  
    bucket_value:  
        octetstring, // SPM: 4096  
) ,
```

// 用于连接数据库的内标识

```
type identifier_type =  
record  
(  
    context_label :  
        context_label_type,  
    identifier_type :  
        identifier_type_type,  
    identifier_value :  
        octetstring, // // SPM: 1024  
) ,
```

//使不同信息库间 CELTS 学习者信息关联的外标识符

```
type hid_type =  
record  
(  
    context_label :  
        context_label_type,  
    identifier_type :  
        identifier_type_type,  
    identifier_value :  
        octetstring, // // SPM: 1024  
) ,
```

// 标识符类型

```
type identifier_type_type =  
    octetstring, // SPM: 256
```

// 数据认证信息类型

```
type data_certification_type =  
record  
(  
    certification_source : //谁认证这条记录
```

```

        octetstring, // SPM: 2048
certification_method : //用什么方法认证
        octetstring, // SPM: 1024
certification_parameter_list :
        octetstring, // SPM: 16384
certification_subset : //在这个记录中已被认证的元素列表
        octetstring, // SPM: 1024
certification_identifier : //相关的标识符
        octetstring, // SPM: 2048
certification_bucket : // other information
        arraylist(bucket_type,100),
),

```

11.2 CELTS 学习者个人信息数据类型

在 ISO/IEC 11404 中，以下是 CELTS 学习者个人信息数据类型的摘要。

```

////////////////////////////////////
//// CELTS 学习者个人信息
////////////////////////////////////

// CELTS 学习者个人信息类型
type personal_info_type =
record
(
    personal_identifier_list : //数据库连接(键)
        arraylist(learner_identifier_type,200),
    personal_hid_list : // HID 连接 (信息库间)
        arraylist(learner_hid_type,200),
    name_list :
        arraylist(name_type,40),
    telephone_list :
        arraylist(telephone_type,15),
    email_contact_list :
        arraylist(email_address_type,25),
    postal_address_list :
        arraylist(postal_address_type,10),
    demography :
        (nationality:
            characterstring(iso-10646-1), //SPM:50
        folk:
            characterstring(iso-10646-1), //SPM:50
        native_place:
            characterstring(iso-10646-1), //SPM:100

```

```

        birth_place:
            characterstring(iso-10646-1), //SPM:100
        birth_date:
            time(day,10,0),
        gender:
            integer,
        id_number:
            characterstring(iso-10646-1), //SPM:64
        politic_status:
            characterstring(iso-10646-1),
        language:
            characterstring(iso-10646-1), //SPM:32
        health:
            characterstring(iso-10646-1),
    )
    personal_bucket :
        arraylist(bucket_type,100),
),

// CELTS 学习者名字。不同应用中需要不同的名字。
type name_type =
record
(
    context_label : // 例, "!(japan)"
        context_label_type,
    official_name :
        arraylist(formal_name_type,5),
    full_informal_name :
        arraylist(full_name_type,5),
}

//正式名字分为姓和名(primary/secondary)在 ICAO(国际民间航空组织),
//IATA(国际航空运输协会), UN(联合国), 和 ATA (航空运输协会)
//标准中已经被定义, 其目的是实现护照信息的自动化。
type formal_name_type =
(
    primary : // 姓
        characterstring(iso-10646-1), // SPM: 70
    secondary : // 名
        characterstring(iso-10646-1), // SPM: 70
),

type full_name_type =
    characterstring(iso-10646-1), // SPM: 140

```

```

// 电话号码
type telephone_type =
record
(
    context_label :
        context_label_type,
    identifier_type :
        identifier_type_type,
    phone_number :
        octetstring, // SPM: 50
),

// 电子邮箱地址
type email_contact_type =
record
(
    context_label :
        context_label_type,
    email_address_type :
        identifier_type_type,
    email_address :
        octetstring, // SPM: 255
),

// 邮寄地址以 ISO 11180, "邮寄地址编制"为准
type postal_address_type =
record
( //注: 所有元素都是可选;大小为最低峰值
    context_label :
        context_label_type,

// 收件人 注: 实现对应学习者名字的内容填写收件人的名字,称谓,等
    addressee_title :
        characterstring(iso-10646-1), // SPM: 35
    addressee_name_given :
        characterstring(iso-10646-1), // SPM: 70
    addressee_name_family :
        characterstring(iso-10646-1), // SPM: 70
    addressee_name_suffix :
        characterstring(iso-10646-1), // SPM: 35
    addressee_occupation :
        characterstring(iso-10646-1), // SPM: 70
    addressee_function :

```

```

        characterstring(iso-10646-1), // SPM: 70
addressee_care_of_address :
        characterstring(iso-10646-1), // SPM: 70

// 单位
organization_name :
        characterstring(iso-10646-1), // SPM: 70
organization_activity :
        characterstring(iso-10646-1), // SPM: 70
organization_division :
        characterstring(iso-10646-1), // SPM: 70

// 邮寄地址
delivery_street_type :
        characterstring(iso-10646-1), // SPM: 35
delivery_street_name :
        characterstring(iso-10646-1), // SPM: 70
delivery_street_id_number :
        characterstring(iso-10646-1), // SPM: 20
delivery_supplementary_address :
        characterstring(iso-10646-1), // SPM: 70
delivery_city :
        characterstring(iso-10646-1), // SPM: 35
delivery_po_box :
        characterstring(iso-10646-1), // SPM: 20
delivery_postcode :
        characterstring(iso-10646-1), // SPM: 20
delivery_routing :
        characterstring(iso-10646-1), // SPM: 20
delivery_office :
        characterstring(iso-10646-1), // SPM: 35
delivery_territory :
        characterstring(iso-10646-1), // SPM: 35
delivery_country :
        characterstring(iso-10646-1), // SPM: 35
),

```

11.3 CELTS 学习者学业信息数据类型

在 ISO/IEC 11404 中，以下是 CELTS 学习者学业信息数据类型的摘要。

```

////////////////////////////////////
//// CELTS 学习者学业信息
////////////////////////////////////

```

```

// CELTS 学习者学业信息类型
type learning_info_type =
record
(
  learning_identifier_list : //数据库连接(键)
    arraylist(identifier_type,200),
  learning_hid_list : // HID 连接 (信息库间)
    arraylist(hid_type,200),
  institute:
    characterstring(iso-10646-1) //SPM:64
  major:
    characterstring(iso-10646-1) //SPM:64
  class:
    characterstring(iso-10646-1) //SPM:64
  learning_scheme:
    characterstring(iso-10646-1) //SPM:1024
  credit:
    characterstring(iso-10646-1) //SPM: 256
  learning_bucket:
    arraylist(bucket_type,100)

```

11.4 CELTS 学习者管理信息数据类型

在 ISO/IEC 11404 中，以下是 CELTS 学习者管理信息数据类型的摘要。

```

////////////////////////////////////
//// CELTS 学习者管理信息
////////////////////////////////////

// CELTS 学习者管理信息类型
type administration_info_type =
record
(
  administration_identifier_list : //数据库连接(键)
    arraylist(identifier_type,200),
  administration_hid_list : // HID 连接 (信息库间)
    arraylist(hid_type,200),
  fares_list:
    arraylist(fares_type,200),
  award_penalty:
    arraylist(award_penalty_type,50),
  event_log:

```

```

        arraylist(event_log_type,500),
graduate_certificate:
    characterstring(iso-10646-1)    //SPM:50
administration_bucket:
    arraylist(bucket_type,100),
),

//管理信息中数据类型
type fares_type=
    (fee:
        characterstring(iso-10646-1),
    pay_method:
        integer
    ),

type award_penalty_type=
record
    ( award:
        award_type,
    penalty:
        penalty_type,
    ),

type award_type=
    (content:
        characterstring(iso-10646-1),    //SPM:100;
    date_time:
        time(second,10,0)    //yyyymmdd
    ),

type penalty_type=
    (penalty_cause:
        characterstring(iso-10646-1),    //SPM:100;
    penalty_level:
        integer,
    date_time:
        time(second,10,0)    //yyyymmdd
    ),

type event_log_type=
record:
    ( event:
        characterstring(iso-10646-1),    //SPM:100
    date_time:

```

```
        time(second,10,0),      //yyyymmdd
    ),
```

11.5 CELTS 学习者关系信息数据类型

在 ISO/IEC 11404 符号表示法中，以下是 CELTS 学习者关系信息数据类型的摘要。

```
////////////////////////////////////
//// CELTS 学习者关系信息
////////////////////////////////////

// CELTS 学习者关系信息类型
type relations_info_type =
record
(
    my_relations_identifiser_list : //数据库连接(键)
        arraylist(identifiser_type,200),
    relations_hid_list : // HID 连接 (信息库间)
        arraylist(hid_type,200),
    relations_catalog_entry_list: //关系类型
        arraylist(catalog_entry_type,50)
    relationship_list : // 个人关系列表
        arraylist(relationship_type,200),
    relations_bucket :
        arraylist(bucket_type,200),
),

type catalog_entry_type=
( catalog:
        characterstring(iso-10646-1),
    entry:
        characterstring(iso-10646-1),
),

// CELTS 学习者关系类型
type relationship_type =
record
( //
    others_identifiser_list : // 相关者的标识
        arraylist(identifiser_type,200),
    relations_label_list : // 关系名称
        arraylist(mlstring_type,200),
    relation_to_them_list : // 学习者对于他们的关系名称
        arraylist(relationship_type_type,200),
```

```

        relation_to_me_list : // 他们对于学习者的关系名称
            arraylist(relationship_type_type,200),
    ),

// 关系类型的数据类型
type relationship_type_type =
enumerated
(
    "同学", // 同学
    "教师", // 学习者的老师
    "教员", // 管理学习者的人员
    "学生", // 教师的学生
    "属于", // 学生所在的团体
    "包含", // 学生与团体的关系
),

```

11.6 CELTS 学习者安全信息数据类型

在 ISO/IEC 11404 符号表示法中，以下是 CELTS 学习者安全信息数据类型的摘要。

```

////////////////////////////////////
//// CELTS 学习者安全信息
////////////////////////////////////

// CELTS 学习者安全信息类型
type security_info_type =
record
(
    my_security_identifier_list : //数据库连接(键)
        arraylist(identifier_type,200),
    security_hid_list : // HID 连接 (信息库间)
        arraylist(hid_type,200),
    security_catalog_entry_list: //
        arraylist(catalog_entry_type,50),
    credential_list : // 安全凭证列表
        arraylist(security_credential_type,500),
    security_bucket :
        arraylist(bucket_type,300),
),

//安全凭证类型
type security_credential_type =
record
(

```

```

context_label :
    context_label_type,
credential_type :
    identifier_type_type,
credential_value :
    octetstring, // SPM: 8192
),

type catalog_entry_type=
( catalog:
    characterstring(iso-10646-1)
entry:
    characterstring(iso-10646-1)
),

```

11.7 CELTS 学习者偏好信息数据类型

在 ISO/IEC 11404 符号表示法中，以下是 CELTS 学习者偏好信息数据类型的摘要。

```

////////////////////////////////////
//// CELTS 学习者偏好信息
////////////////////////////////////

// CELTS 学习者偏好信息类型
type preference_info_type =
record
(
    preference_identifier_list : //数据库连接(键)
        arraylist(identifier_type,200),
    preference_hid_list : // HID 连接 (信息库间)
        arraylist(hid_type,200),
    pre_include_preference_hid_list : // include before this list
        arraylist(hid_type,100),
    post_include_preference_hid_list : // include after this list
        arraylist(hid_type,100),
    device_preference_list : // Device preferences
        arraylist(device_preference_type,50),
    personal_interest_list: //
        arraylist(personal_interest_type,200),
    preference_bucket :
        arraylist(bucket_type,100),
),

// 兴趣类型

```

```

type personal_interest_type=
( character(iso-10646-1), // SPM:50
),
// CELTS 学习者设备偏好选择
type device_preference_type =
record
( //注：所有元素都是可选；大小为最低峰值
security_list : // 安全设备
    arraylist(device_io_preference_type,20),
text_list : //文本设备
    arraylist(device_io_preference_type,20),
speech_list : //语音设备
    arraylist(device_io_preference_type,20),
graphics_list : //图象设备
    arraylist(device_io_preference_type,20),
audio_list : //音频设备
    arraylist(device_io_preference_type,20),
video_list : //视频设备
    arraylist(device_io_preference_type,20),
tactile_list : //触摸设备
    arraylist(device_io_preference_type,20),
session_choosing : //会话选择设备
    arraylist(device_io_preference_type,20),
other : //其它未指定的设备
    arraylist(device_io_preference_type,100),
),

//输入输出设备偏好
type device_io_preference =
(
input :
    device_parameter_type,
output :
    device_parameter_type,
),

// 单个设备的一套偏好选择
type device_parameter_type =
(
context : // 适用范围
    context_label_type,
preference_name : //国际统一的设备名称
    characterstring(iso-10646-1),
preference_rating : //设备的有用程度

```

```

        integer,
    preference_priority : //和类似设备相比的优先度
        integer,
    device_name : //机器可识别的设备名字
        characterstring(iso-10646-1),
    device_type : //机器可识别的设备类型描述, 其值不在本规范规定
        characterstring(iso-10646-1),
    method : //访问的方法, 其值不在本规范规定
        characterstring(iso-10646-1),
    protocol : //用于访问设备的协议范围, 其值不在本规范规定
        characterstring(iso-10646-1),
    coding : //访问所用的编码, 其值不在本规范规定
        characterstring(iso-10646-1),
    encoding : //访问所用的编码, 其值不在本规范规定
        characterstring(iso-10646-1),
    other : //其它信息
        characterstring(iso-10646-1),
),

```

11.8 CELTS 学习者绩效信息数据类型

在 ISO/IEC 11404 中, 以下是 CELTS 学习者绩效信息数据类型的摘要。

```

////////////////////////////////////
//// CELTS 学习者绩效信息
////////////////////////////////////

// CELTS 学习者绩效信息类型
type performance_info_type =
record
(
    performance_identifier_list : //数据库连接(键)
        arraylist(learner_identifier_type,200),
    performance_hid_list : // HID 连接 (信息库间)
        arraylist(hid_type,200),
    owner_identifier : //记录的所有者,不一定是学习者
        characterstring(iso-10646-1), // SPM: 1024
    recording_date_time : //记录时间
        time(second,10,0), // yyyyymmddThhmmss
    valid_date_time_begin : //生效时间
        time(second,10,0), // yyyyymmddThhmmss
    valid_date_time_end : //失效时间
        time(second,10,0), // yyyyymmddThhmmss

```

```

issue_from_identifier : //颁布者
    characterstring(iso-10646-1), // SPM: 1024
issue_date_time : //颁布时间
    time(second,10,0), // yyyyymmddThhmmss
issue_to_identifier : //被颁布的对象,不一定是学习者
    characterstring(iso-10646-1), // SPM: 1024
learning_experience_identifier : //学习体验/内容的标识
    characterstring(iso-10646-1), // SPM: 1024
competency_identifier : //与绩效信息有关的学力
    characterstring(iso-10646-1), // SPM: 1024
granularity : //绩效记录的粒度
    characterstring(iso-10646-1), // SPM: 300
performance_coding : //绩效编码方案
    characterstring(iso-10646-1), // SPM: 1024
performance_metric : //绩效度量
    characterstring(iso-10646-1), // SPM: 300
performance_value : //绩效值
    characterstring(iso-10646-1), // SPM: 2048
certificate_list : //数据认证信息
    arraylist(data_certification_type,200)
performance_bucket : //扩展信息
    arraylist(bucket_type,100),
),

```

11.9 CELTS 学习者作品集信息数据类型

在 ISO/IEC 11404 符号表示法中，以下是 CELTS 学习者作品集信息数据类型的摘要。

```

////////////////////////////////////
//// CELTS 学习者作品集信息
////////////////////////////////////

// CELTS 学习者作品集信息类型
type portfolio_info_type =
record
(
    portfolio_identifier : // 数据库连接(键)
        arraylist(learner_identifier_type,200),
    portfolio_hid : // HID 连接 (信息库间)
        arraylist(hid_type,200),
    media_id_type : // 媒体类型
        arraylist(MIME_type,200),
    media_id : // 作品相关媒体
        arraylist(URI_type,200),

```

```
media_lom_list : // 相关 LOM 引用的记录
    arraylist(LOM_reference_type,200),
media_performance_list : // 相关的绩效
    arraylist(performance_reference_type,200), //绩效引用类型
media_competency_definition_list : // 相关学力
    arraylist(competency_definition_reference_type,200),// 参见学力定义规范
certificate_list : // 数据认证信息
    arraylist(data_certification_type,200)
portfolio_bucket : // 扩展信息
    arraylist(bucket_type,100),
),
```

12 附录 C: XML 编码绑定(条件性标准)

如果 CELTS 学习者的应用在其的实现一致性声明中包括“CELTS 学习者 XML 编码[实现种类]”声明, 则必须符合本附件的要求。

注意: 实现种类在第 4 项“一致性”, 和在 4.2 子项“一致性标签”中定义。

12.1 生成和产生 XML

下列为 CELTS 学习者数据由本标准和 ISO/IEC11404 符号到 XML 文件的转换规则。

- **规则 1:** 除了在规则 2 中注明的之外, 对每个以 ISO/IEC11404 符号表示的数据元素, 将其标识符全部映射成 XML 标记。用成对的 XML 标记将数据元素的值封闭起来。正如在数据元素的“数据类型集的创建者”里所描述的 (ISO/IEC11404 术语), XML 标记的嵌套关系显示了数据元素的结构。对于数组和顺序集, (1) 一个与集合标识同名的 XML 标记表示该集合, (2) 单个数据元素表示方式是重复集合标识的 XML 标记去掉后缀“_list”或“_bucket”, 元素的索引不能表示数据元素。
- **规则 2:** 多语言字符串类型 (mlstring_type) 的数据类型全部映射到:
 - **规则 2A:** 多语言字符串类型 (mlstring_type) 的区域 (locale) 元素在父 XML 元素中设定了语言 (LANG) 属性。
 - **规则 2B:** 字符串 (string) 设置父标记的内容(也就是当前的目标)。

所有生成的数据必须是良好的 XML 结构

基本原理

下列各项规则是为 CELTS 学习者数据类型的转换而设的。

规则 1 的基本原理

规则 1 主要是把 ISO/IEC11404 表示的数据类型转换成 XML 标记规则。 下列例子说明转换方式:

A: record

```
(  
    B: integer,  
    C: record  
    (  
        D: integer,  
        E: characterstring(iso-10646-1),  
    ),  
    F_list: array (0..limit) of (integer),  
    G: sample_mlstring_list_type,  
)
```

第一句, “每个以 ISO/IEC11404 符号表示的数据元素, 将其标识符全部映射成 XML 标记”, 转换标识符, 例如, “X: ” \Rightarrow “<X>”。

第二句, “用成对的 XML 标记将数据元素的值封闭起来”, 要求(1) 标记是成对的, 和(2) 数据元素的值是两个标记之间, 例如, “X: 17” \Rightarrow “<X>17</X>”。

第三句, “正如在数据元素的‘数据类型集的创建者’里的描述, XML 标记的嵌套关系显示了数据元素的结构”, 生成的 XML 标记要与集合(记录、数组、序列/列)中所包含的嵌套关系相同。根据上例中 A 的定义, 以下为元素 B, C, D, 和 E 的嵌套关系:

```
<A>  
    <B>...</B>
```

```

    <C>
      <D>...</D>
      <E>...</E>
    </C>
    ...
  </A>

```

第四句，“数据元素以集合标识的 XML 标记表示，元素的索引不能表示数据元素。”，数组和序列(列表)要以多个同名的标记表示—XML 的典型规则。例如，数据元素 F 表示为：

```

<!-- 正确的 F_list 的 XML 绑定 -->
<A>
  ...
  <F_list>
    <F>...</F>
    <F>...</F>
    <F>...</F>
  </F_list>
  ...
</A>

```

而不是：

```

<!-- 错误的 F_list 的 XML 绑定 -->
<A>
  ...
  <F_list>
    <0>...</0>
    <1>...</1>
    <2>...</2>
  </F_list>
  ...
</A>

```

以学习者个人信息为例，其 XML 表示如下：

```

<personal_info>
  <personal_hid_list>
    <personal_hid>
      <identifier_type>CELT5-13 学生身份标识规范</identifier_type>
      <identifier_value>00112233</identifier_value>
    </personal_hid>
  </personal_hid_list>
  <name_list>
    <name>
      <official_name>
        <primary>阳</primary>
        <secondary>昊</secondary>
      </official_name>
      <full_informal_name>小昊</full_informal_name>
    </name>
  </name_list>
</personal_info>

```

```

    </name>
</name_list>
<telephone_list>
  <telephone>
    <context_label>家</context_label>
    <identifier_type>通话</identifier_type>
    <phone_number>+86 20 8711 2800</phone_number>
  </telephone>
  <telephone>
    <context_label>办公室</context_label>
    <identifier_type>通话</identifier_type>
    <phone_number>+86 20 8711 2800</phone_number>
  </telephone>
</telephone_list>
<email_contact_list>
  <email_contact>
    <context_label>常用</context_label>
    <email_address_type>rfc822</email_address_type>
    <email_address>cs_yh@mail.com</email_address>
  </email_contact>
  <email_contact>
    <context_label>不常用</context_label>
    <email_address_type>rfc822</email_address_type>
    <email_address>foo@bar.com</email_address>
  </email_contact>
</email_contact_list>
<postal_address_list>
  <postal_address>
    <context_label>学校</context_label>
    <addressee_name_given>昊</addressee_name_given>
    <addressee_name_family>阳</addressee_name_family>
    <delivery_street_name>五山</delivery_street_name>
    <delivery_street_type>路</delivery_street_type>
    <delivery_city>广州</delivery_city>
    <delivery_territory>广东</delivery_territory>
    <delivery_routing>510640</delivery_routing>
    <delivery_country>中国</delivery_country>
  </postal_address>
</postal_address_list>
<demography>
  <nationality>中国</nationality>
  <folk>汉</folk>
  <native_place>中国广东省湛江市</native_place>
  <birth_place>中国广东省广州市</birth_place>

```

```

    <birth_date>1979.01.01</birthdate>
    <gender>1</gender>
    <id_number>010101020202030303</id_number>
    <politic_status>团员</politic_status>
    <language>汉语</language>
    <health>好</health>
  </demography>
  <personal>
    <name>特长</name>
    <value>足球</value>
  </personal>
</personal_info >

```

规则 2 的基本原理

CELT5 学习者记录使用几个专门的数据类型，例如，为了描述某些必须用于不同语种和文化环境的“字符串-类型”数据元素，使用多语种的数据类型—统称为国际化和本地特性。以下是使用多语言字符串数据类型的版本例子，它并不与本标准中其它多语言字符串数据类型的定义矛盾。本例中，sample_mlstring_type 的数据类型表示一对：本地化字符串和地域（locale）。数据类型 sample_mlstring_array_type 表示一组这样的字符串对。本例中，数组 example_remarks 包含三个元素，每个元素是一对字符串。假设应用会根据使用的国家从 example_remarks 中选择适当的字符串。以下为值和类型定义的例子。

```

type sample_mlstring_type =
record
  (
    L10N_string: characterstring(iso-10646-1),
    L10N_locale: string_type,
  ),

type sample_mlstring_array_type =
  array (0..limit) of (sample_mlstring_type),

value example_remarks:
  sample_mlstring_array_type =
  (
    (
      L10N_string: "abc abc abc",
      L10N_locale: "en-US",
    ),
    (
      L10N_string: "def def def",
      L10N_map: "fr-CA",
    ),
  )

```

```

        L10N_string: "ghi ghi ghi",
        L10N_map: "de-DE",
    ),
),

```

规则 2，根据规则 1 的数组处理，数据元素转换为 XML:

```

<example_remarks LANG="en-US">abc abc abc</example_remarks>
<example_remarks LANG="fr-CA">def def def</example_remarks>
<example_remarks LANG="de-DE">ghi ghi ghi</example_remarks>

```

12.2 XML 的解释与消费

以下规则描述如何将 XML 记录转换成用 ISO/IEC11404 符号表示的 CELTS 学习者数据元素。

所有使用的数据都应是结构良好的 XML。

- **规则 1:** 进行下列转换:
 - **规则 1A:** 根据 XML 元素的语言 (LANG) 属性设定相应多语言字符串类型 (mlstring_type) 数据元素中的“区域” (locale)。
 - **规则 1B:** 根据标记元素中的内容设定相应多语言字符串类型 (mlstring_type) 数据元素中的字符串 (string)。
- **规则 2:** 与本标准中定义的 CELTS 学习者数据元素标识符相对应，每一段 XML 标记的起始和终止 XML 标记是成对的，并一一对应。上述规则 1 规定的除外，将每个 XML 标记映射成相应的数据元素标识。XML 标记的嵌套关系代表数据元素的嵌套关系，也就是说，本规则是上面 12.1 子项“生成和产生 XML”中的规则 1 的反过程。每个标记元素的内容转换成相应数据元素的值。

基本原理

规则 1 的基本原理

本规则描述从语言 (LANG) 属性反过来到多语言字符串类型 (mlstring_type) 数据类型的映射关系。因为并不是其它全部的 XML 语言 (LANG) 属性对应本标准的多语言字符串类型数据元素，本规则只转换已知的多语言字符串类型数据元素。

规则 2 的基本原理

本规则主要是关于 XML 标记和内容如何转换成数据元素。

第一句，“与本标准中定义的 CELTS 学习者数据元素标识符相对应，每一段 XML 标记的起始和终止 XML 标记是成对的，并一一对应”，(1) 忽略本标准没有的标识符，(2) 将它们配对。

第二句，“上述规则 1 规定的除外，将每个 XML 标记映射成相应的数据元素标识”，生成对应的数据元素，但是不赋值。

第三句，“XML 标记的嵌套关系代表数据元素的嵌套关系，也就是说，本规则是上面 12.1 子项‘生成和产生 XML’中的规则 1 的反过程”，在本标准的范围内，确保 XML 标记的内部结构与数据元素的内部结构相同。

第四句，“每个标记元素的内容转换成相应数据元素的值”，将 XML 标记中的内容转换成数据元素的值，也就是说，“移植”成数据元素。

12.3 基本数据类型的表示方式

下列各子项描述数据元素值到（或从）字符表示的转换方式，该方式用于在 XML 绑定范围里信息交换。

12.3.1 字符和字符串

字符类型的数据元素遵守每个 XML 规范。

注 1: 专用字符，如 “&”、“<”、“>”、“;” 需要转义，尽可能无损转义。

注 2: 某些编码方式，例如 ISO-8859-1 和 UTF-8 允许将字符表达式直接编码，例如 “©”（版权符号）。其它的编码方式，例如 ASCII，需要扩充编码，例如 “©”，来表示这些符号。

12.3.2 整数

整数类型的数据元素必须按“ISO/IEC9899:1999，C 程序设计语言，6.4.4.1 子项，整数常数”规定表示；除了带“U”，“L”，和“LL”后缀和它们的小写变量；开头可带符号，加号（“+”）或减号（“-”），但不能两者都有。

例子：

```
0      // 零
23     // 二十三
0x17   // 16 进制的 23
027    // 8 进制的 23
-34    // 负的三十四
+34    // 正的三十四
```

12.3.3 实数

实型数据元素：

- 如果是整数，可用整数表示，必须按上面的“12.3.2 子项，整数”规定；
- 如果非整型或不以整数表示，必须按“ISO/IEC9899:1999，C 程序设计语言，子项 6.4.4.2，浮点数”规定表示；除了带“F”和“L”后缀和它们的小写变量；开头可带符号，加号（“+”）或减号（“-”），但不能两者都有。

例子：

```
0      // 零
0.0    // 零
130.0  // 一百三十
1.3E2  // 一百三十
+1.3E2 // 一百三十
```

12.3.4 日期和时间值

时间类型的数据元素的表示必须按照“ISO 8601，数据元素和交换格式—信息交换—日期和时间的表示”规定。

注 1: ISO 8601 是规定了从公历在中间 0001 年 1 月 1 到 9999 年 12 月 31 日的日期和时间如何表示。这种表示方法是有不完善的地方，例如，在 1752 年 9 月中天数小于 30，涉及洲际铁路的时间前没有相关时区，不能变更日历的开始月份，不能表示公元前的日期，不能表示 9999 年 12 月 31 之后的日期。

ISO 8601 用法如下：

- 只能用基本格式。例如：“19990102”和“030405”是有效的，但是“1999-01-02”和“03:04:05”是不对的。基本原理：对扩展格式的附加处理容易出错并会减低互操作性；扩展格式的附加语法结构可能与其它周边的语法信息、固有信息或外部信息相互重叠或冲突。
- 十进制小数部分应使用句号（“.”），也作为分段号。例：“199901020304.1”表示1999年1月2日03:04:06。注：ISO 8601允许使用逗号的（“，”）和句号，并规定逗号是优先选用。本标准只允许句（分段）号。基本原理：逗号可能与其它周边的语法信息、固有信息或外部信息相互重叠或冲突。

以下用 ISO 8601 表示时间点：

- 日期只能以日历格式表示。注：日历是以年，月和日表示。ISO 8601 允许以其它格式表示日期，但本标准禁止使用，包括顺序日期（也就是，日期是以三位十进制数字表示，例如，“1985032”是1985年2月1日）和日历中的星期和星期几（也就是，日期是以星期数和每周的第几天表示，例如，1995年1月1日星期日表示为“1994W527”，1996年12月31日星期二表示为“1997W012”）。基本原理：对其它日期格式的附加处理容易出错并会减低互操作性；
- 日期和时间应使用 ISO 8601 中完整和不应使用“T”指示符。例：“19990102030405”代表1999年1月2日早上03:04:05。基本原理：“T”指示器符因无必要而被省略掉。在没有歧义时，ISO 8601 允许这种省略。以下方法可避免歧义：(1) 只允许 ISO 8601 基本格式；(2) 要求确定缺少的日期和时间。
- 要用下划线（“_”）来指出缺少的部分。日期时间的数字部分中，每一个下划线表示一个缺少的数字。例如：“__0102”表现当前的年的1月2日。注：数据中缺少某部分（例如，世纪），在互操作性方面会引起很严重的问题，例如“2000年问题”。实现应避免创建不完整的数据，但这也并不能解决所有问题。基本原理：ISO 8601 使用多个连字符（相当于减号）表示缺失部分，但是连字符可能与其它周边的语法信息、固有信息或外部信息相互重叠或冲突。ISO 8601 使用连字符表示缺失某一部分，（例如，“--0102”），而本标准每个数字用一条下划线代表（例如，“__0102”），处理这样的信息错误较少。
- 时间应以当地时间或参考世界时间（UTC 格林威治时间）。以当地时间表示的时间不用后缀也不指出和 UTC 时间差异。按 ISO 8601 规定，以 UTC 表示的时间要用“Z”后缀。例：“19990102030405Z”纽约的当地时间（UTC 以西 5 小时）与时间“19990102030905Z”相同。基本原理：不用 UTC 而使用其它时区信息，引起额外的数据解释处理，会更易于出错。使用加号（“+”）和减号（“-”）可能与其它周边的语法信息、固有信息或外部信息相互重叠或冲突，不允许时间的增减（例如，“19990102030405+0700”）排除了这种可能。

以下用 ISO 8601 表示时间段：

- 按照 ISO 8601 规定，时间段以“P”开头，例子：“P2Y”是二年的意思。注意：时间段不一定有完整的日期和时间。
- 时间段应使用区分大小写的标志：“Y”为年，“M”为月，“D”为天，“W”为星期，“h”为小时，“m”为分，和“s”为了秒。例子：“P2Y10M25h2m5s”表示时间段2年，10月，25小时，2分，和5秒。时间段中，日期和时间可超过时间点中相应部分的上限，例如，月超过12，小时超过24，和分超过60。基本原理：使用区分大小写的标志简化日期和时间信息的处理。

注2：在 ISO 8601 中的字符只规定了字符的名字，而不是它们的编码方式。ISO 8601, 4.4 条，“用在表达式中的字符”强调：“在本国际标准规定的表达式所使用的数字，字符，和特殊符号规定在 ISO 646. ...

注2：用于交换日期和时间的字符，其编码方式不在本标准的范围中。”

12.3.5 空类型

空类型不应表示，也没有编码。

例子：下列记录

```
A: record
(
  B: integer,
  C: void,
  D: characterstring(iso-10646-1),
)
```

用 XML 表示为：

<!-- C 的正确 XML 表达-->

```
<A>
  <B>17</B>
  <D>hello</D>
</A>
```

而不是：

<!-- C 的错误 XML 表达-->

```
<A>
  <B>17</B>
  <C></C>
  <D>hello</D>
</A>
```

12.3.6 字符表达式的编码

XML 编码方法规定了字符表达式如何编译成字节的值。

与 CELTS 学习者 XML 编码绑定一致的实例必须使用以下编码方式的其中一种：ASCII，ISO/IEC8859-1，ISO/IEC10646-1 UTF-8，或 ISO/IEC10646-1 UTF-16。

与 CELTS 学习者 XML 编码绑定一致的应用必须支持以下所有的编码方式：ASCII，ISO/IEC8859-1，ISO/IEC10646-1 UTF-8，和 ISO/IEC10646-1 UTF-16

12.4 例外与扩展处理

12.4.1 实现定义行为

除了本标准的其它说明外，以下也属于实现定义行为。

在 XML 编码的生成和使用过程中，以下是实现定义行为：

- 严格一致的 CELTS 学习者数据实例，以 XML 编码，可处理的最大字节数。
- XML 记录的最大的嵌套层数。
- 时间类型数据元素的时区信息中没有指明的时区。

12.4.2 未指定的行为

除了本标准的其它说明外，以下也属于未指定行为。

在创建和解释 XML 编码的过程中，以下是未指定行为：

- 数据元素处理的顺序。

在 XML 编码的生成和使用过程中，以下是未指定行为：

- 数据元素值和 XML 规范以外的附加字符的使用。

12.4.3 未定义行为

除了本标准的其它说明外，以下也属于未定义行为。

在 XML 代码的生成和使用过程中，以下为未定义行为：

- 与扩展数据元素相应的 XML 标记的使用。
- 与保留数据元素相应的 XML 标记的使用。
- 本 XML 绑定中未指明的属性或 XML 标记的使用。
- 本标准指令表以外的字符的使用。

13 附件 D: DNVP 编码绑定 (条件性标准)

如果一个CELT学习应用用在它的实现一致性声明中包括“CELT学习DNVP编码【实现种类】”，那么应用须遵守这个附件的要求。

注意 1: 实现种类在第4小节“一致性”中定义，在4.2小节“一致性标签”中总结。

注意 2: 带点名值对 (DNVP) 是以RFC 822通信方式为基础。RFC 822, ARPA英特网文本信息的格式标准, 描述常用作互联网电子邮件的文本信息。这种类型的信息, 开始部分包含标题, 带有多个标题元素。例如,

```
from: sender@host.com
```

```
to: user@host.com
```

```
Subject: 主题行
```

可表示三种标题元素—标题的部分。这类通信的在RFC 822, 3.1小节“概述”中描述。此外, RFC 2068, “超文本传输协议” - HTTP/1.1, 4.2小节, “信息标题”, 它与RFC 822一致(摘自RFC 2068, 4.2小节: “HTTP标题... 遵循RFC 822的第3.1节所示的格式”)。

这种绑定形式 (RFC-822-类) 常在许多交换环境中使用, 例如, 电子邮件系统和网络服务器。

下列是常用格式的例子:

```
name_1: value_1
```

```
name_2: value_2
```

```
name_3: value_3
```

13.1 带点名值对 (DNVPs)

注意: 以下句子从RFC 2068(HTTP/1.1)中摘录、概括和改编, 并与之统一。

13.1.1 基本词汇

新行符 (CRLF) 由它的编码定义。

注意 1: 一个新行符可能是换行 (例如, Unix/Linux系统), 回车 (例如, Macintosh系统) 或回车换行组合。 (例如, windows系统)

当产生数据时, 为了有最大的互操作性, 实现应该使用回车换行组合作为新行符。 实现读取数据时, 如只用其中一个字符 (换行或回车) 作为新行符, 则应忽略另一个字符 (回车或换行)。

行首空格 (LWS) 是紧跟新行符后的至少一个或多个空格或制表符。

注意 2: LWS不包括之前的新行符。前面没有新行符的一系列空格或制表符不是LWS。

控制符 (CTL) 是在以下范围中的一个字符: 0到31 (十进制) 或八位字节127 (十进制), 八位字节9 (HT) 除外。

文本字符是除了控制符以外的任何字符。

以下是特殊符号:

```
( ) < > @  
, ; : \ "  
/ [ ] ? =  
{ } SP HT
```

标记是除了控制符或特殊符号以外的一个或多个字符。

如一串文字处于双引号中, 它将被视为一个单个标记读取 (且特殊符号不再特殊)。

例 1: 字符

```
"abcd -(), :[]"
```

被视为一个单个标记读取。

转义符("\") 可用于表示单个字符，也就是，除去特殊符号的特殊性质，每次一个字符。

例 2: 字符

they're

读作一个单个标记，而单引号(')没有特别的意义。

例 3: 字符

"say \"hello\""

读作一个单个标记， 标记包括 hello 之前和之后的双引号， 且与 hello 相邻的双引号没有特殊含意。

13.1.2 字段名和字段值

名值对(NVP)应包含一个字段名，其后带冒号(":"), 再加上字段值。字段名是一个区分大小写的标记。

注意: DNVP绑定的区分大小写与RFC 822的不同。

字段值之前可以有任意个行首空格(LWS)。一致的实现应以一个空格(SP)作LWS。字段值是零个或多个标记或特殊符号组合而成。

字段值的字符数为零表示字段名相关的值为空。

注意: 名值对从行首开始。

13.1.3 新行处理

名值对可延续多行，新增每行前至少带有一个空格(SP)或横表符(HT)。所有连续的空格，包括折叠的，其意义如同单个SP符。

新行(CRLF)应跟在完整的名值对之后。

13.1.4 语法摘要

本节是信息性的而不是规范性的。

注意: 以下巴柯斯范式(BNF)语法摘要是从RFC 2068(HTTP/1.1)中摘录、归纳、改写，并与之一致:

```
回车换行(CRLF)      = <新行符>
控制符(CTL)         = <任意 US-ASCII 控制字符>
                      (字符 0 - 31) 和 DEL (127), 除了 HT (9)>
行首空格(LWS)      = [CRLF] 1*( SP | HT )
标记(token)        = 1*<除 CTL 和 tspecials 以外的任意字符>
特殊符号(tspecials) = "(" | ")" | "<" | ">" | "@"
                    | ", " | ";" | ":" | "\" | "<"
                    | "/" | "[" | "]" | "?" | "="
                    | "{" | "}" | SP | HT
转义符(quoted-pair) = "\" 一个字符
带引号字符串(quoted-string) = <">*<除没有转义的双引号以外
                              的任意字符><">
报头(message-header) = field-name ":" [ field-value ] CRLF
字段名(field-name)   = token
字段值(field-value)  = *(字段内容 | LWS )
```

字段内容 (field-content) = <构成字段值的字符, 由标记和特殊符号组成>

13.2 生成和产生带点名值对

以下描述CELT5学习者数据元素, 从本标准和ISO/IEC 11404到DNVP表示法的转换规则。

- **规则 1:** 对于每种数据元素, 将所有标识映射为完全-受限的字段名。完全-受限的名字可表示数据元素的嵌套结构, 通过它的“聚合数据类型生成者”描述(ISO/IEC 11404 术语)。在完全-受限的字段名中嵌套的各层之间要用分隔符(“.”)分开。对于数组和顺序集, (1) 各数据元素以作为集合标识的 DNVP 字段名除去后缀“_list”或“_bucket”表示, 而不是数据元素的索引; (2) 数组或顺序集中的每个元素应以两个值为空的 DNVP 包围起来, 每个元素开头以字段名加后缀“.__begin”, 并以字段名加后缀“.__end”结束。字段名后带冒号(“:”), 再加上数据元素相应的值。

- **规则 1 的基本原理**

规则1是从ISO/IEC 11404的数据类型到DNVP表示法的主要转换。下列各例用如下定义说明其转换:

```
A: record
(
  B: integer,
  C: record
  (
    D: integer,
    E: characterstring(iso-10646-1),
  ),
  F_list: array (0..limit) of (integer),
)
```

开始的三个句子, “对于每种数据元素, 将所有标识映射为完全受限的字段名。完全受限的名字可表示数据元素的嵌套结构, 通过它的‘聚合数据类型生成者’描述(ISO/IEC 11404术语)。在完全受限的字段名中嵌套的各层之间要用分隔符(‘.’)分开。” , 把标识转换成字段名, 例如:

```
A.B
A.C.D
A.C.E
```

第四个句子, “对于数组和顺序集, 各数据元素以作为集合标识的DNVP字段名表示, 而不是数据元素的索引”, 数组和序列(列表)要以多个名字相同的DNVP表示 - RFC 822类系统允许这样, 但意义有些不同。(RFC 822允许各行并列, 而此CELT5学习者编码绑定不允许DNVP的自动合并)。例如, 数据元素F可表示为:

(F的正确的DNVP绑定)

```
A.F.__begin:
A.F: xxx
A.F.__end:
A.F.__begin:
A.F: yyy
A.F.__end:
A.F.__begin:
A.F: zzz
A.F.__end:
```

而不是:

(F的不正确的DNVP绑定)

A.F.0: xxx

A.F.1: yyy

A.F.2: zzz

第五个句子,“字段名后带冒号(“:”),再加上数据元素相应的值”,把字段名和字段值连起来,并用冒号(“:”)隔开。

例:

A.B: 17

A.C.D: 34

A.C.E: yellow pigs

A.F: 51

A.F: 68

A.F: 85

13.3 带点名值对的读取和解释

以下用本标准和ISO/IEC 11404符号表示法,描述从DNVP到CELTSL学习者数据元素的转换规则。

- **规则 1:** 由于每个字段名对应本标准的 CELTS 学习者数据元素中定义的一个标识,将每个字段名映射为相应数据元素的标识。字段名的嵌套关系代表了数据元素的嵌套关系,就是说,与上述 13.2 小节“生成和产生带点名值对”的规则 1 中的操作相反。每个字段值转换成对应数据元素的值。集合的字段值为空表示集合存在,而不是其元素的值。

- **规则 1 的基本原理**

这条规则是关于从DNVP到数据元素的主要转换。

第一个句,“由于每个字段名对应本标准的CELTSL学习者数据元素中定义的一个标识,将每个字段名映射为相应数据元素的标识”,(1)忽略所有不在本标准中的标识,(2)把相应的字段名和数据元素标识配对,并(3)联结数据元素,但不要赋值。

第二个句,“字段名的嵌套关系代表了数据元素的嵌套关系,就是说,与上述13.2小节‘生成和产生带点名值对’的规则1中的操作相反”,确保DNVP的内部结构与数据元素的内部结构一致,符合本标准的要求。

第三个句,“每个字段值转换成对应数据元素的值”,转换数据元素的字段值,也就是,把它“移植”到数据元素中。

第四个句,“集合的字段值为空表示集合存在,而不是其元素的值”,仅仅是建立集合。

以下片段中,A.C中有空值,意味着集合存在:

A.B: 17

A.C:

A.C.D: 34

A.C.E: yellow pigs

当集合中含有可选数据元素时,空值是有用的,就是说,它表示集合存在,但其中没有元素:

A.B: 17

A.C:

空类型不会用空类型来表示。

13.4 基本数据类型的表示法

以下各小节描述数据元素值与字符表达式的相互转换，转换目的是为了在DNVP绑定范围内交换信息。

13.4.1 字符和字符串

根据RFC 1522的规则编码时，字符型的数据元素只能以ISO/IEC 8859-1中的字符表示。

13.4.2 整数

整数类型的数据元素的表示方法如上面12.3.2小节所述。

13.4.3 实数

实数类型的数据元素的表示方法如上面12.3.3小节所述。

13.4.4 日期和时间的值

时间类型的数据元素的表示方法如上面12.3.4小节所述。

13.4.5 空类型

空类型不应表示也没有编码。

例：以下记录

```
A: record
(
  B: integer,
C: void,
D: characterstring(iso-10646-1),
)
```

RFC 822 类的绑定表示为：

(C 的正确 DNVP 绑定)

```
A.B: 17
A.D: hello
```

而不是：

(C 的不正确 DNVP 绑定)

```
A.B: 17
A.C:
A.D: hello
```

13.5 字符表达式的编码

一致的CELTIS学习者的DNVP编码绑定把字符表达式按ISO/IEC 8859-1标准编译成数值。该指令集以外的字符会依照RFC 1522的规则进行编译。

13.6 例外和扩展处理

13.6.1 实现定义行为

除了在本标准的其它地方说明外，以下各项也是实现定义行为。

在DNVP编码的产生和读取过程中，下列是实现定义的行为：

- 新行符的编码字符。注意：实现可用回车换行符替代新行符，而不用实现定义行为。参阅前面的 13.1.1 小节“基本词汇”。
- 严格一致的 CELTS 学习者数据实例的 DNVP 编码中，能成功处理的最大字符数。
- 未指定时区的时间类型数据元素的时区信息。

13.6.2 未指定行为

除了在本标准其它方面描述之外，以下是未指定行为。

在DNVP的生成或解释过程中，以下是未指定行为：

- 处理数据元素的次序。

13.6.3 未定义行为

除了在本标准其它方面描述之外，以下是未定义行为。

在DNVP的产生或读取过程中，以下是未定义行为：

- 扩展数据元素相应字段名的使用。
- 保留数据元素相应字段名的使用。
- 本 DNVP 编码绑定没有指定的字段名使用。
- 本标准说描述的指令集以外的字符使用。