



中华人民共和国国家标准

GB/T×××—2007

信息技术 学习、教育和培训 测试互操作

Information technology - Learning, education and training -
Standard for test interoperability

(报批稿)

2009-××-××发布

2009-××-××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.....	7
2 规范性引用文件.....	7
3 术语和定义.....	8
4 规范用例	1
4.1 用例角色.....	2
5 题项.....	3
5.1 题项时间段生命周期.....	4
6 题项变量.....	8
6.1 响应变量.....	21
6.1.1 内嵌响应变量.....	22
6.2 结果变量.....	23
6.2.1 内嵌结果变量.....	25
7 内容模型.....	25
7.1 基本类.....	25
7.2 XHTML 元素	26
7.2.1 文本元素.....	28
7.2.2 列表元素.....	29
7.2.3 对象元素.....	30
7.2.4 呈现元素.....	31
7.2.5 表格元素.....	31
7.2.6 图像元素.....	34
7.2.7 超文本元素.....	34
7.3 MathML.....	34
7.3.1 联合模版变量和 MathML	34
7.4 变量内容.....	34
7.4.1 数字格式化规则.....	34
7.5 带样式表的格式化题项.....	36
8 交互.....	37
8.1 简单交互.....	39
8.2 基于文本的交互.....	44
8.3 图像交互.....	48
8.4 混合交互.....	53
8.5 结束交互的替换方式.....	55
9 响应处理.....	55
9.1 响应处理模板.....	55
9.1.1 标准模板.....	56

9.2	普通响应处理.....	56
10	模式反馈.....	59
11	题项模板.....	59
11.1	题项中使用模板变量.....	60
11.2	操作符属性值中使用模板变量.....	60
11.3	模板处理.....	61
12	测试.....	62
12.1	浏览与提交.....	63
12.2	测试结构.....	65
12.3	时间限制.....	70
13	结果处理.....	71
14	测试级别反馈.....	73
15	前提和分支.....	73
16	表达式.....	74
16.1	内嵌普通表达式.....	74
16.2	结果处理表达式.....	77
16.3	操作符.....	79
17	题项和测试片断.....	86
18	基本数据类型.....	87

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：清华大学、湖南大学。

本标准主要起草人：郑莉、杨贯中、张新钰、李秀、李鹏、陶勇、孔婷。

引 言

本标准定义了表示测试/试题内容结构的基本数据模型，目的是使得测试数据能在不同的学习管理系统、内容开发系统和资源库等之间进行互用，最终达到测试/试题内容的互操作需求。

通过定义一个通用的概念数据模型，保证测试/试题内容的绑定之间有较高程度的语义互操作性。这样，不同绑定之间的转换就会变得简单直接。

本标准规定了一个基本的信息模型，在此之上能建立各种不同的实现。

信息技术 学习、教育和培训

测试互操作

1 范围

本标准规定了测试/试题的基本信息模型，主要与学习内容开发者、学习工具开发者、学习环境和测试/试题使用者有关。本标准适用于满足测试/试题内容的互操作性需求，主要有以下职能：

- a) 不论用户处于何种学习环境，考试中心都能为用户提供试题；
- b) 用户能使用所处学习环境中的各种试题资源；
- c) 支持各种工具开发兼容的题项/题库；
- d) 支持不同形式的结果报告之间、基于不同平台的结果报告之间能够交互。

本标准据此提出以下按优先级进行排列的需求：

- a) 标记一系列必选或可选元素，以定义试题、响应、计分机制以及元数据等的属性；
- b) 测试的互操作性——封装和发布的定义；
- c) 可扩展的结果报告的标准化模式；
- d) 可扩展的评估、跟踪和编排方式的模式；
- e) 动态题项获取和计分/评估机制的接口（APIs）；
- f) 考虑到未知和特定领域等的需求，本标准具有相当的可扩展性和灵活性。

本标准定义了一个概念上的数据模型，用于定义测试/试题实例的结构。目的是使得测试数据能在不同的学习管理系统、内容开发系统和资源库等之间进行互换。本标准具有可扩展性且易于实现。

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 1988-1998 信息处理交换用的七位编码字符集（eqv ISO/IEC 646:1991）

GB/T 2659-2000 世界各国和地区名称代码（eqv ISO 3166-1:1977）

GB 4880-1991 语种名称的编码表示（ISO 639-1: 2002, MOD）

GB/T 7408-1994 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 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）

GB13000 信息技术统一的多字节编码字符集（UCS）——第1部分：体系结构和基本多语言文种平面（idt ISO/IEC 10646-1: 1993）

IETF RFC 2048: 1996 多用途的网际邮件扩充协议（MIME）第四部分：注册过程，1996-11

IETF RFC 2425: 1998 目录信息的 MIME 内容一类型

3 术语和定义

下列术语和定义适用于本标准。

3.1

自适应题项 Adaptive Item

自适应题项可以调整自己的外观，或是计分（响应处理），或是两者，以响应应试者的每次尝试。

例如，一个自适应题项开始时可能给应试者提供文本编辑框以回答问题，一旦接收到一个不正确的回答，就改为单一选择形式与应试者交互，并对后来的正确回答判较低的分。

3.2

自适应测试 Adaptive Test

自适应测试是根据应试者已经完成题项的情况，改变呈现给应试者题项的测试。

3.3

评估 Assessment

评估是测试应试者某方面的过程，在本标准的里，评估是用测试实现的，术语评估被视为与评估测试相同。

3.4

评估测试 Assessment Test

评估测试是一个有组织的题项集合，题项用于决定应试者的能力。评估测试包含题项排序和计算评分的说明。

3.5

评估变量 Assessment Variable

评估变量是用于保持题项时间段或测试时间段相关值的变量。例如，应试者给定的响应的值，或单独题项或整个测试结果的值。

3.6

评估递送系统 Assessment Delivery System

用于管理和递送评估给应试者的系统。参见递送机制。

3.7

尝试 Attempt

尝试（对于一个题项）是应试者在一个或多个应试者时间段内，与一个题项进行交互的过程，可能指定或更新相关的响应变量。

3.8

开发系统 Authoring System

开发者用于创建和编辑题项和评估的系统。

3.9

基础类型 Base-type

基础类型是一个预先定义的数据类型，它定义了一个供题项变量选取的值的集合。这些值对于本标准描述的运行时模型是不可再分的。

3.10

基本题项 Basic Item

基本题项是包含且只包含一个交互的题项。

3.11

应试者 Candidate

通过回答问题参与测试、评估或考试的人。

3.12

应试者时间段 Candidate Session

应试者与题项进行交互，作为尝试的一部分的一段时间。一次尝试可由应试者多个时间段组成。例如，对于某个问题答案没把握的应试者可能会浏览同一个测试中的另一道问题，随后再返回第一个问题。当他们离开第一个问题的时候，他们终止了应试者时间段，但是他们没有终止尝试。尝试只是暂停，直到后来的应试者时间段结束它，触发响应处理和（可能的）反馈。

3.13

克隆机制 Cloning Engine

克隆机制是用于从一个题项模版创建若干相似题项（克隆题项）的机制。

3.14

复合题项 Composite Item

复合题项是包含多于一个交互的题项。

3.15

容器 Container

容器是一个聚合数据类型，可以包含若干简单的基础类型的值。容器可以是空的。

3.16

递送机制 Delivery Engine

调整题项和响应的评价呈现和递送，以产生分数和反馈的过程。

3.17

反馈 Feedback

返回给学习者，以指导学习过程的信息。

3.18

交互 Interaction

交互允许应试者与题项交互。通过交互，应试者选择或创建一个响应。

3.19

综合反馈 Integrated Feedback

综合反馈是被综合到题项的题项体中的反馈。与模式反馈不同，在浏览综合反馈的时候，应试者可以自由更新他们的响应

3.20

题项 Item

最小的可交换的评估对象。题项不只是一个问题，它需要包含呈现的问题和说明，应用于应试者响应和可能呈现的反馈（包括提示和解答）的结果处理

3.21

克隆题项 Item Clone

克隆题项是由题项模版创建的题项。

3.22

题项片断 Item Fragment

题项片断是题项的一部分，由包含它的题项独立管理。参见题项组。

3.23

题项时间段 Item Session

题项时间段是一个应试者对一个特定的题项所有尝试的累积。

3.24

题项组 Item Set

题项组是一系列共享一些共同特征的题项。例如，组内的所有题项可能被同一个题项片断引用，在这种情况下，片段经常被称为组的领导。

3.25

题项模版 Item Template

题项模版是用于产生大量相似题项的模版。这样的题项经常被称为克隆题项。题项模版通过专门的克隆机制产生题项，或者在递送机制支持的情况下，用于在题项时间段开始时，直接动态选择克隆。除了给定的一系列模版变量的值，从题项模版克隆的每个题项都是等同的。所以，如果题项声明一个或多个模版变量并且包含一系列为它们赋值的模版处理规则，该题项就是题项模版。

3.26

题项变量 Item Variable

记录题项时间段部分状态的变量。应试者的响应和任何响应处理赋值的结果存储在题项变量中。题项变量也用于定义题项模版。题项变量是一种特殊的评估变量。

3.27

素材 Material

素材是所有提供给用户而不是被系统解释的静态文本、图象或媒体对象。交互不是素材。

3.28

模式反馈 Modal Feedback

模式反馈是独自呈现给应试者的反馈，不被综合到题项的题项体内。

3.29

多选题 Multiple Response

多选题是从基础类型值的集合中选取若干值的一种容器。多选题被处理为这些值的无序链表。链表可以为空。

3.30

非自适应题项 Non-adaptive Item

非自适应题项是不会根据应试者的尝试而调整自己的题项。

3.31

对象库 Object Bank

对象库是用于评估，包括题项、题项片断、测试或测试部分的对象的集合。对象库在本信息模型里未被直接呈现。

3.32

有序响应 Ordered Response

有序响应是响应变量，该响应变量是若干值都从一个基础类型定义的值的集合中选取的容器。有序响应被处理为值的有序链表。链表可以是空的。

3.33

结果 Outcome

评估测试或题项的结果。结果被一个或多个结果变量呈现。

3.34

结果处理 Outcome Processing

题项结果（或响应）的值被合计而获得测试结果的过程。

3.35

结果变量 Outcome Variable

结果变量被结果声明声明。它们的值或者由声明本身默认给定，或者被响应处理（对于题项结果）或结果处理（对于测试结果）时遇到的响应规则设定。

3.36

池 Pool

一组和元数据一起传输的关联题项，该元数据把组描述成一个整体。池是对象库的特例。

3.37

响应 Response

应试者通过与题项或题项的一部分交互提供的的数据。应试者响应的值由响应变量呈现。

3.38

响应处理 Response Processing

响应变量的值被评判（计分），以及题项结果的值被赋值的过程

3.39

响应变量 Response Variable

响应变量被响应声明定义，受限于题项体的交互。它们记录应试者的响应。

3.40

计分机制 Scoring Engine

评估系统的一部分，处理基于应试者响应和响应处理规则的计分。

3.41

测试反馈 Test Feedback

基于测试结果的值，有条件的呈现给应试者的反馈。

3.42

单一响应 Single Response

单一响是从由一种基础类型定义的值集合中取一个单一值的响应变量。

3.43

模版处理 Template Processing

用于设定模版变量值的一系列规则，典型的包括一些随机过程，从而选择用于题项时间段的具体克隆。

3.44

模版变量 Template Variable

模版变量被模版声明声明，用于记录初始化一个题项模版必需的值。值决定题项模版定义的相似题项集合中哪个克隆用于给定题项时间段。

3.45

测试片断 Test Fragment

测试片断是测试的一部分，由包含测试片断的测试独立管理。

3.46

测试报告 Test Report

测试报告是关于测试时间段的报告。

3.47

测试时间段 Test Session

测试时间段是应试者与测试及其所包含的题项的交互。

3.48

时间相关题项 Time Dependent Item

时间相关题项是记录测试者累积消耗时间的题项，记录存在响应处理使用的响应变量里。

3.49

时间无关题项 Time Independent Item

时间无关题项是在响应处理中不使用累积消耗时间的题项。在实践中，这个信息可以被递送机制收集并且作为测试报告的一部分。

4 规范用例

图1 介绍了基本的QTI体系结构所包含的系统组件、数据结构和主要角色。

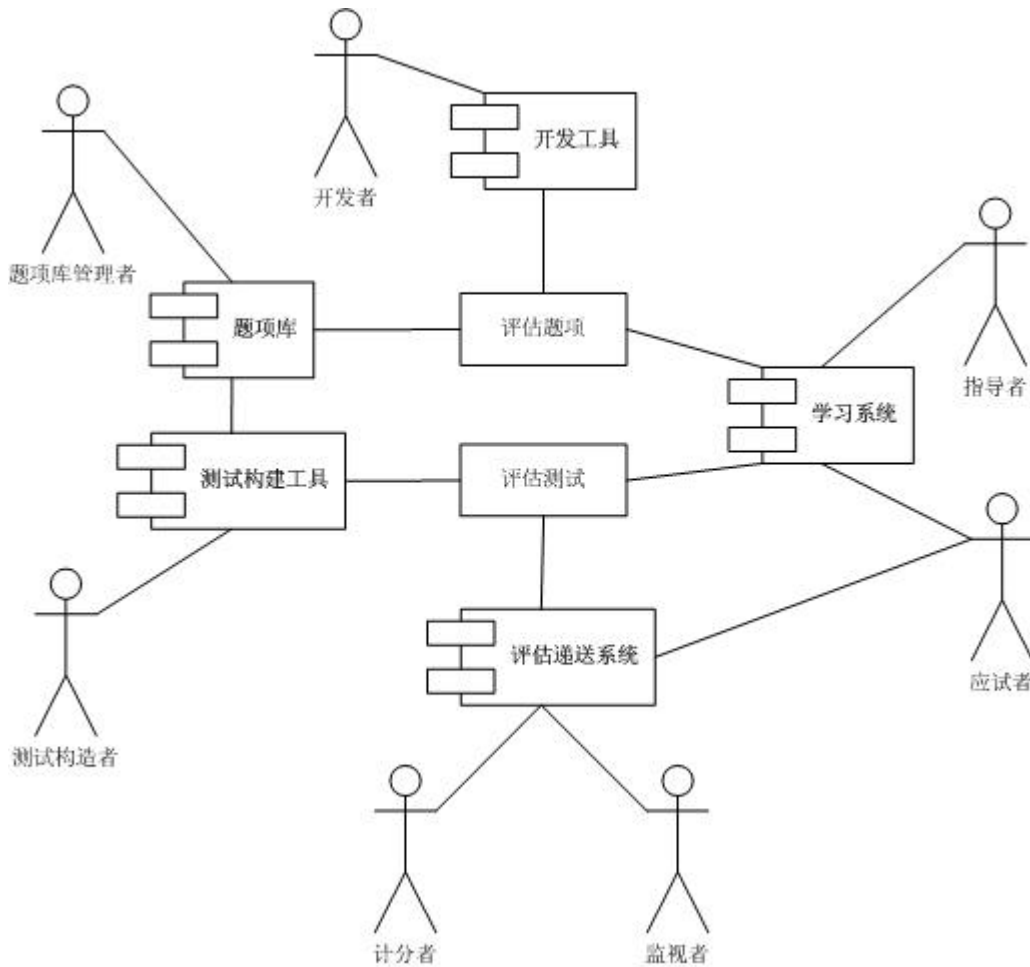


图1 评估测试和评估题项的角色

——其中关键系统组件有：

- 开发工具——开发者用于创建和修改、评估题项的工具。
- 题项库——储存和管理评估题项的系统
- 测试构建工具——将每个题项组合成测试的工具。
- 评估递送系统——负责评估的传递，包含递送引擎，利用该引擎可以把题项传给应试者，也可以把应试者的响应传给计分者。
- 学习系统——学习活动中由指导者授权、调整或指导学习者的系统。本标准中，与学习系统交互的部分学习者，依然描述为应试者。虽然学习系统采用的管理和安全模型与评估递送系统有较大区别，但学习系统也可认为包含递送机制。

4.1 用例角色

本标准中所用到的角色简化为小的抽象角色集合。而真实学习和评估系统中的角色可能更加复杂，为了简化叙述，假定它们可以被抽象为一个或多个这里定义的角色：

- 开发者——开发评估题项的角色。在简单情况下，一个题项可以有单一开发者，在更复杂的情况下，题项可能经过一个包括很多人的创建和质量控制的过程。在本标准中，认为所有这些人的角色都为开发者。开发者与题项的内容有关，这一点与题项库管理者角色相区别。开发者通过开发工具与题项交互。
- 题项库管理者——利用题项库管理评估题项的角色。
- 测试构造者——测试构造者的任务是从题项库中选出典型题项，并组成测试。

- d) 监视者——负责监视评估递送的人，常被称为监考人。在本标准中，监视者是与递送过程有关但是不承担评估应试者任务的人。
- e) 计分者——在评估递送期间，负责评估应试者响应的人或外部系统。计分者是可选的，例如很多测试题项可以用题项自定义的响应处理规则自动计分。
- f) 指导者——管理、指导或支持学习者的学习过程的人，并不限于（同一个）评估系统。
- g) 应试者——被评估测试或评估题项评估的人

5 题项

题项包括呈现给应试者的题目本身信息和计分信息，计分发生在应试者回答题项，并在应试规则转换为结果的时候。有时需要多个不同的题项，在应试者看来是相同的，但计分不同。在本标准中，这些题项在定义上必须有不同的标识符。为了能方便的交换和共享题项中的重要信息，本标准支持在题项中定义和管理题项片断。图 2 表明了题项类和其他相关类的关系，后面将分别进行详细的介绍。

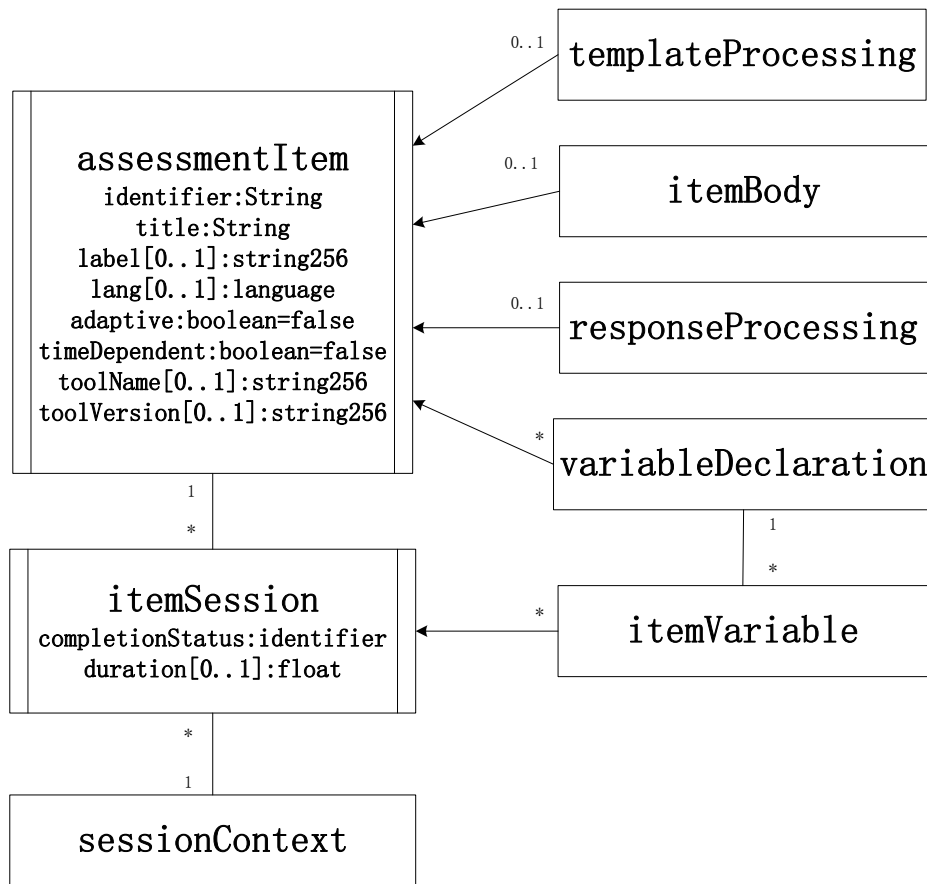


图 2 题项类 (assessmentItem) 和其他相关类关系

评估题项类及其属性说明

Class : assessmentItem

属性名	多重性	类型	说明
-----	-----	----	----

identifier	[1]	string	题项的标识符。
title	[1]	string	题项的标题。应试者可以通过标题选择题项，而不必浏览题项内容。递送机制可以在任何时候显示标题给应试者，但这一点并不是必须的。
label	[0..1]	string256	
lang	[0..1]	language	
adaptive	[1]	boolean = false	题项分为自适应题项和非自适应题项
timeDependent	[1]	boolean	
toolName	[0..1]	string256	工具名属性的作用是通过题项可以识别创建它的工具。其他系统可用这个信息解释应用具体数据的内容。如题项体Itembody的元素标签。
toolVersion	[0..1]	string256	通过题项可以识别创建自己工具的版本。该属性只能和toolName属性一起使用。

包含:

类名称	多重性
responseDeclaration	[*]
outcomeDeclaration	[*]
templateDeclaration	[*]
templateProcessing	[0..1]
stylesheet	[0..*]
itemBody	[0..1]
responseProcessing	[0..1]
modalFeedback	[*]

5.1 题项时间段生命周期 (itemSession)

题项时间段生命周期对应图2中的itemSession,是应试者对于特定题项的实例作出的所有尝试的时间累积。在某些类型的测试中,同一个题项可能向一个应试者呈现多次。题项实例的每次出现都与题项时间段有关。

图3阐述了在题项生命周期内,用户和题项的交互。在实际场景中并不是所有的交互状态都会出现。例如,题项可能不提供反馈,或者说在题项使用的环境中反馈不允许。与此类似,应试者可能不允许检查他们的答案等。在实际中,系统可以只支持有限数量的状态转换和/或支持这里没有的其它状态转换。

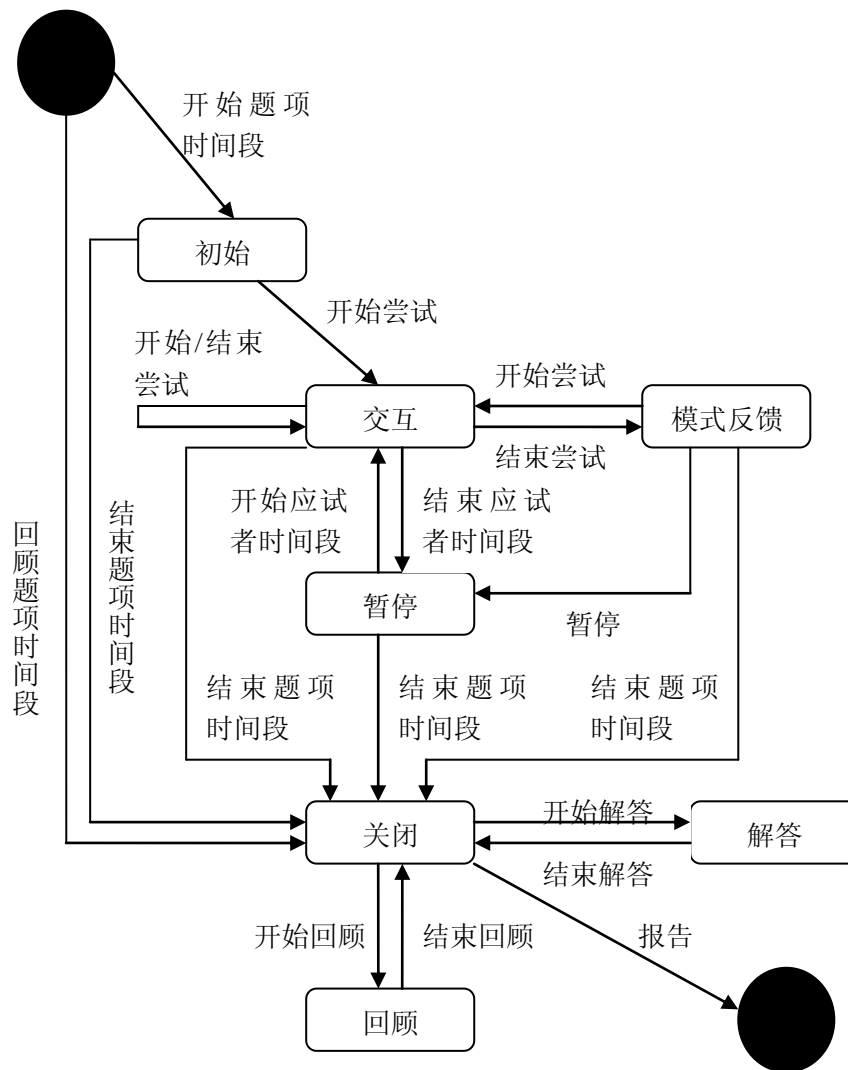


图3 题项时间段的生命周期

对于系统开发者，将需求应用于实现的重要一步是分析系统中应该支持哪些状态，并且在自己的事件模型图表中，表明支持的状态及转换。

题项时间段生命周期开始于题项第一次符合递送条件，传递给应试者的时候。此后，题项的状态就通过应试者的行为被保持和更新，直到时间段结束。在任何时候，时间段的状态都可以进入题项响应结果状态。为了能让应试者在题项中可以看到自己的回答以及重新开始一个回答，递送系统也可允许题项响应结果状态做为新题项时间段生命周期的开始。

题项时间段的初始状态是指题项被决定递送给应试者后，递送动作发生之前的状态。

在典型的非自适应测试中，会提前选出题项，而不管应试者是否尝试过所有题项，应试者与所有题项的交互在测试时间段结束时才被报告。在测试开始的时候，所有题项的时间段处于初始状态，并且保持并行。在自适应测试中，将要呈现的题项根据已经呈现的题项的响应结果而呈现。递送机制实际上只报告应试者与被选中的题项的交互。

应试者与一个题项的交互被分解成0次或多次尝试。在每次尝试中，应试者在一个或多个时间段中与题项进行交互。应试者时间段结束时，题项可以置于暂停状态，并为下一个应试者时间段准备就绪。在一个应试者时间段中，题项时间段处于交互状态。一次尝试结束，响应处理就发生，响应处理之后可以开始新的尝试。

对于非自适应题项，响应处理发生次数有限，通常只有一次。对于自适应题项，没有此限制，因为响应处理根据题项的呈现顺序而调整赋给结果变量的值。在以上两种情况下，尝试结束时都会调用响应处理。题项体的外观，以及模式反馈是否出现，取决于结果变量的值。

当不允许在进行更多尝试的时候，题项时间段进入到关闭状态。一旦进入关闭状态，响应变量的值就成为固定的。在题项到达关闭状态之后，递送系统或报告工具还可以允许呈现题项。这种呈现发生在回顾状态，如果响应处理发生并设定一个合适的结果变量，概要反馈此刻也可以是可见的。

最后，对于支持解答显示的系统，题项时间段可以进入解答状态。在此状态中，应试者的响应暂时被正确答案取代。下面是对题项时间段控制类的说明。

Class : itemSessionControl

相关类 testPart, sectionPart

说明：当题项作为测试的一部分时，测试可以在尝试次数和允许的状态上施加约束。这些约束可以具体到单独题项，整个节，或者整个 testPart 类。默认情况下，testPart 类级别的设定影响该部分的所有题项，除非设定在 assessmentSection 级别或最终在单独的 assessmentItemRef 中被覆盖。下面给定的默认值只用于没有任何加约束的情况。

属性名	多重性	类型	说明
maxAttempts	[0..1]	integer	对非自适应题项，maxAttempt 控制允许尝试的最大值，由于非自适应题项的计分规则对于每次尝试都是相同的，所以通常这个值为 1，该值为 0 表示没有限制。如果该值未指定，对于非自适应题项，值将被视为 1。对于自适应题项，maxAttempts 被忽略，由于尝试的次数被 completionStatus 内嵌结果变量的值所限制。在定义中 maxAttempts 的值大于 1，表明任何适用反馈必须被显示。这一点应用于适用的模式反馈和综合反馈。然而，一旦允许使用尝试的最大值，反馈是否显示就被 showFeedback 的约束所控制。
showFeedback	[0..1]	Boolean	影响最后一次尝试结束后反馈的可见性。如果值为假那么反馈不可见。即使应试者可以进入回顾状态，模式反馈和综合反馈也都不被显示。该值默认为假。
allowReview	[0..1]	Boolean	此约束只应用在最后一次尝试结束之后。如果被设定为真，允许题项时间段进入回顾状态，在此状态中，应试者可以回顾题项以及他们给出的响应，但是不能更新或重新提交。如果设定为假，一旦提交了最后一次的尝试，应试者不能回顾题项或他们的响应。默认值为真。 如果允许回顾状态，但是不允许反馈，递送系统必须注意不显示由作为回顾过程的一部分的最后一次尝试产生的综合反馈。然而，反馈可以采用将以前可见的素材隐藏的形式，以及将以前隐藏的材料显示的形式。

			<p>为了消除不确定性，对于非自适应的题项，在每次尝试开始时显示给应试者的题项的版本定义为没有反馈。换句话说，具有结果变量的默认值决定的任何综合反馈的可见性，而没有通过响应处理调用更新的结果变量的值。</p> <p>对于自适应题项，由于响应处理的迭代特性情况会更复杂，为了避免要求递送机制缓存结果变量的值，对于自适应题项，<code>showFeedback</code> 的设定在 <code>allowReview</code> 值为真的情况下应该被忽略。在回顾状态时，结果变量的最终值应该用于决定综合反馈的可见性。</p>
<code>showSolution</code>	[0..1]	Boolean	约束控制系统是否会提供给应试者解答状态。默认值为假。
<code>allowComment</code>	[0..1]	Boolean	约束控制是否允许应试者在时间段中对题项提供评论。一些递送系统支持获取应试者的评论。评论不是被评估响应的一部分，但是在评估过程中，可以让应试者向其他角色提供反馈。
<code>allowSkipping</code>	[0..1]	boolean	如果应试者没有提供任何响应，题项就被定义为允许跳过。换句话说，所有响应变量在提交时有默认值或为 <code>NULL</code> 。这个定义与在 <code>outcomeProcessing</code> 中 <code>numberResponded</code> 操作符一致。如果为假，禁止应试者跳过题项，或者换句话说，直到他们对至少一个响应变量提供一个非默认的值，才允许提交题项。根据定义，不能跳过没有响应变量的题项。此属性的值只有在题项属于一个具有单独提交模式的测试部分的情况下才适用。注意如果 <code>allowSkipping</code> 值为真，递送机制必须确保应试者可以选择无响应的提交，例如，通过提供“跳过”键。
<code>validateResponses</code>	[0..1]	Boolean	此属性在应试者提交非法响应时，控制递送机制的行为。非法响应定义为不满足与响应相关的交互施加的约束的响应。参见交互以获取更多信息。当 <code>validateResponses</code> 值为真，禁止应试者提交题项，直到他们对所有交互提供了合法的相应。当值为假时，非法响应可以被系统接受。这个属性的值只有在题项属于具有单独提交模式的 <code>testPart</code> 的情况下才适用。

题项变量是描述题项相关量的定义，与其他类的关系如下：

抽象类 variableDeclaration

派生类 outcomeDeclaration, responseDeclaration, templateDeclaration

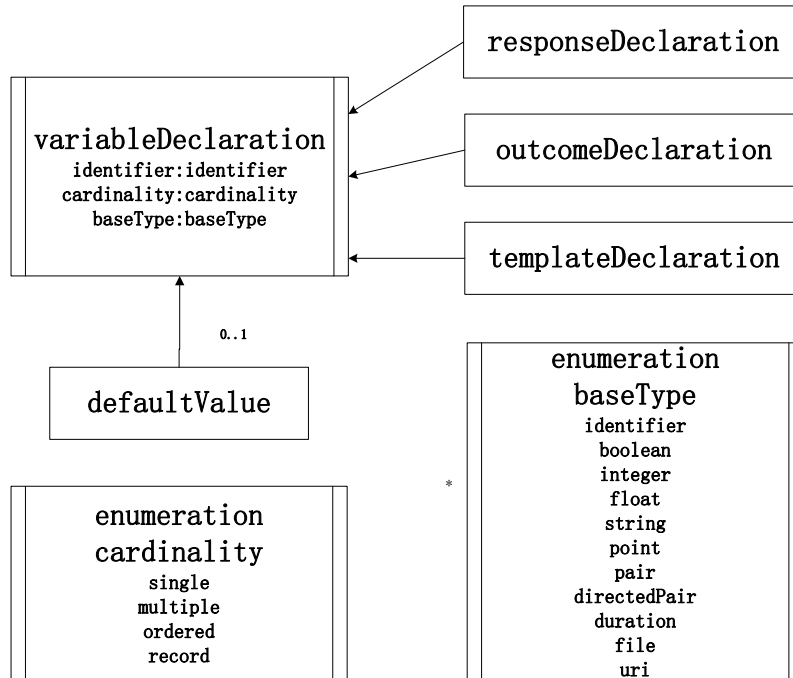


图4 变量定义

通过变量声明可定义题项变量。除了下面提到的内嵌时间段变量，被隐含声明外，所有的变量使用前必须声明。声明的目的是将标识符与变量联系起来，并且识别变量值的运行时类型。

题项变量可以没有任何值，在这种情况下，称它有特殊的值 NULL。例如，如果应试者没有响应交互，那么，任何相关响应变量都为 NULL。空的容器和空的字符串总是被视为 NULL 值。以下是变量定义类的说明。

Class: variableDeclaration

属性名	多重性	类型	说明
identifier	[1]	identifier	内嵌时间段变量的标识符是预留的。它们是 completionStatus, numAttempts 和 duration。一个题项里声明的所有题项变量共享同一个名字域。不同的题项有不同的名字域。
cardinality	[1]	cardinality	每个变量不是单值就是多值的。多值的变量被称为容器，并且以有序、无序和记录类型出现。参见多重性。
baseType	[0..1]	baseType	变量值从中选取的（或者在容器的情况下，单独的值从中选取的）值空间被识别为具有 baseType。baseType 选择预先定义的在运行时数据模型中被认为具有原子值的一组类型之一。具有记录多重性的变量没有基础类型。

包含:

名称	多重性
defaultValue	[0..1]

类 **Class** : value

相关类:

ordinaryStatistic, templateVariable, candidateResponse, correctResponse, outcomeVariable, defaultValue

在变量声明和结果报告中，能呈现任何 baseType 的单一值的类。基础类型被声明中的 baseType 属性定义，除了具有记录多重性变量的情况。

属性名	多重性	类型	说明
fieldIdentifier	[0..1]	identifier	本属性只用于对记录的一部分值指定域标识符
baseType	[0..1]	baseType	本属性只用于指定形成记录的一部分的值的基础类型。

类 **Class**: defaultValue

相关类: Associated classes, variableDeclaration

属性名	多重性	类型	说明
interpretation	[0..1]	string	默认值的说明

包含:

名称	多重性	值
cardinality	[0..1]	single,multiple,ordered,record

题项变量或其表达式既可以是单值的，也可以是多值的。多值的表达式（或变量）被称为容器。容器包含值的列表，列表可以是空的，在这种情况下被视为 NULL。所有多重的或者有序的容器中的值是从同一个集合中选取的，然而区别是容器可以包含同一个值的多次出现。换句话说，[A,B,B,C]对于容器来说是可接受的值。对于具有多值多重性的容器，值[A,B,C]和值[C,B,A]是相等的，然而这两个值对于具有有序多重性的容器被认为是不同的。当用作响应变量值的时候，区别由在多响应多选项任务的选择选项和有序对象任务的排序选项中的不同来代表。

记录容器类型是包含一系列由特有的标识符标识的，具有特有的基础类型的独立值的特殊的容器。本标准不直接利用记录容器类型，然而，可以使用定制交互操纵更复杂的响应，定制操作返回更复杂的值。

Enumeration: baseType

基础类型是一系列原子值的简单描述。用于定义运行时数据模型的基类型与用于定义规范本身属性值的基本数据类型具有等同的定义。用枚举定义运行时模型的基类型而不用类表示，是为了区别这两个不同级别的模型。

Identifier

标识符值的集合与 identifier 类定义的值集合相同。

Boolean

GB/Txxx—xxxx

boolean 值的集合与 **boolean** 类定义的值的集合相同。

Integer

integer 值的集合与 **integer** 类定义的值的集合相同。

Float

float 值的集合与 **float** 类定义的值的集合相同。

String

string 值的集合与 **string** 类定义的值的集合相同。

Point

点对应于一个绘图点。其包含的两个整数分别对应水平（x 轴）和垂直（y 轴）位置。轴的上/下和左/右的意义是与上下文相关的。

Pair

表示两个对象相联系的一对标识符。联系是无方向的，所以(A,B)和(B,A)是相等的。

directedPair

表示两个对象有向联系的一对标识符。联系是有方向的，两个标识符对应于源和目标对象。

Duration

指两个时间点之间的距离即时间段。**duration** 是以秒为单位测量的，可以有小数部分。

File

文件值是由文件的（例如，由应试者作为交互的一部分上传的）内容和可选的文件名组成。文件的内容类型是一种由[RFC2045]定义的 MIME 类型。

Uri

统一资源标识符

Class: mapping

相关类: **responseDeclaration**, **categorizedStatistic**

用于创建一个任何 **baseType**（除了 **file** 和 **duration**）的源集合到单精度浮点的映射的特殊类。考虑到匹配浮点值的难度，应该避免从基础类型的值到 **float** 的映射。参见匹配操作符部分。映射容器时，结果是目标集合映射的值的总和。参见 **mapResource** 以获得细节信息。

属性名	多重性	类型	说明
lowerBound	[0..1]	float	映射容器的结果的下界限。如果未指定，就没有下界限。
upperBound	[0..1]	float	映射容器的结果的上界限。如果未指定，就没有上界限。
defaultValue	[1]	float	当没有为源值给定具体的映射时，设定

使用的目标集合的默认值

包含:

名称	多重性	说明
mapEntry	[1..*]	映射是由一系列的 mapEntry 定义的, 每个 mapEntry 从源集合到一个单精度 float 映射一个单一的值。

Class : mapEntry

相关类 mapping

属性名	多重性	类型	说明
mapKey	[1]	valueType	源值
mappedValue	[1]	float	映射值

6.1 响应变量

Class : responseDeclaration

相关类 assessmentItem

响应变量由响应声明声明, 用于与题项体 itemBody 的交互。运行时, 响应变量被实例化为题项时间段的一部分。不管声明中是否给定了默认值, 它们的值总是被初始化为 NULL, 具有 NULL 值的响应变量表示应试者还没有提供一个响应, 或是根本尚未尝试题项, 或是尝试但选择不提供响应。如果为响应变量提供了默认值, 那么在第一次尝试开始的时候变量就设定为这个值。如果应试者从未尝试题项, 换句话说, 题项时间段直接从初始状态到关闭状态, 没有经过交互状态, 那么响应变量保持 NULL, 不使用默认值。当具有默认响应变量值的题项执行用户界面时, 递送机制的執行者应该小心。如果相关的交互在默认状态(即呈现默认值), 那么系统应确信应试者提交这个默认值。尤其是在应试者的尝试由于外部时间而结束的情况下, 更是如此。

包含

名称	多重性	说明
correctResponse	[0..1]	响应声明可以指派一个可选的 correctResponse。这个值可以指出响应变量的正确值。对于那些用比正确/错误更加复杂的刻度测量的响应, 这个值应该被设定为最佳值。最后, 对于没有定义这样最佳值的响应, correctResponse 必须被忽略。如果递送机制支持显示解答, 那么应该对应试者显示响应(在定义的地方)的正确值。正确值被显示的时候, 必须和应试者自己的响应有清晰的区别。
mapping	[0..1]	mapping 提供一个从基础值的集合到数字值的集合的映射。
areaMapping	[0..1]	areaMapping 只能用在有 baseType 点的变量声明中, 它为 mapping 提供一种替换形式, 可以测试坐标空间的区域代替映射单独的值(即单独的点)。

Class : correctResponse

相关类: responseDeclaration, responseVariable

属性名	多重性	类型	说明
interpretation	[0..1]	string	正确值的解释。

包含

名称	多重性	说明
value	[1..*]	

Class : areaMapping

相关类 responseDeclaration

用于创建从源点值集合到浮点值目标集合映射的特殊类。当映射容器时，结果是目标集合映射值的总和。参见 mapResponsePoint。属性与 mapping 中类似命名的属性意义相同。

属性名	多重性	类型	说明
lowerBound	[0..1]	float	
upperBound	[0..1]	float	
defaultValue	[1]	float=0	

包含

名称	多重性	说明
areaMapEntry	[1..*]	映射是由一系列 areaMapEntry 定义的，每个映射由一个坐标空间的区域到一个浮点值。当映射点时，每个区域按顺序被测试，覆盖交叉点上的点的区域有优先权。

Class : areaMapEntry

相关类 areaMapping

属性名	多重性	类型	说明
shape	[1]	shape	区域的形状。
coords	[1]	coords	区域的尺寸和位置，与形状关联。
mappedValue	[1]	float	映射的值

6.1.1 内嵌响应变量

内嵌响应变量有两个：numAttempts 和 duration，是隐含定义的，不能出现在响应定义中。

所有递送机制必须保持 numAttempts 的值。这是一个记录应试者对每个题项尝试次数的单精度整数。这个值最初默认为 0，然后在每次尝试开始时增加 1。

支持时间相关题项的系统必须同样保持 duration 的值。duration 含义是所有尝试记录的所有应试者时间段的总累计时间（以秒为单位）。它是一个单精度浮点数。换句话说，题项时间段从开始到结束的时间减去任何处与暂停状态的时间段。Duration 的分辨率必须小于 1s，应该是 0.1s 或更小。如果解析度用 epsilon 表示，那么每个 duration 的值代表 $duration \leq t < duration + \epsilon$ 的区间。即 duration 值被

截短了。时间无关题项的 `duration` 值不能用于任何题项或测试级别表达式，如果值已知，系统在 `itemResult` 中依然报告。

6.2 结果变量

Class : `outcomeDeclaration`

相关类 `assessmentTest`, `assessmentItem`

结果变量被结果声明所声明。值或者在声明中默认给定，或者在响应处理中被响应规则给定。运行时，结果变量被实例化为题项时间段的一部分。在响应处理中，其值可以被一个默认值和/或集合初始化。如果在声明中没有给定默认值，那么结果变量被初始化为 `NULL`，除非结果是数字类型 (`integer` 或 `float`)，在这种情况下它被初始化为 `0`。对于非自适应题项，在每次响应处理进行前，结果变量的值被重新设定为默认值。对于自适应题项，结果变量保持前一次响应处理调用时的值。参见结果处理。

属性名	多重性	类型	说明
<code>view</code>	[*]	<code>view</code>	<code>view</code> 属性用来设定结果变量的查看者。如果 <code>view</code> 没有指定，结果变量视为与所有查看有关。复杂题项，例如自适应题项或复杂模板，可能声明应试者根本不感兴趣的结果，仅仅用于在题项或测试时间段保持中间值或其他有用信息。这样的变量声明时应该有开发者（对于题项结果）的查看或 <code>testConstructor</code> （对于测试结果）。如果合适，根据它们声明的 <code>view</code> ，系统可以从结果报告中排除结果变量值的解释。
<code>interpretation</code>	[0..1]	<code>String</code>	结果变量值解释的扩展。
<code>longInterpretation</code>	[0..1]	<code>uri</code>	<code>normalMaximum</code> 属性定义结果变量的最大值。必须是正值。如果给定，结果变量的值可以被 <code>normalMaximum</code> 除，然后缩小以获得一个在[-1.0,1.0]区间的规格化的分数。 <code>normalMaximum</code> 对 <code>responseProcessing</code> 或结果变量自身可以取的值没有影响。
<code>normalMaximum</code>	[0..1]	<code>float</code>	定义数字结果变量的最小值，可以是负的
<code>normalMinimum</code>	[0..1]	<code>float</code>	对数字结果变量定义一个值，这个值表示应试者掌握了被测试的某个方面。
<code>masteryValue</code>	[0..1]	<code>float</code>	

包含类 `lookupTable`[0..1]

抽象类 `lookupTable`

派生类 `interpolationTable`, `matchTable`

相关类 `outcomeDeclaration`

该类是与结果变量定义相联系的抽象类，用于创建一个查找表。查找表定义了从数字原值到单个结果变量值的对应关系。查找表与类似映射的相反的意义工作，它定义源数字值如何被转换为目标数字值，然

GB/T×××—××××

而（响应）映射定义响应值如何被映射到目标数字值。转换发生时使用响应处理或结果处理中的查找结果值规则。

属性名	多重性	类型	说明
defaultValue	[0..1]	valueType	没有找到匹配表格条目时要使用的默认的结果值。如果忽略，则使用 NULL 值。

Class : matchTable

matchTable通过查找第一个与源信息精确匹配的matchTableEntry来转换源整数

包含: matchTableEntry [1..*]

Class : matchTableEntry

相关类: matchTable

属性名	多重性	类型	说明
sourceValue	[1]	integer	必须精确匹配的源整数。
targetValue	[1]	valueType	找到一个匹配时，用于设定结果的目标值。

类 **Class :** interpolationTable

interpolationTable 通过查找第一个源值小于或等于源值的条目来转换源浮点数。例如，插补表可以用于映射未处理的分数到表示成绩的标识符。也可用于实现数字转换，例如从简单的数据到校准刻度。

包含 interpolationTableEntry [1..*]

类 **Class :** interpolationTableEntry

相关类 interpolationTable

属性名	多重性	类型	说明
sourceValue	[1]	float	源值匹配条目的下界限。
includeBoundary	[0..1]	boolean	决定 sourceValue 与条目是否精确匹配。如果为默认值真，那么值的精确匹配认为是表项的精确匹配。
targetValue	[1]	valueType	当找到一个匹配时，用于设定结果的目标值

6.2.1 内嵌结果变量

内嵌结果变量有一个，为 completionStatus，是隐含声明，不能出现在结果变量定义中。递送机制必须保持内嵌结果变量 completionStatus 的值，它是一个单一的标识符。开始于一个预留值“not_attempt”。第一次尝试开始时，变为预留值“unknown”，在题项时间段内保持这个值，除非被响应处理的设置结果变量规则设定为一个不同的值。有四种允许的值：completed、incomplete、not_attempted 和 unknown。在响应处理过程中，这些值中的任何一个可以被设定，如果自适应题项设定 completionStatus 为 completed，那么时间段必须被置于关闭状态，然而题项时间段并未被要求在终结之前等到 completed 信号，通过消耗完时间或一些其它情况，它可以响应应试者的直接要求而终结。自适应题项必须保持合适的值，并且设定 completionStatus 为“completed”以指出交互、响应处理和反馈

的周期必须停止。不要求非自适应题项为 `completionStatus` 设定值，但是可以这样

7 内容模型

Class : `itemBody` (`bodyElement`)

相关类: `assessmentItem`

包含:

名称	多重性	说明
<code>block</code>	[*]	

题项体中包含文本、图像、媒体对象，以及描述题项内容和题项组织的交互。题项体与样式表联合呈现，明确或隐含的使用递送或开发系统的默认样式规则。

当题项时间段处在交互状态时，其题项体必须呈现给应试者。在这种状态下，应试者必须能够与每一个可见的交互进行操作，从而设定或更新相关的响应变量的值。当题项时间段处在关闭或回顾状态时，题项体可能会呈现给应试者，在这些状态中，尽管应试者的响应应该是可见的，但交互必须被禁用，以防止应试者设定或更新相关响应变量的值。最后，题项体可能在解答状态中呈现给应试者，此时响应变量的正确值必须是可见的，而相关的交互是被禁用的。

本标准使用的内容模型采用了很多[XHTML]的概念。实际上，这部分标准定义了一个 XHTML 的配置文件。在题项中，只有一些 XHTML 定义的元素是允许使用的，其中，有些元素的属性还有额外约束。可以参见 XHTML Elements 以获取更多信息。最后，本标准定义一些新的元素，用于呈现交互操作、控制反馈和限制内容的显示。

抽象类: `bodyElement`

派生类: `atomicBlock`, `atomicInline`, `caption`, `choice`, `col`, `colgroup`, `div`, `dl`, `dlElement`, `hr`, `interaction`, `itemBody`, `li`, `object`, `ol`, `printedVariable`, `prompt`, `simpleBlock`, `simpleInline`, `table`, `tableCell`, `tbody`, `templateElement`, `tfoot`, `thead`, `tr`, `ul`

题项内容模型的所有内容对象的根类都是 `bodyElement`。它定义了一些对内容模型所有元素都共有的属性。

属性名	多重性	类型	说明
<code>id</code>	[0..1]	<code>identifier</code>	<code>id</code> 在题项里必须是唯一的。
<code>class</code>	[*]	<code>styleclass</code>	可以把类赋值给单独的题项体元素。可以给定多个类名。这些类名识别元素作为列出的类的一个成员。类的成员可以被开发系统用于区分本标准中没有区分的内容模型。通常该信息可以用于在相关的样式表中基于定义的格式化操作。
<code>lang</code>	[0..1]	<code>language</code>	元素的主要语言。本属性是可选的，通常从封装的元素中继承。
<code>label</code>	[0..1]	<code>string256</code>	<code>label</code> 属性利用应用程序提供的数据为开发系统提供了标注内容模型元素的机制。如果题项使用了 <code>label</code> ，那么相关的 <code>toolName</code> 和 <code>toolVersion</code> 的值也必须提供。

7.1 基本类

有一些抽象类支撑内容模型，用于将题项体中的元素分组定义到对等组的类别中。

抽象类： bodyElement

派生类： flow, param

相关类： object

可以出现在对象中的元素。

抽象类： inline

派生类： inlineInteraction, inlineStatic

相关类： simpleInline, dt, caption, atomicBlock

表现为文本片断的元素，例如段落的内容。

抽象类： block

派生类： blockInteraction, blockStatic, customInteraction, positionObjectStage

相关类： itemBody, simpleBlock

提供文本结构的元素，例如段落、表格等。大多数元素是内联或块元素之一。

抽象类： flow (objectFlow)

派生类： blockInteraction, customInteraction, flowStatic, include, inlineInteraction

相关类： tableCell, div, dd, li

可以出现在列表题项，表格单元等位置内部的元素。包括块类型和内联类型元素。

属性名	多重性	类型	说明
base	[0..1]	uri	用于改变本对象范围内的解析相关 URI 的基址的可选的 URI。当解析相关的 URI 作为包含于题项片断一部分时，需要特别注意。参见题项和测试片断以获得更多信息

抽象类： inlineStatic (inline)

派生类： atomicInline, gap, hottext, math, object, printedVariable, simpleInline, templateInline, textRun

相关类： hottext, prompt, templateInline

inline类的一个子类，其中不包含交互。

抽象类： blockStatic (block)

派生类： atomicBlock, div, dl, hr, math, ol, simpleBlock, table, templateBlock, ul

相关类： templateBlock, gapMatchInteraction, hottextInteraction

block类的一个子类，其中不包含交互。

抽象类: flowStatic (flow)

派生类: atomicBlock, atomicInline, div, dl, hottext, hr, math, object, ol, printedVariable, simpleBlock, simpleInline, table, templateBlock, templateInline, textRun, ul

相关类: simpleAssociableChoice, testFeedback, modalFeedback, simpleChoice

flow类的一个子类，其不包含交互。

下面的类定义了少量的XHTML使用的通用元素类型。

抽象类: simpleInline (bodyElement, flowStatic, inlineStatic)

派生类: a, abbr, acronym, b, big, cite, code, dfn, em, feedbackInline, i, kbd, q, samp, small, span, strong, sub, sup, tt, var

包含:

名称	多重性	说明
inline	[*]	

抽象类: simpleBlock (blockStatic, bodyElement, flowStatic)

派生类: blockquote, feedbackBlock, rubricBlock

包含:

名称	多重性	说明
block	[*]	

抽象类: atomicInline (bodyElement, flowStatic, inlineStatic)

派生类: br, img

抽象类: atomicBlock (blockStatic, bodyElement, flowStatic)

派生类: address, h1, h2, h3, h4, h5, h6, p, pre

包含:

名称	多重性	说明
inline	[*]	

Class : textRun (flowStatic, inlineStatic, textOrVariable)

文本的执行就是对字符的执行操作。不像内容模型中其他元素，它不是bodyElement的子类。为了给文本的执行分配属性，必须改用span元素

7.2 XHTML 元素

从[XHTML]提取的内容模型结构元素根据在[XHTML_MOD]中的分类而按组归档。只有在这里列出的属性可以使用（包括从父类继承的属性）。默认情况下，元素和属性与相应的[XHTML]的元素和属性有相同的解释和限制。

7.2.1 文本元素

GB/T×××—××××

Class : abbr (simpleInline)

注意：不支持XHTML定义的标题属性。

Class : acronym (simpleInline)

注意：不支持XHTML定义的标题属性。

Class : address (atomicBlock)

Class : blockquote (simpleBlock)

属性名	多重性	类型	说明
cite	[0..1]	uri	

Class : br (atomicInline)

Class : cite (simpleInline)

Class : code (simpleInline)

Class : dfn (simpleInline)

Class : div (blockStatic, bodyElement, flowStatic)

包含：

名称	多重性	说明
flow	[*]	

Class : em (simpleInline)

Class : h1 (atomicBlock)

Class : h2 (atomicBlock)

Class : h3 (atomicBlock)

Class : h4 (atomicBlock)

Class : h5 (atomicBlock)

Class : h6 (atomicBlock)

Class : kbd (simpleInline)

Class : p (atomicBlock)

Class : pre (atomicBlock)

虽然pre继承atomicBlock，必须不能包含任何下列对象：img、object、big、small、sub、sup，无论是直接或间接的。

Class : q (simpleInline)

属性名	多重性	类型	说明
-----	-----	----	----

cite	[0..1]	uri
------	--------	-----

Class : samp (simpleInline)

Class : span (simpleInline)

Class : strong (simpleInline)

Class : var (simpleInline)

7.2.2 列表元素

Class : dl (blockStatic, bodyElement, flowStatic)

包含:

名称	多重性	说明
dlElement	[*]	

虚类 : dlElement (bodyElement)

派生类: dd, dt

相关类: dl

Class : dt (dlElement)

包含:

名称	多重性	说明
inline	[*]	

Class : dd (dlElement)

包含:

名称	多重性	说明
flow	[*]	

Class : ol (blockStatic, bodyElement, flowStatic)

包含:

名称	多重性	说明
li	[*]	

Class : ul (blockStatic, bodyElement, flowStatic)

包含:

名称	多重性	说明
li	[*]	

Class : li (bodyElement)

相关类：ul, ol

包含：

名称	多重性	说明
flow	[*]	

7.2.3 对象元素

Class : object (bodyElement, flowStatic, inlineStatic)

相关类：positionObjectStage, graphicInteraction, positionObjectInteraction, mediaInteraction, drawingInteraction, gapImg.

包含：

名称	多重性	说明
objectFlow	[*]	

属性名	多重性	类型	说明
data	[1]	string	数据属性用于定位对象相关数据的 URI
type	[1]	contentType	
width	[0..1]	length	
height	[0..1]	length	

Class : param (objectFlow)

属性名	多重性	类型	说明
name	[1]	string	参数名，与对象解释的相同。
value	[1]	string	为命名的参数传递对象的值。这个值受限于模板变量扩展。如果值是 paramVariable 为真的模版变量的名字，那么模版变量的 value 作为给定参数的值传给对象。 当以参数值扩展模版变量时，除了 identifier、string 和 uri 之外的类型必须被转换为 string。数字类型用合适的 "%i" 或 "%G" 格式转换为 string（参见 printedVariable 以获得数字格式的讨论）。基础类型 boolean 被扩展为 string 类型的 "true" 或 "false" 之一。基础类型 point 被用 "%i" 格式按照横纵坐标的顺序扩展为用两个空格间隔的整数。基础类型 pair 和 directPair 的值被转换为包含由空格间隔的两个标识符的 string。基础类型 duration 的值被用 "%G" 格式转换。基础类型 file 不能用于属性扩展。

			如果 valueType 是 REF，模版变量必须是基础类型 uri 的。
valueType	[0..1]	paramType = DATA	本标准支持使用 DATA 和 REF，但不支持 OBJECT。
type	[0..1]	mimeType	用于为 value 以及 aluetypeREF 提供类型。

Enumeration: paramType

DATA

REF

7.2.4 显示元素

Class : b (simpleInline)

Class : big (simpleInline)

Class : hr (blockStatic, bodyElement, flowStatic)

Class : i (simpleInline)

Class : small (simpleInline)

Class : sub (simpleInline)

Class : sup (simpleInline)

Class : tt (simpleInline)

7.2.5 表格元素

Class : caption (bodyElement)

相关类: table

包含:

名称	多重性	说明
inline	[*]	

Class : col(bodyElement)

相关类: table, colgroup

属性名	多重性	类型	说明
span	[0..1]	integer=1	

Class : colgroup(bodyElement)

相关类: table

属性名	多重性	类型	说明
span	[0..1]	integer=1	

包含:

名称	多重性	说明
col	[*]	

Class : table(blockStatic, bodyElement, flowStatic)

属性名	多重性	类型	说明
summary	[0..1]	string	

包含:

名称	多重性	说明
caption	[0..1]	
col	[*]	如果表格直接包含col, 那么不能包含任何colgroup元素。
colgroup	[*]	如果表格包含colgroup, 那么不能直接包含任何col元素。
thead	[0..1]	
tfoot	[0..1]	
tbody	[1..*]	

虚类 : tableCell (bodyElement)

派生类: td, th

相关类: tr

在 XHTML 中, 表格单元由 th 或 td 呈现, 共享下列的属性和内容模型:

属性名	多重性	类型	说明
headers	[*]	identifier	
scope	[0..1]	tableCellScope	
abbr	[0..1]	string	
axis	[0..1]	string	
rowspan	[0..1]	integer	
colspan	[0..1]	integer	

包含:

名称	多重性	说明
flow	[*]	

Enumeration: tableCellScope

row

col

rowgroup

colgroup

Class : tbody (bodyElement)

相关类: table

包含:

名称	多重性	说明
tr	[1..*]	

Class : td (tableCell)

Class : tfoot (bodyElement)

相关类: table

包含:

名称	多重性	说明
tr	[1..*]	

Class : th (tableCell)

Class : thead (bodyElement)

相关类: table

包含:

名称	多重性	说明
tr	[1..*]	

Class : tr (bodyElement)

相关类: thead, tbody, tfoot

包含:

名称	多重性	说明
tableCell	[1..*]	

7.2.6 图像元素

Class : img (atomicInline)

属性名	多重性	类型	说明
src	[1]	uri	
alt	[1]	string256	
longdesc	[0..1]	uri	
height	[0..1]	length	
width	[0..1]	length	

7.2.7 图像元素

Class : a (simpleInline)

虽然a从simpleInline继承，但不能直接或间接包含另一个a。

属性名	多重性	类型	说明
href	[1]	uri	
type	[0..1]	mimeType	

7.3 MathML

MathML 定义一种用 XML 描述数学符号的标记语言。MathML 的主要目的是为在其它文档，特别是 HTML 文档中嵌入数学表达式提供一种语言。

Class : math (blockStatic, flowStatic, inlineStatic)

math 类是由 MathML 规范外部定义的。可以在题项内容里表现为 inline、block 或 flow 元素。

7.3.1 联合模板变量和 MathML

创建题项模版时，经常需要改变数学表达式的元素。虽然不能直接嵌入在数学对象中定义的 printedVariable 元素，本节描述的技术可以达到类似效果。

在 MathML 中，通过 <mn> 或 <cn> 来实现数字的呈现或者内容呈现。类似的有 <mi> 和 <ci> 标识符。如果在模版变量声明中设定 mathVariable，那么所有匹配模版变量名称的 <mi> 和 <ci> 的实例必须分别用模版变量作为它们的内容，被 <mn> 和 <cn> 替换。

将来这种扩展模版变量的技术可能会被应用到 MathML 的其它元素。

7.4 变量内容

本标准定义了两种 assessmentItem 的内容随题项时间段的状态变化而改变的方法。第一种方法是基于结果变量的值。

虚类 : feedbackElement

派生类: feedbackBlock, feedbackInline

属性名	多重性	类型	说明
outcomeIdentifier	[1]	identifier	结果变量的标识符必须有一个 identifier 的基础类型，而且其 cardinality 需要是 single 或者是 multiple。通过在响应处理中赋值给这个结果变量来控制反馈元素的可见性。
showHide	[1]	showHide = show	showHide 属性决定如何控制 feedbackElement 的可见性。如果设定为 show，那么反馈默认被隐藏，只有在相关的结果变量匹配或包含 identifier 属性的值时才显示。如果设定为 hide，那么反馈默认被显示，当相关结果变量匹配或包含 identifier 属性的值时隐藏。
identifier	[1]	identifier	与 showHide 属性联合决定反馈的可见性的标识符。

形成非自适应题项的一部分的反馈元素不能直接或间接的包含交互对象。当交互包含在隐藏反馈元素中时，必须被隐藏。应试者不能设定或更新相应变量的值。

Enumeration: showHide

show

hide

Class : feedbackBlock (feedbackElement, simpleBlock)

Class : feedbackInline (feedbackElement, simpleInline)

Class : rubricBlock(simpleBlock)

相关类: assessmentSection

属性名	多重性	类型	说明
view	[1..*]	view	题目块内容的视图。 题目块内容等同于对一个或多个查看题项的角色呈现说明的评估题项的题项体的一部分。虽然题目块被定义为 simpleBlock，但不能包含交互。

嵌套的 bodyElement 或 rubricBlock 的可见性由最外面的元素决定。换句话说，如果元素被决定隐藏，那么它的所有内容也被隐藏，包括有条件可见的元素。

Class : printedVariable (bodyElement, flowStatic, inlineStatic, textOrVariable)

属性名	多重性	类型	说明
identifier	[1]	identifier	结果变量以及模板变量必须已定义，且其 cardinality 为 single。响应变量的值不能直接显示出来，如果必要，他们的值可以在响应处理中赋值，作为只在适当的反馈状态中可见的 bodyElement 的一部分并且显示给应试者。 如果变量值是 NULL，那么元素被忽略。 基础类型 string 的变量被视为简单文本。 基础类型 integer 或 float 的变量用下面描述的格式规则转换为文本 (string)。float 变量应该只能按照 e, E, f, g, G, r 或 R 的方式进行格式化。 基础类型 duration 被视为 float，代表以秒为单位的持续时间。
format	[0..1]	string256	当转换数字值为字符串时使用的格式转换说明。参见 7.4.1 数字格式化规则。

base	[0..1]	integer=10	用 i 转换类型码转换整数变量为字符串时使用的进制。 基础类型 file 的变量用来授权用户打开文件。如果存在的话，控制应该显示与文件相关的名字。 基础类型 uri 的变量用来授权用户打开确定的资源，例如，通过 URL 情况下的超文本链接。
------	--------	------------	--

虚类 : textOrVariable

派生类: printedVariable, textRun

相关类: gapText

在只允许纯文本的环境中，例如inlineChoice中的文本，用于授权变量值替换的抽象类。

7.4.1 数字格式化规则

format属性的语法是基于定义在C programming language [ISO_9899]中的格式化转换说明中的，与printf和相关函数一起使用。

每个转换说明开始于‘%’字符，跟随0个或多个标志字符(#, 0, -, " " [space] 和+)，可选数字字符串指示域宽度的最小值、可选精度（包括“.”和随后的0个或多个数字）和最终转换类型码中的一个：E, e, f, G, g, r, R, i, o, X, 或x。除了i之外，这些都可由C语言的标准解释，其中i可以利用base属性格式化非十进制数字，如果要求的数字的精度位数小于数字本身的小数位数，除了科学计数意外，r/R和g/G一样都可以利用由精度给出的有效数字进行四舍五入。

7.5 用 Stylesheet 格式化题项

Class : stylesheet

相关类: assessmentItem

用于将外部样式表与assessmentItem相联系。

属性名	多重性	类型	说明
href	[1]	uri	外部样式表的标识符或位置。
type	[1]	contentType	外部样式表的类型
media	[0..1]	string	可选的描述样式表应用于媒体的描述符。
title	[0..1]	string	可选的样式表标题

8 交互

抽象类 variableDeclaration

派生类 outcomeDeclaration, responseDeclaration, templateDeclaration

交互允许应试者与题项交互。通过交互，应试者选择或创建一个响应。应试者的响应储存在响应变量里。每个交互（至少）与一个响应变量相关。

属性名	多重性	类型	说明
responseIdentifier	[1]	identifier	与交互相关的响应变量。

交互的状态用于设定相关响应变量的值。相关响应变量的声明将响应的值限定为特定的基础类型和基值。一些交互对允许的响应集合施加额外约束，例如，通过约束交互中最少和/或最多可选择的选项数量。在一些交互递送机制中，可能在应试者与题项交互时检查这些约束。当应试者浏览测试时，他们会看见关于哪些题项有合法的响应以及哪些需要注意的指示，或者应试者可能被阻止在测试中前进，直到选择/创建一个合法的响应。（参见validateResponse以获得在测试中如何执行本操作模式。）在有些情况下，递送机制甚至会提供消除某些非法响应种类的交互控制，例如，通过限制可选择的选项数量来防止超出交互指定的最大值。

由于应试者可能将交互置于不满足这些附加约束的状态，题项开发者必须确保响应处理规则适当处理响应变量的非法值。

抽象类 inlineInteraction (flow, inline, interaction)

派生类 endAttemptInteraction, inlineChoiceInteraction, textEntryInteraction

在线交互

抽象类 blockInteraction (block, flow, interaction)

派生类 associateInteraction, choiceInteraction, drawingInteraction, extendedTextInteraction, gapMatchInteraction, graphicInteraction, hottextInteraction, matchInteraction, mediaInteraction, orderInteraction, sliderInteraction, uploadInteraction

表现为内容模型块的交互。大多数交互是这种类型的。

包含：

名称	多重性	说明
prompt	[0..1]	可选的对于交互的提示。

类 prompt (bodyElement)

相关类 blockInteraction

包含：

名称	多重性	说明
inlineStatic	[*]	提示不能包含任何嵌套交互。

抽象类 choice (bodyElement)

相关类 associableChoice, hotspotChoice, hottext, inlineChoice, simpleChoice

很多交互包括一个或多个预定义的选项。这些选项通常具有下列共同属性：

属性名	多重性	类型	说明
identifier	[1]	identifier	选项的标识符。本标识符不能用于任何其他选项或题项变量。
fixed	[0..1]	boolean = false	如果一个选项的此属性固定为真，即使支持选项重置的立即装入交互，在交互

			中选项的位置也不能被递送机制改变，如果没有指定值，那么选项可以被自由排列。 在题项模版中，选项的可见性可以通过模版处理过程中设置相关模版变量的值来控制。获取题项模版的信息，参见题项模板
templateIdentifier	[0..1]	identifier	模版变量的标识符，必须是identifier基础类型的，可以有单值或多值的基数。当相关的交互是题项模版的一部分时，标识的模版变量的值用于控制选项的可见性。当选项被隐藏，则不可选，除非有其他规定，否则选项的内容对应试者是不可见的。
showHide	[0..1]	showHide = show	showHide 属性决定选项的可见性如何控制。如果设定为显示，那么选项被默认隐藏，只有在相关模版变量匹配或包含选项的标识符时，才被显示。如果设定为隐藏，那么选项被默认显示，只有在相关模版变量匹配或包含选项的标识符时，才被隐藏。

抽象类 associableChoice (choice)

派生类 associableHotspot, gap, gapChoice, simpleAssociableChoice

其他交互包括预先设定的相关选项对。这些选项都有下列共同属性：

属性名	多重性	类型	说明
matchGroup	[0..*]	identifier	本选项可能匹配的相关选项集合，排除所有其他的。如果 matchGroup 没有给定，或者是空的，那么所有其他选项可以和这个选项相关联，并服从它们自己的匹配约束。

8.1 简单交互

类 choiceInteraction (blockInteraction)

选择交互向应试者呈现一个选项集。应试者的任务是选择一个或多个选项，最多可选maxChoices个，没有相应的最小选择数量。交互一般初始化为没有选项被选择。

choiceInteraction必须受限一个响应变量，该响应变量是identifier基础类型的，并且有单值或多值的基数。

属性名	多重性	类型	说明
shuffle	[1]	boolean = false	如果 shuffle 属性为真，那么递送机制必须随机决定选项呈现的顺序，但该属性

			受限于 <code>fixed</code> 属性。
<code>maxChoices</code>	[1]	<code>integer = 1</code>	应试者允许选择的选项的最大数量。如果 <code>maxChoices</code> 是 0，那么没有限制。如果 <code>maxChoices</code> 比 1 大，那么交互必须受限于一个具有多值基数的响应。
<code>minChoices</code>	[0..1]	<code>integer = 0</code>	要求应试者选择可以形成合法响应的选项的最小数量。如果 <code>minChoices</code> 是 0，那么不要求应试者选择任何选项。 <code>minChoices</code> 必须小于或等于 <code>maxChoices</code> 施加的限制。

包含：

名称	多重性	说明
<code>simpleChoice</code>	[1..*]	显示给用户的有序的选项列表。顺序就是呈现给用户的顺序，除非 <code>shuffle</code> 属性值为真。

类 `orderInteraction` (`blockInteraction`)

在排序交互中，应试者的任务是重新将选项排序，选项初始显示的顺序是有意义的。默认情况下，应试者的任务是将所有的选项排序，但是可以用 `maxChoice` 和 `minChoice` 来请求选项的子集。在指定的情况下，应试者必须选择一个选项的子集并且对它们进行排序。

如果一个与排序交互相关的响应变量的默认值已被指定，那么它的值应该用于覆盖这里指定的选项的顺序。

根据特性，排序交互难以呈现为未回答状态，尤其是在默认情况下，所有选项都要被排序。执行者应该意识到在响应变量一节中描述的关于默认值的使用的问题。

`orderInteraction` 必须受限于一个响应变量，该响应变量是 `identifier` 基础类型的，并且只有有序的基数。

包含：

名称	多重性	说明
<code>simpleChoice</code>	[1..*]	显示给用户的有序选项列表。列表的顺序是呈现给用户的选项初始顺序，除非 <code>shuffle</code> 属性值为真。

属性名	多重性	类型	说明
<code>shuffle</code>	[1]	<code>boolean = false</code>	如果 <code>shuffle</code> 属性为真，那么递送机制必须随机决定选项呈现的顺序，但该属性受限于 <code>fixed</code> 属性。

maxChoices	[0..1]	integer	应试者响应交互时允许选择和排序的选项的最大数量。与minChoice联合使用，如果被指定，maxChoices必须大于或等于minChoices，而且不能超过提供的选项数量。如果未指定，所有选项可以被排序。
minChoices	[0..1]	integer	要求应试者选择和排序用以形成交互合法响应的选项的最小数量。如果指定minChoices，那么必须是1或者更大的数字，但是不能超过提供的选项数量。如果未指定，所有选项必须是有序的，并且maxChoice被忽略。
orientation	[0..1]	orientation	orientation属性提供给显示系统，排序有固有的垂直或水平解释的提示。

类 simpleChoice (choice)

相关类 orderInteraction, choiceInteraction

包含：

名称	多重性	说明
flowStatic	[*]	simpleChoice 是包含 flowStatic 对象的选项。simpleChoice 不能包含任何嵌套交互。

类 associateInteraction (blockInteraction)

associateInteraction类是一个呈现给应试者一些选项并且允许应试者在选项间创建联系的blockInteraction类。

associateInteraction必须受限于一个响应变量，该响应变量是pair基础类型的，并且有单值或多值的基数。

属性名	多重性	类型	说明
shuffle	[1]	boolean = false	如果 shuffle 属性为真，那么递送机制必须随机决定选项呈现的顺序，但该属性受限于选项的 fixed 属性。
maxAssociations	[1]	integer = 1	允许应试者构造联系的最大数值。如果 maxAssociation是0，那么没有限制。如果maxAssociation大于1（或0），那么交互必须受限于一个具有多值基数的响应。
minAssociations	[0..1]	integer = 0	应试者为形成合法响应而需要构造的联系的最小数。如果 minAssociations 为 0，那么不要求应试者构造任何联系。minAssociation 必须小于或等于

maxAssociation 施加的限制。

包含：

名称	多重性	说明
simpleAssociableChoice	[1..*]	一个有序的选项集

类 matchInteraction (blockInteraction)

matchInteraction类是呈现给应试者两个选项集合并且允许在两个集合，而不是在同一个集合的选项对之间创建联系的BlockInteraction类。

matchInteraction必须受限于一个响应变量，该响应变量是directedPair基础类型的，并且有单值或多值的基数。

属性名	多重性	类型	说明
shuffle	[1]	boolean = false	如果 shuffle 属性为真，那么递送机制必须随机决定选项呈现的顺序，但该属性受限于是选项自身的 fixed 属性。
maxAssociations	[1]	integer = 1	允许应试者构造联系的最大数值。如果 maxAssociation是0，那么没有限制。如果maxAssociation大于1（或0），那么交互必须受限于是具有多值基数的响应。
minAssociations	[0..1]	integer = 0	应试者为形成合法响应而需要构造的联系的最小数。如果 minAssociations 为 0，那么不要求应试者构造任何联系。minAssociation 必须小于或等于 maxAssociation 施加的限制。

包含：

名称	多重性	说明
simpleMatchSet	[2]	两个选项集，第一个集合定义了源选项，第二个集合定义了目的选项。

类 simpleAssociableChoice (associableChoice)

相关类 associateInteraction, simpleMatchSet

属性名	多重性	类型	说明
matchMax	[1]	integer	本选项表示匹配选项的最大数量。如果 matchMax是0，那么没有限制。
matchMin	[0..1]	integer = 0	为了构成合法响应，本选项必须匹配的选项最小数目。如果 matchMin 是 0，那

么不要求应试者将本选项与其它的相联系。`matchMin` 必须小于或等于 `matchMax` 施加的限制。

包含：

名称	多重性	说明
<code>flowStatic</code>	[*]	<code>associableChoice</code> 是包含 <code>flowStatic</code> 对象的选项，不能包含嵌套交互。

类 `simpleMatchSet`

相关类 `matchInteraction`

包含：

名称	多重性	说明
<code>simpleAssociableChoice</code>	[*]	关于集合的有序选项集。

类 `gapMatchInteraction (blockInteraction)`

`gapMatchInteraction` 类是一个包含很多空白的 `blockInteraction` 类，应试者可以从相关的选项集合填充这些空白。应试者必须能回顾选项指出的上下文填充的内容。

`gapMatchInteraction` 必须受限一个响应变量，该响应变量是 `directedPair` 基础类型的，并且可以有单值或多值的基数，这取决于空白的数量。选项显示了用于配对的源和目标空白。每个空白可以有至多一个选项与之关联。选项的最大出现次数由 `gapChoice` 的 `matchMax` 属性控制。

属性名	多重性	类型	说明
<code>shuffle</code>	[1]	<code>boolean = false</code>	如果 <code>shuffle</code> 属性为真，那么递送机制必须随机决定选项（不是空白）呈现的顺序，但该属性受限选项自身的 <code>fixed</code> 属性。

包含：

名称	多重性	说明
<code>gapChoice</code>	[1..*]	为了填充空白的有序的选项列表。如果要求，选项可以比空白更少。
<code>blockStatic</code>	[1..*]	交互的内容仅仅是一篇包含空白的内容。如果块包含多于一个的空白，那么交互必须受限一个具有多值基数的响应。

类 `gap` (`associableChoice`, `inlineStatic`)

`gap`是只能出现于`gapMatchInteraction`内的内联静态元素。

属性名	多重性	类型	说明
<code>required</code>	[0..1]	<code>boolean = false</code>	如果为真，那么应试者必须填充这个空白，从而构成交互的一个合法响应。

抽象类 `gapChoice` (`associableChoice`)

派生类 `gapImg`, `gapText`

相关类 `gapMatchInteraction`

用于填充`gapMatchInteraction`中空白的选项，或为简单的文本的顺串，或为简单图像对象，二者都来源于`gapChoice`。

属性名	多重性	类型	说明
<code>matchMax</code>	[1]	<code>integer</code>	可以与本选项关联的选项的最大数量。如果 <code>matchMax</code> 值为0，那么没有限制。
<code>matchMin</code>	[0..1]	<code>integer = 0</code>	与本选项关联，以构成合法响应的最小空白数量。如果 <code>matchMin</code> 是 0，那么不要求应试者将本选项与任何空白联系。 <code>matchMin</code> 必须小于或等于 <code>matchMax</code> 施加的限制。

类 `gapText` (`gapChoice`)

用户要插入于空白处的简单文本串，可以用`printVariable`进行变量值替换。

包含：

名称	多重性	说明
<code>textOrVariable</code>	[*]	

类 `gapImg` (`gapChoice`)

相关类 `graphicGapMatchInteraction`

包含单一图像对象的空白对象，应试者可以在空白对象中插入一个空白。

属性名	多重性	类型	说明
<code>objectLabel</code>	[0..1]	<code>string</code>	将要插入的图像对象的可选标签。

包含：

名称	多重性	说明
<code>object</code>	[1]	

8.2 基于文本的交互

类 inlineChoiceInteraction (inlineInteraction)

内联选项是一个inlineInteraction，向用户呈现一个选项集，该集中的每个选项都是一个简单的文本块。应试者的任务是从选项中选择一个。与choiceInteraction不同，递送机制必须允许应试者回顾他们的选项以及选项所处的上下文。

inlineChoiceInteraction必须受限一个响应变量，该响应变量是identifier基础类型的，并且仅有单值基数。

包含：

名称	多重性	说明
inlineChoice	[1..*]	显示给用户的选项的有序列表。除非混洗值为真，否则列表的顺序是呈现给用户的顺序。

属性名	多重性	类型	说明
shuffle	[1]	boolean = false	如果shuffle属性为真，那么递送机制必须随机决定选项呈现的顺序，但该属性受限fixed属性。
required	[0..1]	boolean = false	如果为真，那么应试者必须选择一个选项，从而构成交互的一个合法响应。

类 inlineChoice (choice)

相关类 inlineChoiceInteraction

显示给用户的简单的文本串，可以用printedVariable进行变量值替换。

抽象类 stringInteraction

派生类 extendedTextInteraction, textEntryInteraction

字符串交互受限数字响应变量，如果需要的话，可代替字符串。

如果需要关于数字响应的细节信息，那么字符串交互可以受限记录基数的响应变量。结果值包含下列域：

- stringValue: 应试者输入的字符串。
- floatValue: 应试者输入的字符串的数值，按浮点型处理。
- integerValue: 应试者输入的字符串的数值，如果没有指定小数和指数，则为NULL。整数。
- leftDigits: 小数点左边的数字个数。整数。
- rightDigits: 小数点右边的数字个数。整数。
- ndp: 应试者指定的小数数字个数。如果未给定指数，与rightDigits相同。整数。
- nsf: 应试者输入的有意义的数字个数。整数。
- exponent: 应试者给定的整数指数，如果没有指定，值为NULL。

属性名	多重性	类型	说明
base	[0..1]	integer = 10	如果字符串交互受限数字响应变量，

			那么base属性必须用于解释应试者输入值的数字基
stringIdentifier	[0..1]	identifier	如果字符串属性受限于数字响应变量，那么应试者实际输入的字符串也可以通过将交互绑定到另一个响应变量（string基础类型）而获取。
expectedLength	[0..1]	integer	expectedLength 属性给应试者提供关于需要的响应的预期总长度的提示。递送机制应该使用本属性的值以设定适用的响应框的尺寸。它不是有效性约束。
patternMask	[0..1]	string	如果 patternMask 被指定，则指出了应试者的合法响应必须遵守的正规表达式。正规表达式使用的语言定义在附录 F[XML_SCHEMA2]中。注意，应该确保应试者清楚需要的输入格式，尤其是在需要进行响应合法性检查的测试中。例如，这可以在提示中提供说明性的样例响应。
placeholderText	[0..1]	string	在可见环境中，字符串交互通常是以一个应试者可以写或打字的空白框来呈现。然而，在基于语音的环境中，有一些可以用于有声交互的 placeholderText 是很有帮助的。在需要的时候，递送机制应该使用属性的值（如果提供）来代替默认的占位符文本。执行者应该知道关于响应变量一节描述的默认值使用的问题。

类 textEntryInteraction (inlineInteraction, stringInteraction)

textEntry交互是一个从应试者处获取简单文本块的inlineInteraction。像inlineChoiceInteraction一样，递送机制必须允许应试者在上下文中回顾他们的选择。

textEntry交互必须受限于一个只具有单一基数的响应变量，基础类型必须是string、integer或float之一。

类 extendedTextInteraction (blockInteraction, stringInteraction)

扩展的文本交互是允许应试者输入扩展数量文本的blockInteraction

extendedTextInteraction必须受限于具有string、integer或float类型的响应变量。当受限于单一基值的响应变量时，需要从应试者那里获得单一的文本字符串。当受限于具有多值或有序基数的响应变量时，需要从应试者那里获得几个独立的文本字符串，参见下面的maxString。

属性名	多重性	类型	说明
maxStrings	[0..1]	integer	当交互受限于一个容器响应变量时，需要maxStrings属性。递送机制必须使用这个属性的值来控制可以从应试者接受的

			独立的字符串的最大数量。当多值字符串被接受， <code>expectedLength</code> 应用于每个字符串。
<code>minStrings</code>	[0..1]	<code>integer = 0</code>	<code>minStrings</code> 属性指定了构成一个合法响应，需要从应试者获得的独立的（非空）字符串的最小数量。如果 <code>minString</code> 是 0，那么不要求应试者输入任何字符串。 <code>minString</code> 必须小于或等于 <code>maxString</code> 施加的限制。如果交互不是受限于容器，那么 <code>minString</code> 为 1 是一种特殊情况。在这种情况下，应试者必须输入一个非空的字符串以构成一个合法的响应。构成字符串的更复杂的约束可以通过 <code>patternMask</code> 属性控制。
<code>expectedLines</code>	[0..1]	<code>integer</code>	<code>expectedLines</code> 属性提供给应试者关于要求的预期输入行数的提示。递送机制应该使用本属性的值来设定可适用的响应框的尺寸。它不是合法性约束。
<code>format</code>	[0..1]	<code>textFormat = plain</code>	用于控制应试者输入文本的格式。参见下面的 <code>textFormat</code> 。这个属性影响相关的响应变量的值被相应处理机制解释的方式，并且也控制在递送机制被获取的方式。

枚举 `textFormat`

`plain`

表明应试者输入的是纯文本。这种格式适合于短的无结构的响应。递送机制应该保留空白空间字符，以防应试者在响应中只输入了空白空间字符，在这种情况下它应该被视为空白字符串（NULL）。

`preFormatted`

表明应试者输入的文本是有预先格式的，并且应该用与[XHTML]中定义的

```
一致的方式呈现。递送机制应该保留空白空间字符，以防应试者在响应中只输入了空白空间字符，在这种情况下它应该被视为空白字符串（NULL）。
```

`xhtml`

表明应试者输入的是结构化的文本。响应变量的值是在XHTML中标记的文本。递送机制应该预留一个合适的接口，用以捕获结构化的文本，该文本可能是用一系列简单文本标记约定解释的纯文本，例如wiki页面编辑器和或是一个完全的WYSIWYG编辑器中使用的约定。

类 `hottextInteraction (blockInteraction)`

`Hottext`交互为应试者呈现一个选项集，该集合以可选的文本串的方式呈现，该文本串嵌入在上下文环境中，例如一个简单文本段落。像`choiceInteraction`一样，应试者的任务是选择一个或多个选项，最多到`maxChoices`。交互由相关响应变量的默认值初始化，通常情况下，NULL值指示没有选项被选择。

`hottextInteraction`必须受限于具有标识符基础类型的响应变量和单一或多值基数。

属性名	多重性	类型	说明
maxChoices	[1]	integer = 1	应试者可以选择的最大选项数量。如果 matchChoice 是 0，那么没有约束。如果 maxChoices 大于 1（或 0），那么交互必须受限于是具有多值基数的响应。
minChoices	[0..1]	integer = 0	要求应试者选择以构成合法响应的选项的最小数量。如果 minChoices 为 0，那么不要求应试者选择任何选项。minChoices 必须小于或等于 maxChoices 施加的限制。

包含：

名称	多重性	说明
blockStatic	[1..*]	交互的内容是简单的一条内容，例如包含热文本区域的一个简单文本段落。

类 hottext (choice, flowStatic, inlineStatic)

hottext 用在 hottextInteraction 内容中，提供单独选项。它不能包含任何嵌套的交互或其它 hottext 区域。

当 hottext 选项（被相关模版变量的值）隐藏的时候，选项的内容必须仍然可以呈现给应试者，但仅以周围素材的一部分的形式呈现。在 hottext 的情况下，隐藏选项的效果仅是使得文本串不可被应试者选择。

包含：

名称	多重性	说明
inlineStatic	[*]	

8.3 绘图交互

抽象类 hotspot

派生类 associableHotspot, hotspotChoice

一些绘图交互包括有特殊定义区域的图像或热点。

属性名	多重性	类型	说明
shape	[1]	shape	热点的形状。
coords	[1]	cords	热点的尺寸和位置，与形状联合被解释。
hotspotLabel	[0..1]	string256	对本（热）图像区域的替换文本，如果指定，必须按照与 img 替换文本相同的方式对待。对于隐藏热点，本标签被忽

类 hotspotChoice (choice, hotspot)

相关类 hotspotInteraction, graphicOrderInteraction

类 associableHotspot (associableChoice, hotspot)

相关类 graphicAssociateInteraction, graphicGapMatchInteraction

属性名	多重性	类型	说明
matchMax	[1]	integer	本选项可以关联的选项最大数目。如果 matchMax 为 0，就没有限制。
matchMin	[0..1]	integer = 0	本选项必须关联以构成合法响应的选项最小数量。如果 matchMin 为 0，那么不要求应试者将该选项与任何其它相关联。matchMin 必须小于或等于 matchMax 施加的限制。

抽象类 graphicInteraction (blockInteraction)

派生类

graphicAssociateInteraction, graphicGapMatchInteraction, graphicOrderInteraction, hotspotInteraction, selectPointInteraction

包含：

名称	多重性	说明
object	[1]	每个绘图交互都有一个相关图像，该图像以对象的形式给定，并且必须是类型属性中指定的 image 类型。

类 hotspotInteraction (graphicInteraction)

Hotspot 交互是一个具有选项集的图形交互，该选项集定义为图形区域。应试者的任务是选择一个或多个区域（热点）。热点交互只应该在题项对选项的特殊互相（被绘图对象呈现）关联需求重要的时候才使用。否则，choiceInteraction 应该使用独立的素材来代替每个选项。

递送机制必须清晰的指示图像选定的区域，也可以指示未选择的区域。有隐藏热点的交互通过 selectPointInteraction 来完成。

热点交互必须受限具有 identifier 基础类型和单一或多值基数的响应变量。

属性名	多重性	类型	说明
maxChoices	[1]	integer = 1	允许应试者选择的选项的最大数量。如果 maxChoices 是 0，那么没有约束。如果 maxChoices 大于 1（或 0），那么交互必须受限具有多值基数的响应。

minChoices	[0..1]	integer = 0	要求应试者选择以构成合法响应的选项的最小数量。如果 minChoices 是 0，那么不要求应试者选择任何选项。minChoices 必须小于或等于 maxChoices 施加的限制。
------------	--------	-------------	--

包含：

名称	多重性	说明
hotspotChoice	[1..*] {ordered}	定义应试者可以选择的选项的热点。如果递送机制不支持基于点的选择，那么给定选项的顺序必须是它们提供给应试者选择的顺序。例如，在简单的键盘导航中用到的tab顺序。如果热点交迭，那么先列出的热点覆盖后出现的热点。如果定义了默认热点，那么默认热点必须最后出现。

类 selectPointInteraction (graphicInteraction)

像hotspotInteraction一样，selectPointInteraction也是graphicInteraction。应试者的任务是选择一个或多个点。相关联的响应可以有一个areaMapping，根据与预先定义区域进行比较而对响应计分，但是递送机制不能指出这些图像的区域。只有应试者选择的实际的点可以指出。

选择点交互必须受限具有point基础类型和单一或多值基数的响应变量。

属性名	多重性	类型	说明
maxChoices	[1]	integer = 1	本属性被解释为允许应试者选择的点的最大数量。如果maxChoices大于1(或0)，那么交互必须受限具有多值基数的响应。
minChoices	[0..1]	integer = 0	要求应试者选择以构成合法响应的点的最小数目。如果 minChoices 为 0，那么不要求应试者选择任何点。minChoices 必须小于或等于 maxChoices 施加的限制。

类 graphicOrderInteraction (graphicInteraction)

graphicOrderInteraction是一个具有相应选项集的图形交互，该选项集定义为图形区域图像。应试者的任务是对区域（热点）进行排序。graphicOrderInteraction只应该在选项的特殊互相（被绘图对象呈现）关联对题项的需求重要的时候才被使用。否则，orderInteraction应该使用独立的素材来代替每个选项。

递送机制必明确指示图象的所有定义区域。

graphicOrderInteraction必须受限具有identifier基础类型和顺序基数的响应变量。

包含：

名称	多重性	说明
hotspotChoice	[1..*]	定义应试者排序选项的热点。如果递送机制不支持基于点的选择，那么给定选项的顺序必须是提供给应试者选择的顺序。例如，在简单的键盘导航中用到的tab顺序。

属性名	多重性	类型	说明
minChoices	[0..1]	integer	为了形成一个有效响应，应试者必须选择或排序的最少选项数。如果要指定minChoices，minChoices必须是1或更大但是不能超过可选的选项数量。如果未指定，所有的选项必须有序并且maxChoices属性被忽略。

类 graphicAssociateInteraction (graphicInteraction)

graphicAssociateInteraction是一个具有相应选项集的图形交互，该选项集定义为图形区域图像。应试者的任务是使区域（热点）互相关联。绘图关联交互只应该在选项的特殊互相（被绘图对象呈现）关联对题项的需求重要的时候才被使用。否则，associateInteraction应该使用独立的素材来代替每个选项。

递送机制必须明确的指示所有定义的图像区域。

associateHotspotInteraction必须受限于有pair基础属性和单一或多值基数的响应变量。

属性名	多重性	类型	说明
maxAssociations	[1]	integer = 1	允许应试者创建的关联的最大数量。如果maxAssociations为0，没有约束。如果maxAssociations大于1（或0），那么交互必须受限于具有多值基数的响应。

包含：

名称	多重性	说明
associableHotspot	[1..*]	定义需要应试者关联的选项的热点。如果递送机制不支持基于点的选择，那么给定选项的顺序必须是提供给应试者选择的顺序。例如，在简单的键盘导航中用到的tab顺序。

类 graphicGapMatchInteraction (graphicInteraction)

绘图空白匹配交互是由一系列定义为绘图图像区域（热点）和另一系列定义在图像外的空白选项组成的绘图交互。应试者必须用图像的空白来关联空白选项，并且能回顾选项指出的上下文空白中填充的图像。当设计这些交互时应注意，要确保图像的空白和期望接收到的空白选项的尺寸合适。应试者必须明确知道每个选项被关联到哪个热点。当被关联时，如果可能，选项必须在空白内整体出现，并且，在需要的交迭地方，不应该完全互相隐藏。如果应试者通过在空白上放置选项（例如，拖放）来指出联系，系统应该将它“对齐”到满足这些需要的最近的位置。

`graphicGapMatchInteraction`必须受限于具有有`directedPair`基础类型和多值基数的响应变量。选项代表配对的源，图像中的空白代表目标。不像简单`gapMatchInteraction`，如果需要，每个空白可以有几个相关选项，更进一步，同一个选项可以与`associableHotspot`关联多次，在这种情况下相应的有向对以响应变量的值的形式重复出现。

包含：

名称	多重性	说明
<code>gapImg</code>	[1..*]	用于填充空白的有序列表。如果需要，选项可以比空白少。
<code>associableHotspot</code>	[1..*]	定义应试者填充的空白热点。如果递送机制不支持基于点的选择，那么给定选项的顺序必须是提供给应试者选择的顺序。例如，在简单的键盘导航中用到的 <code>tab</code> 顺序。默认的热点必须是未定义的。

类 `positionObjectInteraction (interaction)`

相关类 `positionObjectStage`

位置对象交互包括必须被应试者放置于另一个绘图图像（背景）上的单独图像。像`selectPointInteraction`一样，相关联的相应可以有一个`areaMapping`，根据与预先定义区域进行比较而对响应计分，但是递送机制不能指出这些图像的区域。只有被应试者选择的实际的点可以被指出。

位置对象交互必须受限于具有`point`基础类型和单一或多值基数的响应变量。点记录了与背景有关的被放置在图像中间点的坐标

属性名	多重性	类型	说明
<code>centerPoint</code>	[0..2]	<code>integer</code>	<code>centerPoint</code> 属性以从图像左边开始的水平、垂直方向的偏移的形式，定义了已被放置的并且将要被视为中心的图像的点。默认情况下，它为图像矩形边界的中心。图像放置的背景可以在一些位置对象交互中被共享，因此，它自己被定义为类： <code>positionObjectStage</code> 。
<code>maxChoices</code>	[1]	<code>integer = 1</code>	图像可以放置的位置（在背景上）的最大数值。如果 <code>matchChoices</code> 为0，没有限制。如果 <code>maxChoices</code> 大于1（或0），那么交互必须受限于具有多值基数的响应。

minChoices	[0..1]	integer	为了形成一个有效响应必须放置的图像的最小数量。如果指定，minChoices必须是1或更大，但是不能超过maxChoices施加的限制。
------------	--------	---------	--

包含：

名称	多重性	说明
object	[1]	应试者放置在背景上的图像。

类 positionObjectStage (block)

包含：

名称	多重性	说明
object	[1]	被用作背景的图像，每个独立的positionObjectInteractions允许应试者在它上面放置它们的对象。
positionObjectInteraction	[1..*]	

8.4 混合交互

类 sliderInteraction (blockInteraction)

滑动交互作用是给应试者在上、下界限之间选择数值的控制。它必须受限于具有单一基数和integer或float基础类型的响应变量。

属性名	多重性	类型	说明
lowerBound	[1]	float	如果相关响应变量是integer类型的，那么lowerBound必须近似到小于或等于给定值的最大整数。
upperBound	[1]	float	如果相关响应变量是integer类型的，那么upperBound必须近似到大于或等于给定值的最小整数。
step	[0..1]	integer	控制移动的步长。例如，如果lowerBound和upperBound是[0,10]，step是2，那么响应会被约束到值的集合{0,2,4,6,8,10}中。如果是整数响应，默认步长是1，否则假定滑动条运行于几乎连续的刻度。
stepLabel	[0..1]	boolean = false	默认情况下，滑动条在它们的两端标注。stepLabel属性控制是否每个滑动条的步长也要被标注。递送机制不可能保证标注步长，所以这个属性应该只视为请求。
orientation	[0..1]	orientation	方向属性为呈现系统提供了一个提示，

			提示是滑动条通过固有的垂直或水平解释来指示量的值。例如，指示高度值的交互会将方向设置为垂直，如果水平呈现会在一定程度上增加题项的难度。
reverse	[0..1]	Boolean	反向属性为呈现系统提供了一个提示，该提示滑动条用于指示正常感觉的上下界限是颠倒的量的值。例如，用于指示海平面以下深度的交互会指定垂直方向和设定反向。 注意，除非为相关响应变量设定默认值，否则滑动交互没有默认或初始化位置。如果目前有值被选中，必须明确的给应试者指出。

类 mediaInteraction (blockInteraction)

媒体交互使应试者与基于时间的媒体对象的交互方式有更多的控制，并且存在相关响应变量说明媒体对象被浏览的次数，此响应变量必须是integer基础类型和单一基数的。

属性名	多重性	类型	说明
autostart	[1]	Boolean	自动开始属性决定媒体对象是否应该在应试者开始尝试（真）时就开始，或者媒体对象是否应该在应试者控制下（假）开始。
minPlays	[0..1]	integer = 0	minPlays属性指示媒体对象被应试者播放的最小次数。实现它所需的技术随系统改变而改变，在一些系统中它是根本不可能的。默认没有最小值。播放媒体对象未达到最小次数，会产生一个非法响应。
maxPlays	[0..1]	integer = 0	maxPlays属性指示媒体对象可以至多被播放maxPlays次——应试者不能播放媒体对象多于maxPlay次。为0的值（默认）表示没有限制。
loop	[0..1]	boolean = false	循环属性用于设定连续播放模式。在连续播放模式中，一旦媒体对象开始播放，它便会连续播放（受限于maxPlays）。

包含：

名称	多重性	说明
object	[1]	媒体对象自身。

类 drawingInteraction (blockInteraction)

画图交互允许应试者使用普通画图工具集合来修饰给定的绘图图像（画布）。它必须受限于具有文件类型和单一基数的响应变量。结果是和原始图像具有相同格式的文件。

包含：

名称	多重性	说明
object	[1]	在画图过程中用作画布的图像，必须是image类型的对象，和类型属性指定的一样

类 uploadInteraction (blockInteraction)

上传交互允许应试者上传代表它们响应的文件。它必须受限于具有文件类型和单一基数的响应变量。

属性名	多重性	类型	说明
type	[0..1]	mimeType	预期的上传文件的mime类型。

类 customInteraction (block, flow, interaction)

定制交互为本标准增加了扩展性，以提供对目前未归档的交互的支持的机会。

8.5 结束尝试的替换方式

类 endAttemptInteraction (inlineInteraction)

结束尝试交互是一种特殊类型的交互，允许题项开发者提供给应试者对于应试者终止一次尝试的方式的控制。应试者可以用交互来立即终止尝试（触发响应处理），代表性的，请求提示。它必须受限于具有boolean基础类型和单一基数的响应变量。

如果应试者用endAttemptInteraction调用响应处理，那么相关的响应变量设为真。如果响应处理通过任何其它方式调用，通过一个不同的endAttemptInteraction或通过递送机制的默认方法，那么相关的响应变量被设为假。响应变量的默认值总是被忽略的。

属性名	多重性	类型	说明
title	[1]	string	使用本属性作为结束尝试的提示字符串显示给应试者。它应该比较短，最好一个词。典型值是“Hint”。例如，在绘图环境中，它会被呈现为按钮的标签，当按下时，结束尝试。

9 响应处理

响应处理是递送机制基于应试者的响应而产生结果的过程。结果可以用于为应试者提供反馈。反馈既可以在应试者尝试结束之后立即提供，也可以在一段时间后提供。也可以作为题项时间段概要报告的一部分。因此，对试题解答的尝试结束和响应处理，必须是对用户动作的直接响应，或对某些预期事件的响应，例如测试的结束。

对于非自适应题项，在每次调用响应处理之前，结果变量的值被重设为它们的默认值（或 NULL，如果没有给定默认值）。然而，尽管对于非自适应题项，递送机制可能多次调用响应处理，但它必须只

报告产生的第一批结果。也可有其他协议限制尝试次数。对于自适应题项，结果变量的值不被重设定为它们的默认值。支持自适应题项的递送机制必须允许应试者为修改和提交他们的响应，并且必须只报告最后一批响应结果。此外，它必须对应试者呈现所有适用的模型和综合反馈。并发的响应处理可以考虑应试者更新时间短结果时看到的反馈。通过设定内嵌结果变量，自适应题项可以通知递送机制应试者完成交互。

9.1 响应处理模板

响应处理包括一系列响应规则的应用，包括响应条件的测试和题项变量表达式的评估。对设计为只支持简单用例的递送系统，系统的实现、条件测试和处理会有一些障碍。

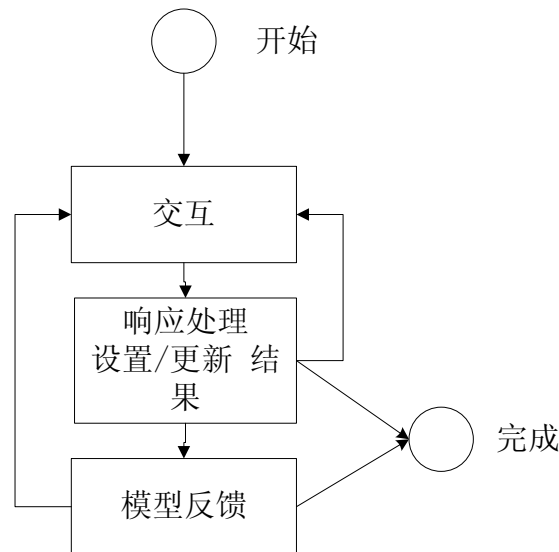


图5 反馈及进一步交互

为了解决该问题，普遍响应处理的实现可作为一项可选的特征。不支持它的机制可实现标准响应处理器代替（下文描述为响应处理模版）。这些模版用本标准定义的处理语言描述，并且一起发布（以XML的形式）。支持普遍响应处理的递送机制不需要实现特殊的机制以支持模版，因为当处理评估题项时，模版可以直接转换。不支持普遍响应处理，但支持不属于下面描述的标准模版的响应处理机制的递送机制，应该可以定义自己的模版。希望为递送机制写题项的开发者就可以引用这些定制模版。发布这些定制模版就可以确保这些题项可以用于不支持普遍响应处理的递送机制。

9.1.1 标准模版

正确匹配

正确匹配响应处理模版用匹配操作符匹配响应变量 **RESPONSE** 和它的正确值。根据测试结果设定结果变量 **SCORE** 为 0 或 1。被称为 **RESPONSE** 的响应变量必须声明，并且有一个相关的正确值。类似，结果变量 **SCORE** 也必须声明。模版可应用于任何基础类型，对复杂类型应用有一定限制。这个模版不应该用于测试浮点类型的数值的相等。注意模版总是为 **SCORE** 设定一个值，即使没有给定 **SCORE**。

映射响应

映射响应处理使用 `mapResponse` 操作符来映射响应变量 **RESPONSE** 的值到一个为结果 **SCORE** 设定的值。注意使用 `mapResponse` 时，**SCORE** 需要是浮点类型的。两个变量必须都被声明，**RESPONSE** 必须有一个相关的映射。模版应用于任何基础类型和多重性的响应。参见 `mapResponse` 的解释以获得应用于容器时行为细节。如果 **RESPONSE** 为 **NULL**，**SCORE** 被设为 0.0。

映射响应点

映射响应点处理模版使用 `mapResponsePoint` 操作符来映射响应变量 `RESPONSE` 值到一个为结果 `SCORE` 设定的值。两个变量必须声明，`RESPONSE` 必须有点基础类型。参见 `mapResponsePoint` 的解释。如果 `RESPONSE` 为 `NULL`，`SCORE` 被设为 0。

9.2 普遍响应处理

Class : `responseProcessing`

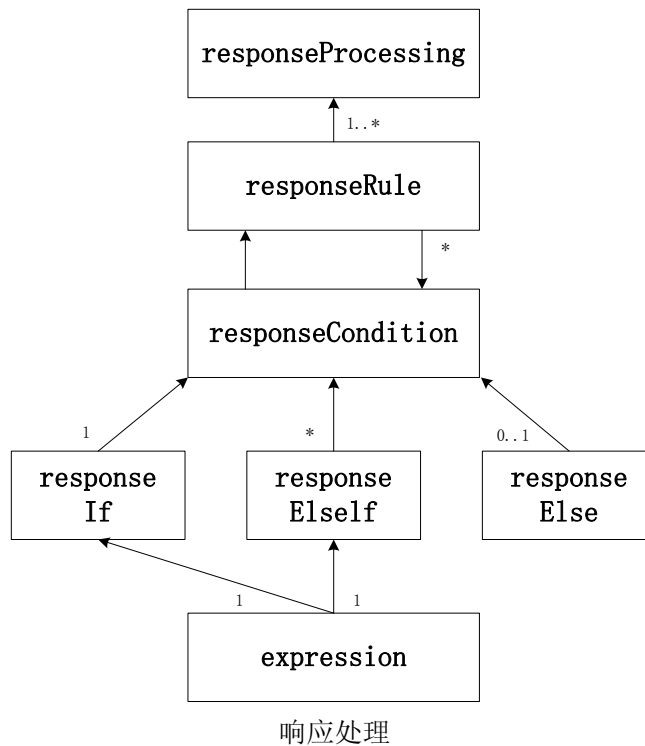
相关类: `assessmentItem`

属性名	多重性	类型	说明
<code>template</code>	[0..1]	<code>uri</code>	如果给定模版标识符，它可能用于定位外部定义的响应处理模版。从外部模版得到的规则可以用于代替题项自身定义的规则，尽管如果两个都给定，内部规则依然是首选。
<code>templateLocation</code>	[0..1]	<code>uri</code>	在实践中，模版属性可以良好的包含 URN 或存储在遥远 web 服务器上的模版 URI，例如本标准定义的标准响应处理模版。当处理 <code>assessmentItem</code> 时，离线工作的工具不能从 URN 或遥远的 URI 获得模版。 <code>templateLocation</code> 属性提供替换 URI，如：要解析的离 <code>assessmentItem</code> 自身的地点近的相关 URI，可以用于获得响应处理模版的复制。如果递送系统能单从模版标识符决定正确行为，那么模板位置应该被忽略。例如，递送系统可能有用于处理上文定义的标准模版的内嵌过程。

包含:

`responseRule` [*]

从应试者赋给的响应变量值到为题项结果变量设置的合适的值的映射，通过一些规则实现。



抽象类: responseRule

派生类:

exitResponse, include, lookupOutcomeValue, responseCondition, responseProcessingFragment, setOutcomeValue

相关类:

responseElse, responseProcessingFragment, responseIf, responseProcessing, responseElseIf

响应规则是 responseCondition、简单动作或 responseProcessingFragment。响应规则定义用于从响应（即分数）得到结果所必需的轻量级编程语言。注意这种编程语言包含控制结构的最小值，更复杂的积分规则必须用其它语言编码，并且用 customOperator 引用。

Class : responseCondition

包含:

名称	多重性	说明
responseIf	[1]	
responseElseIf	[*]	
responseElse	[0..1]	

如果在 responseIf 或 responseElseIf 中给定的表达式评估为真，那么遵循包含的子规则，任何后面的 responseElseIf 或 responseElse 部分对于这个响应条件被忽略。如果在 responseIf 或 responseElseIf 中给定的表达式评估为假，如果有更多的 responseElseIf，需要传递到下一个 responseElseIf，， responseElse 的子规则也被遵循（如果指定）。

Class : responseIf

相关类: responseCondition

包含:

名称	多重性	说明
expression	[1]	
responseRule	[*]	

responseIf 部分必须包含有效布尔基础类型和单一多重性的表达式。为了获得采用的运行时数据模型更多信息，参见 Expressions。它也包含一系列子规则。如果表达式为真，那么子规则被处理，否则，它们被跳过（包括如果表达式为 NULL 的情况），后续的 responseElseIf 或 responseElse 部分（如果存在）被代替考虑。

Class : responseElseIf

responseElseIf 与 responseIf 用等同的方式被定义。

相关类 responseCondition

包含：

名称	多重性	说明
expression	[1]	
responseRule	[*]	

Class : responseElse

相关类：responseCondition

包含：responseRule [*]

Class : setOutcomeValue (outcomeRule, responseRule)

属性名	多重性	类型	说明
identifier	[1]	identifier	要设定的结果变量。

包含：

Expression[1]

该表达式必须在基类型和多重性方面与要设定的结果变量相同。setOutcomeValue 规则设定结果变量的值为从相关表达式得到的值。结果变量可以用于前面的值更新，换句话说，正在设定的结果变量可能出现在表达式中，其值在前面赋给它。

使用数字基本类型时需要专门注意，因为浮点值不能被赋给整数变量，反之亦然。truncate, round, 或 integerToFloat 操作符必须用于实现数字类型转换。

Class : lookupOutcomeValue (outcomeRule, responseRule)

lookupOutcomeValue 规则设定结果变量的值是通过在与结果声明相关的 lookupTable 的表达式相关的值中查找而获得。

属性名	多重性	类型	说明
identifier	[1]	identifier	要设定的结果变量。

包含：expression [1]

必须有单一多重性和 integer, float, 或 duration 有效基础类型的表达式。当相关表格是 matchTable 时需要整数类型。

Class : exitResponse (responseRule)

退出响应后立即结束响应处理

10 模式反馈

Class : modalFeedback

相关类: assessmentItem

模式反馈在响应处理结束之后直接显示给应试者。结果变量的值和显示隐藏及标识符属性联合使用以决定反馈是否通过与反馈元素相似的方式显示。

属性名	多重性	类型	说明
outcomeIdentifier	[1]	identifier	
showHide	[1]	showHide	
identifier	[1]	identifier	
title	[0..1]	string	

递送机制可以给应试者显示标题，如作为模式弹出窗口的标题。但并不要求一定要呈现。

Contains : flowStatic [*]

modalFeedback 的内容不能包含任何交互。

11 题项模板

题项模版是用于产生大量相似题项的模版。这样的题项经常被称为克隆题项。题项模版可以通过专门的克隆机制产生题项，或在递送机制支持的情况下，在题项时间段开始时，直接动态产生克隆。

每个由题项模版克隆的题项除给定一系列模版变量的值外，都是相同的。因此，如果评估题项包含一个或多个模版定义和一系列为其赋值的模版处理规则，那么它就是题项模版。

创建克隆题项的递送机制必须给每个克隆赋不同的标识符，并且记录模版变量的值。通过将题项模版的标识符替换为克隆题项的标识符，并且在报告中加入模版变量的值。具有这样克隆的题项时间段的报告就能被转换为与原始题项模版相同的报告。

Class : templateDeclaration (variableDeclaration)

相关类: assessmentItem

模版定义用于克隆题项的题项变量，值只可以在模版处理中被设定，在题项体中被引用，为了个别克隆，如果克隆处理影响题项被计分的方式，可能也用在相应处理规则中。

模版变量被初始化为题项时间段的一部分。它们的值在模版处理中被初始化，在时间段中表现为常量。

属性名	多重性	类型	说明
paramVariable	[0..1]	boolean=false	决定模版变量的值是否应该被替换为匹配它名字的目标参数值。参见 param 以获取更多信息。
mathVariable	[0..1]	boolean=false	决定模版变量值是否应该被替换为在 MathML 中匹配它名字的标识符。参见结合模版变量和 MathML 以获取更多信息。

11.1 在题项体中使用模版变量

模版变量可以在题项体中被打印变量对象引用。模版变量的值用于创建合适的显示文本。模版变量也可以通过与有反馈元素的结果变量相似的方式控制两个模版元素的内容。最终，模版变量可以用于控制交互选项的可见性。

Abstract class : templateElement (bodyElement)

派生类: `templateBlock`, `templateInline`

属性名	多重性	类型	说明
<code>templateIdentifier</code>	[1]	<code>identifier</code>	必须是标识符基础类型和单一或多值多重性的模版变量的标识符。模板元素的可见性由其值控制。
<code>showHide</code>	[1]	<code>showHide=show</code>	
<code>identifier</code>	[1]	<code>identifier</code>	

`showHide` 和 `identifier` 标识符属性决定 `templateElement` 的可见性如何控制。这与 `feedbackElement` 的 `showHide` 和标识符属性的作用大体相同。模版元素不能直接或间接的包含任何交互。

Class : `templateBlock` (`blockStatic`, `flowStatic`, `templateElement`)

包含 : `blockStatic` [*]

Class : `templateInline` (`flowStatic`, `inlineStatic`, `templateElement`)

包含 : `inlineStatic` [*]

11.2 操作符属性值中使用模版变量

一些表达式中的操作符有控制表达式行为的属性。通过引用具有适当基础类型和多重性的模版变量, 在操作符自身评估的同时计算模版变量的值,, 就可以代替属性目标类型的值空间中的值。支持这类属性值替换的属性被声明为下列类型之一。

绑定变量为字符串时, 变量标识符必须括在花括号内。例如, 字符串 "{myVariable}" 指具有 `myVariable` 标识符的模版变量。必须在相应的模板定义中匹配标识符。

Datatype: `integerOrTemplateRef`

用于整数类型属性的类型, 可以用具有整数类型和单一多重性的模版变量值替换。

Datatype: `floatOrTemplateRef`

用于 `float` 类型属性的类型, 可以用具有 `float` 基础类型和单一多重性的模版变量值替换。

Datatype: `stringOrTemplateRef`

用于 `string` 类型属性的类型, 可以用具有 `string` 基础类型和单一多重性的模版变量值替换。

当模版参考受限于 `string` (使用上述的强制绑定) 时, 有潜在的不确定性。因此, 如果 `string` 属性表示的是模版变量的参考, 而且是没有给定名称的变量, 它应该被简单视为 `string` 值。

11.3 模板处理

Class : `templateProcessing`

相关类 `assessmentItem`

Contains : `templateRule` [1..*]

模版处理包括一个或多个模板规则, 然后克隆机制或递送系统根据该规则赋值给模版变量, 而不是结果变量。

Abstract class : `templateRule`

派生类 classes:

`exitTemplate`, `setCorrectResponse`, `setDefaultValue`, `setTemplateValue`, `templateCondition`

相关类 classes:

templateProcessing, templateElseIf, templateIf, templateElse

模版规则是模板条件或简单动作。模版规则定义创建克隆题项所必需的轻量编程语言。编程语言包含最小数量的控制结构，更复杂的克隆规则超出本标准的范围。

用在模板规则中的表达式不能引用响应变量或结果变量的值。只能引用模版变量的值。

Class : templateCondition (templateRule)

包含

属性名	多重性	说明
templateIf	[1]	
templateElseIf	[*]	
templateElse	[0..1]	

如果 templateIf 或 templateElseIf 中给定的表达式评估为真，那么它的子规则被遵循，并且任何之后的 templateElseIf 或 templateElse 部分对于本模版条件被忽略。

Class : templateIf

相关类: templateCondition

包含:

属性名	多重性	说明
expression	[1]	
templateRule	[*]	

templateIf 部分必须包含有效 boolean 基础类型和单一多重性的表达式。更多关于采用运行数据模型的信息参见 Expressions。templateIf 也包含一系列子规则。如果表达式为真，那么子规则被处理，否则跳过（包括表达式为 NULL 的情况），之后的 templateElseIf 或 templateElse 部分（如果有）执行。

Class : templateElseIf

相关类 templateCondition

包含

属性名	多重性	说明
expression	[1]	
templateRule	[*]	

templateElseIf 与 templateIf 定义等同。

Class : templateElse

相关类 templateCondition

包含 : templateRule [*]

Class : setTemplateValue (templateRule)

属性名	多重性	类型	说明
identifier	[1]	identifier	要被设定的模版变量

包含 expression [1]

必须具有有效基础类型和多重性，匹配模版变量的基础类型和多重性的表达式。

GB/T×××—××××

setTemplateValue 规则设定模版变量的值为从相关表达式获得的值。模版变量可以参考之前赋的值而更新，换句话说，被设定的模版变量可以出现于之前赋给它值的表达式中。

Class : setCorrectResponse (templateRule)

属性名	多重性	类型	说明
identifier	[1]	identifier	具有正确值集合的响应变量

包含 expression [1]

Class : setDefaultValue (templateRule)

属性名	多重性	类型	说明
identifier	[1]	identifier	响应或结果变量的缺省值集合

包含 expression [1]

Class : exitTemplate (templateRule)

退出规则立即中止模版响应。

12 测试

测试由 assessmentTest 类呈现。

Class : assessmentTest

测试是一组具有一系列决定应试者可以看到那些题项、以什么顺序、以及应试者以什么方式与题项交互的相关规则的 assessmentItem。这些规则描述了贯穿测试的可靠路线，响应何时提交给响应处理，以及何时给定反馈（如果有）。

属性名	多重性	类型	说明
identifier	[1]	string	测试的基本标识符。这个标识符必须在测试的元数据中有相应的条目。参见元数据和使用数据以获得更多信息。
title	[1]	string	使用 assessmentTest 的 title 的目的是可以在任何测试时间段之外选择测试。。因此，递送机制可以在任何时候显示 title 给应试者，但不是必须这样做。
toolName	[0..1]	string256	工具名属性允许工具创建测试以识别自己。其它处理系统可以使用这个信息以解释应用具体数据的内容，例如测试题目元素的标签。
toolVersion	[0..1]	string256	工具版本属性允许创建测试的工具以识别自己的版本。这个值只能在 toolName 的环境中解释。

包含：

名称	多重性	说明
outcomeDeclaration	[*]	每个测试有一系列相关的结果。这些结果的值由测试的 outcomeProcessing 规则

		设定。
timeLimits	[0..1]	选择性的控制，应试者在整个测试中允许的时间量。
testPart	[1..*]	每个测试被分解为一个或多个部分，每个部分又可能轮流着被分解为节、子节等等。testPart 代表测试的主要分割，它用于控制在分解的部分中应用于所有节和子节的基本模式参数。
outcomeProcessing	[0..1]	计算测试结果的值的一系列规则。
testFeedback	[*]	包含由测试结果控制的测试级别的反馈。

12.1 导航和提交

规定了每个测试两种行为方式：导航模式和提交模式。

Enumeration: navigationMode

linear

nonlinear

导航模式决定应试者可以采用的普通路线。linear（线性）模式的 testPart 限制应试者要依次尝试每个题项。一旦应试者继续前进，就不允许他们返回。nonlinear（非线性）模式的 testPart 去除这个限制。应试者可以在任何时候自由移动到任何题项。测试递送系统在遵守当前有效的导航模式条件下可以自由实现它们自己的用户界面，以协助导航。测试递送系统可以简单的通过提供在测试中向前或向后移动的方法来实现非线性模式。

Enumeration: submissionMode

individual

simultaneous

提交模式决定应试者的响应何时提交给响应处理。individual（单独）模式的 testPart 要求应试者按照逐个题项的原则提交他们的响应。在 simultaneous（并发）模式下，应试者的响应在 testPart 的最后一起提交。

提交模式的选择决定在测试中每个题项时间段可以通过的状态。在并发模式中，在 testPart 完成之前响应处理不能发生，所以此时每个题项时间段只经历交互和暂停状态。根据定义，应试者对于每个题项只能进行一次尝试，反馈在测试中不可见。在测试后，应试者是否可以返回以查看它们的响应和/或任何题项级别的反馈，已经超出了本标准的范围。并发模式是典型的基于纸张的测试。

在单独模式，响应处理可以发生在测试中，题项时间段经历题项中描述的任何状态，强制服从 itemSessionControl 的设定。当同时支持非线性导航模式和单独提交的系统设计用户界面时，应该格外小心。应试者可能改变他们对于一个题项的响应，然后通过转移到同一个测试部分的另一个题项，而将它置于暂停状态。测试递送系统需要使应试者明确，在测试部分结束的时候有未提交的响应（类似于在传统文档编辑系统中未存储的改动）。如果合适的话，在单独模式中测试递送系统可以强制应试者在切换到一个不同的题项之前，提交或放弃这样的响应。

Class : testPart

属性名	多重性	类型	说明
identifier	[1]	identifier	在测试中，测试段的标识符必须是唯一的，而且不能与 assessmentSection 和 assessmentItemRef 的标识符相同。
navigationMode	[1]	navigationMode	
submissionMode	[1]	submissionMode	

包含:

名称	多重性	说明
preCondition	[*]	测试中可选的评估条件集合，决定这部分是否被跳过。
branchRule	[*]	可选的规则集合，在测试中评估，为了设定作为下一个测试部分的替换目标。
itemSessionControl	[0..1]	用于控制在这部分中每个题项时间段允许状态的参数。这些值在节和题项级别可以被覆盖。
timeLimits	[0..1]	可选的控制，在这个测试部分应试者允许的时间量。
assessmentSection	[1..*]	每个 testPart 包含的题项被安排进节和子节中。
testFeedback	[*]	具体针对这一测试部分的测试级别的反馈。

12.2 测试结构

对于每个测试时间段，根据包含它们的节所定义的规则，题项和子节选择以及安排顺序。定义测试基本结构。在结构中应试者可以采用的路线被测试部分和 preConditions 或 branchRules 的模式设定所控制。

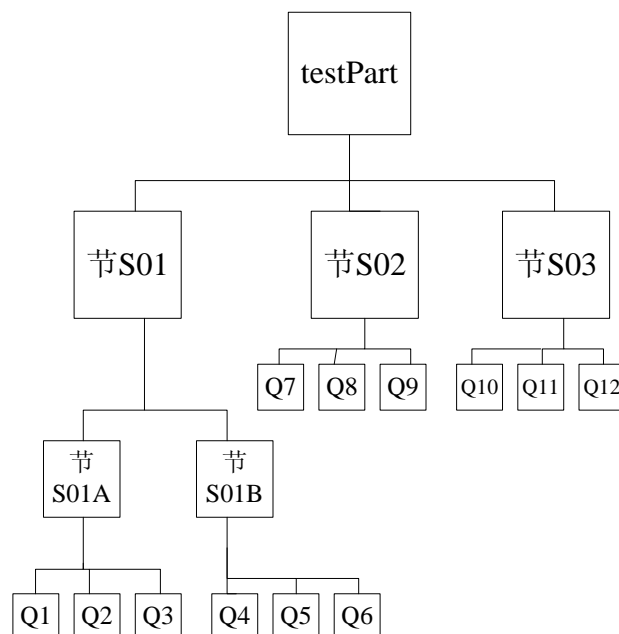


图12.1 选择和排序

图 12.1 说明测试部分和题项结构化为节和子节的方式。

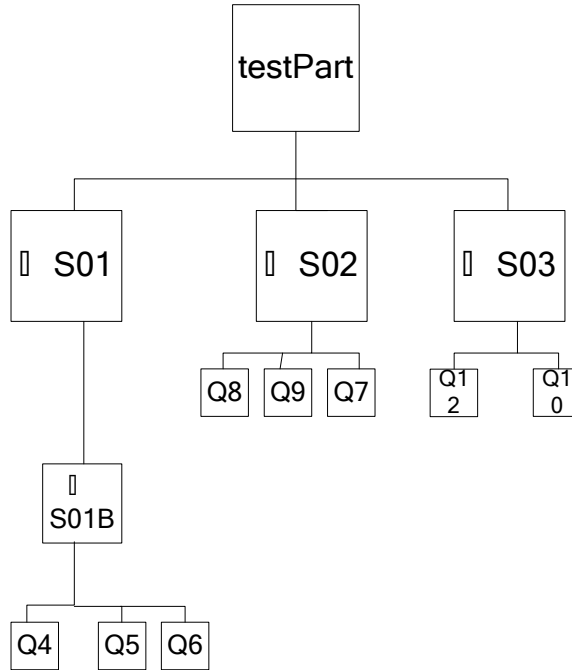


图12.2 选择和排序

第二幅图说明一个应用选择和排序的规则之后的同一个测试部分的具体实例。节 S01 的一个规则只选择了 S01 中的一个——S01B，S02 的规则混合了它包含的题项，最后，S03 的规则选择它包含的 3 个题项中的两个，并且对结果进行了合并。

Class : selection

相关类: assessmentSection

selection 类指定用于选择每个测试时间段的节的子元素的规则。如果没有给定选择规则，我们假定所有元素将被选择。

属性名	多重性	类型	说明
select	[1]	integer	将被选择的子元素的数量。不管字节有多少子元素，以及它们是否可见，子节总是当作 1 来计算，。当且仅当 withReplacement 为真，要选择子元素的数量可能会超过定义子元素的数量。
withReplacement	[0..1]	boolean=false	选择子元素时，通常每个元素只能选择一次。换句话说，当从{A,B,C,D}选择 3 个元素时，可能的结果是 {A,B,C}, {A,B,D}, {A,C,D}, 和 {B,C,D}。通过设定 withReplacement 为真，每个元素可以被选择多次。从{A,B,C,D}选择 3 个节

点,就可以有联合的结果,例如{A,A,A}, {A,A,B}等。

selection 类也提供机会来扩展本标准, 以支持更多复杂选择算法。

Class : ordering

相关类: assessmentSection

ordering 类用于指定选择之后的节中的子元素的规则。如果没有给定排序规则, 我们假设元素将被排序为它们定义的顺序。

属性名	多重性	类型	说明
shuffle	[1]	boolean=false	如果为真, 将会导致子元素的顺序是随机的, 如果为假, 使用子元素被定义的顺序。

子节总是被视为单一的供选择的块, 但是在混合时被处理的方式取决于它们的可见性。可见的子节总是被处理为单一块, 但是不可见的子节只在它的 keepTogether 属性为真的条件下被处理为单一块。否则, 不可见子节的子元素在混合之前被合并到父项的选择中。

ordering 类也提供扩展本标准的接口, 以包含更多排序算法的支持。选择和排序规则为测试的每个实例定义一个序列的题项。序列开始于第一个测试部分第一节的第一个题项, 持续到最后一个测试部分最后一节的最后一个题项。序列在测试中是不变的。通常, 这是应试者感知的逻辑顺序, 但是 preConditions 和/或 branchRules 的使用可以影响所采用的具体规范路线。

具有替换的选择的使用使得被同一个 assessmentItemRef 引用的两个或更多的同一题项的实例出现在测试的题项序列中。因此使这样一个题项成为 branchRule 的目标是一个错误。此外, 当报告测试结果时, 每个题项的序列数量也要被报告, 以避免不确定性。参见结果报告。

Abstract Class : sectionPart

派生类: assessmentItemRef, assessmentSection, include

相关类: assessmentSection

聚集单独题项引用和/或子节的节。两种类型的子元素共享一些共同参数。

属性名	多重性	类型	说明
identifier	[1]	identifier	在测试中, 节或题项参考的标识符必须是唯一的, 不能是任何 testPart 的标识符。
required	[0..1]	boolean=false	如果需要子元素, 在选择中 required 必须出现 (至少一次)。节选择少于它包含必须原色的数量的子元素的选择规则, 是有错误的。
fixed	[0..1]	boolean=false	如果子元素是固定的, 它不能被混洗。当与选择规则联合使用时, 直到选择发生之后, 固定元素的位置将不固定。例如, 从{A,B,C,D}不替换的选择 3 个元素, 可能结果是{A,B,C}。如果选择服从混洗但是 B 是固定的, 那么像{A,C,B}之类的置换是不允许的, 尽管像{C,B,A}之类的置换被允许。

包含:

名称	多重性	说明
precondition	[*]	可选的，在测试中评估的条件集合，决定题项或节是否被跳过。（在非线性模式中，precondition 将被忽略）。
branchRule	[*]	可选的规则集合，在测试中被评估，用于设定一个替换目标作为下一个题项或节。（在非线性模式中，分支规则被忽略）。
itemSessionControl	[0..1]	用于控制每个题项时间段允许状态的参数（在子节或题项级别可以被覆盖）。
timeLimits	[0..1]	可选的控制，在题项或节中应试者允许的时间量。

Class : assessmentSection (sectionPart)

相关类: testPart

属性名	多重性	类型	说明
title	[1]	string	节的 title 的目的是在节的内容不可提供的情况下使节可以被选择，例如，当应试者浏览测试的时候。因此，递送机制可以在测试中的任何时候显示 title 给应试者，但是不必须这样做。
visible	[1]	boolean	可见的节是应试者可以确认的节。例如，递送机制可能提供测试的层次视图以协助导航。在这种导航中，可见的节是在层次中可见的节点。相反的，不可见的节是对应试者不可见的节点——不可见节中的子元素呈现给应试者，好像它们是父节（或 testPart）的部分。节的可见性不影响它的子元素的可见性。每个节的可见性由它自己的可见性属性的值单独决定。
keepTogether	[0..1]	boolean=true	有受到混洗的父节的不可见节可以指定它的子节是否被混洗为块，或与父节的其它子节混合。它的子节会呈现给应试者，好像它们是父节的一部分。

包含:

名称	多重性	说明
selection	[0..1]	用于在测试的每个实例中，选择节的哪个子节将使用规则。
ordering	[0..1]	用于决定测试的每个实例中，节的子节被安排的顺序的规则。 每个子节在父节之前有自己的选择和排

			序规则来遵守。子节可以混合自己子节的顺序，虽然在混合父节时依然要求它们保持在一起。
rubricBlock	[*]		节题目呈现给应试者，每个题项被节包含（直接或间接）。由于节是可嵌套的，对每个题项所呈现的题目是从最顶层的节直到题项直接装入的节标题块的串联。
sectionPart	[*]		空节被假定为使用选择扩展机制来描述它的内容的节。

Class : assessmentItemRef (sectionPart)

题项通过引用，而不是直接聚合组成测试。注意引用标识符在测试外不具有任何意义。而且在任何类别的环境中，不要求它是唯一的，或者呈现在题项的元数据中。标识符的语法比 `assessmentItem` 本身的标识符属性限制更多。

属性名	多重性	类型	说明
href	[1]	uri	用于引用题项文件的 uri（例如，同一个内容包的其他地方）。不要求它是唯一的。测试可以在其中多次引用同一个题项。注意无论如何，每个引用必须有一个唯一的标识符。
category	[*]	identifier	题项可以随意的分配给一个或多个类别。在结果处理中，类别作用是允许哪些题项结果被合计。

包含:

名称	多重性	说明
variableMapping	[*]	变量映射用于为测试改变题项结果的名
weight	[*]	weights 允许按照题项结果的比例自定义值。
templateDefault	[*]	缺省模版用于改变被题项定义的模版变量的默认值，基于测试级别评估的表达式。参见 <code>templateDefault</code> 以获取更多信息。

Class : weight

相关类: `assessmentItemRef`

单独题项分数对于全部测试分数的贡献的随测试不同而改变。题项的分数被加权。权重被定义为测试中题项(`assessmentItemRef`)的每个引用的一部分。

属性名	多重性	类型	说明
identifier	[1]	identifier	题项可以有任意数量的权重，每个要给定一个标识符，用于在结果处理中引用。（参见变量和 <code>testVariables</code> 定义。）

value	[1]	float	权重是浮点值。权中可以应用于具有 float 或 integer 基础类型的结果变量。权重应用于结果处理中变量值被评估的时刻。即使变量自身被声明为 integer 基础类型，结果总是看作 float 基础类型。
-------	-----	-------	---

Class : variableMapping

相关类: assessmentItemRef

变量映射允许声明时名为相应题项中的 sourceIdentifier 的结果变量被视为如同它们在 outcomeProcessing 中被声明时名为 targetIdentifier。在使用 testVariables 时，变量映射允许对结果合计的方式进行更多控制。

属性名	多重性	类型	说明
sourceIdentifier	[1]	identifier	
targetIdentifier	[1]	identifier	

Class : templateDefault

相关类: assessmentItemRef

基于模版实例化的测试上下文，题项中模版变量的默认值可以被覆盖。这个值可以通过计算在测试级别的题项引用中的表达式获取，因此可能有一些值依赖于在测试中其他题项的变量的值或者在测试级别定义的结果中的变量的值。

对于一致的结果，在正确的时间评估表达式是至关重要的。当 assessmentItemRef 存在于用线性模式导航的 testPart 中，表达式在第一次尝试开始之前立即评估，在任何前提评估和遵守之后，但是在题项自身的 templateProcessing 规则被遵循之前。在非线性模式中，表达式在 testPart 开始时被评估。在每种情况下，计时在 submissionMode 生效的时候失效。需要注意确保响应变量的值在它们被提交之前不被使用，以及结果变量在它们的值被相应的响应规则或结果处理步骤设定之前不被使用。

属性名	多重性	类型	说明
templateIdentifier	[1]	identifier	受影响的模版变量的标识符。

包含:

名称	多重性	说明
expression	[1]	表达式的结果必须具有匹配 templateDeclaration 的相关变量的声明的基础类型和多重性。 覆盖模版默认值的功能允许题项模版在测试中链接到其它题项（或模版）。从一个题项得到的应试者响应可以直接影响呈现以及题项呈现的方式。然而，一旦模版被实例化，它独立运转，题项之间不能创建动态链接，它们之间没有共享变量空间的概念。

12.3 时间限制**Class : timeLimits**

相关类: assessmentTest, testPart, sectionPart

属性名	多重性	类型	说明
minTime	[0..1]	duration	
maxTime	[0..1]	duration	

在具体的 assessmentTest 环境中, 题项或题项组, 可以服从时间约束。本标准支持最小和最大时间约束。对于单独题项的控制时间是由内嵌的响应变量 duration 定义的题项时间段的持续时间。对于 assessmentSection, testPart 和整个的 assessmentTest, 时间限制与所有的题项时间段加上任何其它用于浏览那个测试部分的时间相关。换句话说, 时间包括用于没有题项交互的状态, 例如专用导航屏幕。

最小时间适用于 assessmentSections 和 assessmentItems, 只有在线性导航模式生效的时候。

当时间限制有效时, 递送机制必须追踪和报告每个测试部分的时间。如果时间限制对于给定测试部分或节无效, 那么消耗的时间可以被追踪和报告, 但是不是必须的。类似的, 如果时间限制对于非时间相关题项无效, 那么消耗的时间可以被报告, 但是也不是必须的。

用在测试的时间被记录, 如同它是被称作持续时间的内嵌响应变量, 在测试级别被声明, 具有 duration 基础类型和单一多重性。类似的, 被当作内嵌响应变量的测试部分或节的所用时间在各自的作用域内被声明。通过变量名称 duration 以及在时期特征之后的部分或节的标识符作为前缀, 这些持续时间的值可以在 outcomeProcessing 中被引用。

13 结果处理

Class : outcomeProcessing

相关类: assessmentTest

结果处理在每次应试者提交一个(当处于单独提交模式)或一组(当处于并行提交模式)题项响应时进行。它发生在任何(题项级别)被提交触发的响应处理之后。测试的结果变量的值总是在实现 outcomeRules 描述的说明之前, 重设为它们的默认值。因为结果处理发生在每次应试者提交响应, 测试级别的结果值可以用于在测试中激活测试级别的反馈, 或用于控制 preConditions 和 branchRules 的使用中的并发部分的行为。

结果变量的结构类似于 responseProcessing 的结构。

包含:

名称	多重性	说明
outcomeRule	[*]	

Abstract Class : outcomeRule

派生类: exitTest, include, lookupOutcomeValue, outcomeCondition, outcomeProcessingFragment, setOutcomeValue

相关类: outcomeIf, outcomeElse, outcomeProcessingFragment, outcomeProcessing, outcomeElseIf

Class : outcomeCondition

包含:

名称	多重性	说明
outcomeIf	[1]	
outcomeElseIf	[*]	
outcomeElse	[0..1]	

如果 outcomeIf 或 outcomeElseIf 给定的表达式评估为真, 那么它包含的子规则被遵循, 而且任何 outcomeElseIf 或 outcomeElse 部分在结果条件中被忽略。

如果 `outcomeIf` 或 `outcomeElseIf` 给定的表达式不为真，那么需要考虑下一个 `outcomeElseIf`，或者如果没有更多 `outcomeElseIf` 部分，那么 `outcomeElse` 的子规则被遵循（如果指定）。

Class : `outcomeIf`

相关类: `outcomeCondition`

包含:

名称	多重性	说明
<code>expression</code>	[1]	
<code>outcomeRule</code>	[*]	

`outcomeIf` 部分包含必须具有有效 `boolean` 基础类型和单一多重性的表达式。参见 `Expressions` 它也包含一系列子规则。如果表达式为真，那么子规则被处理，否则被跳过（包括表达式为 `NULL` 的情况），并且之后的 `outcomeElseIf` 或 `outcomeElse` 部分（如果有）被代替考虑。

Class : `outcomeElseIf`

相关类: `outcomeCondition`

包含:

名称	多重性	说明
<code>expression</code>	[1]	
<code>outcomeRule</code>	[*]	

`outcomeElseIf` 与 `outcomeIf` 用相同的方式定义。

Class : `outcomeElse`

相关类: `outcomeCondition`

包含:

名称	多重性	说明
<code>outcomeRule</code>	[*]	

Class : `exitTest (outcomeRule)`

14 测试级别反馈

Class : `testFeedback`

相关类: `assessmentTest`, `testPart`

属性名	多重性	类型	说明
<code>access</code>	[1]	<code>testFeedbackAccess</code>	测试反馈可以在结果处理（测试）过程中直接显示给应试者，或者在 <code>testPart</code> 或 <code>assessmentTest</code> 结束时的适当时候（称为 <code>atEnd</code> ）显示给应试者。 结果变量的值和 <code>showHide</code> 和 <code>identifier</code> 属性结合使用，从而检测反馈是否与 <code>feedbackElement</code> 采用了同样的方式显示
<code>outcomeIdentifier</code>	[1]	<code>identifier</code>	
<code>showHide</code>	[1]	<code>showHide</code>	

identifier	[1]	identifier	
title	[0..1]	string	不要求递送机制呈现 title 给应试者，但是可以这样做，例如，作为模式弹出窗口的标题或联合报告的子标题。

包含：

名称	多重性	说明
flowStatic	[*]	testFeedback 的内容不能包含任何交互。

枚举：testFeedbackAccess

attend

during

15 前提和分支

前提和分支规则是用于将适应性要素引入 assessmentTest 规范的高级概念，引入这些特征主要考虑的是：当依据应试者前面的作答使某些题项可以跳过时，辅助递送机制的复杂度可以降低。

前提和分支规则只适用于用线性模式浏览的测试部分。在非线性模式中忽略。拱形测试部分有效的通过非线性模式浏览，所以可以和前提和分支规则关联。

Class : precondition

相关类：

testPart, sectionPart

preCondition 是依附于 assessmentSection 或 assessmentItemRef 的简单表达式，如果题项呈现，assessmentSection 或 assessmentItemRef 评估必须为真。在测试中，前提在相关的题项、节或 testPart 将被应试者尝试的时候评估。它们随着选择和排序的规则而变化（参见测试结构），这些规则在测试开始之前或测试开始时都要遵循。

如果表达式评估为假，或者具有 NULL 值，跳过相关的题项或节。

包含：

名称	多重性	说明
expression	[1]	

Class : branchRule

相关类

testPart, sectionPart

branch-rule 是依附于在题项、节或部分呈现给应试者后被评估的 assessmentItemRef, assessmentSection 或 assessmentItemRef 的简单表达式。如果表达式评估为真，测试跳转到目标标识符指示的题项、节或部分。在题项或节的情况下，目标必须指示同一个 testPart 中还未展示的题项或者节，在 testPart 情况下，目标必须指示另一个 testPart。

上述的定义将通过线性测试部分浏览路线限制为树形。换句话说，循环是禁止的。在大多数情况下，重复可以通过使用一个节来实现，这个节选择 withReplacement 到合适循环上限，并且选择 preCondition

或 branchRule，当（或如果）达到一个确定的结果，preCondition 或 branchRule 可提前终止节。（例如，这个技术可以和一个或多个题项模版联合用于实现反复练习）。然而不支持无界限的循环。正在寻找关于这个方法是否限制过度的问题上的意见。

属性名	多重性	类型	说明
target	[1]	identifier	在用作目标标识符时，下面的值被预留并且具有特殊意义 EXIT_SECTION 跳过当前节所有剩余的字节至紧随其后的题项（或节） EXIT_TESTPART 立即结束当前 testPart EXIT_TEST 立即结束整个 assessmentTest。

包含：

名称	多重性	说明
expression	[1]	

16 表达式

派生类：and, anyN, baseValue, containerSize, contains, correct, customOperator, default, delete, divide, durationGTE, durationLT, equal, equalRounded, fieldValue, gt, gte, index, inside, integerDivide, integerModulus, integerToFloat, isNull, lt, lte, mapResponse, mapResponsePoint, match, member, multiple, not, null, numberCorrect, numberIncorrect, numberPresented, numberResponded, numberSelected, or, ordered, outcomeMaximum, outcomeMinimum, patternMatch, power, product, random, randomFloat, randomInteger, round, stringMatch, substring, subtract, sum, testVariables, truncate, variable

相关类：

and, gt, ordered, divide, setCorrectResponse, random, responseIf, substring, member, equalRounded, outcomeIf, integerToFloat, lookupOutcomeValue, setTemplateValue, integerDivide, gte, index, durationLT, contains, durationGTE, branchRule, lt, match, patternMatch, product, multiple, power, outcomeElseIf, setDefaultValue, customOperator, stringMatch, setOutcomeValue, templateDefault, not, templateElseIf, integerModulus, subtract, responseElseIf, anyN, preCondition, round, containerSize, inside, equal, or, isNull, templateIf, lte, sum, truncate, fieldValue, delete

表达式用于赋值给题项变量，以及按条件控制响应行为和模版处理。表达式可以简单引用题项变量的值，即基类型或层次表达式操作符定义的值集合中的一个常量值。与题项变量相同，每个表达式也可以有特殊值NULL。

16.1 内嵌普通表达式

本节描述一些用于处理常量、随机值和题项变量值的内嵌常规表达式。

Class : baseValue (expression)

属性名	多重性	类型	说明
baseType	[1]	baseType	基础类型值

最简单的表达式返回由baseType定义的值的集合中的一个值。

Class : variable (expression)

属性名	多重性	类型	说明
identifier	[1]	identifier	表达式查找已定义的或是内嵌的题项变量的值。结果具有服从加权结果（参见下文）类型提升的变量声明的基础类型和多重性。在结果处理中，从单独题项时间段得到的值可以通过题项标识符和前缀来查找。例如，为了获得在题项中作为Q01引用的SCORE变量的值，你可以使用具有Q01.SCORE的变量实例。查找相应变量的值时，它总是采用应试者最后一次提交给它的值。未提交的响应在表达式评估时不可提供。在同一个 assessmentItemRe 中被实例化多次的 f 题项中取得的题项变量的值，如果提交是同时的，值为最后一次提交实例，否则是未定义的。
weightIdentifier	[0..1]	identifier	将用于变量值的可选的权值。权值只定义在测试环境中（因此只在结果处理中），并且只定义在题项标识符前缀被应用于查找题项变量值时。权重标识符参考相应 assessmentItemRef 中的权重定义。如果没有找到匹配定义，权重被假定为 1.0。权值只被应用于具有基础类型 integer 和 float 的题项变量。如果题项变量是任何其它类型，那么权值被忽略。所有权值被视为具有 float 基础类型，通过将变量的值和相关权重相乘获得结果值。当将权值应用于具有 integer 基础类型的变量时，值服从类型提升，表达式的结果具有 float 基础类型。

Class : default (expression)

属性名	多重性	类型	说明
identifier	[1]	identifier	表达式查找题项变量的声明，并且返回相应的缺省值，如果没有声明默认值，返回 NULL。当用在结果处理中，题项标识符前缀（参见变量）可以用于从单独题项中获得默认值。

Class : correct (expression)

属性名	多重性	类型	说明
identifier	[1]	identifier	表达式查找响应变量的声明，并且返回相关正确响应，如果没有声明正确值，返回 NULL。用在结果处理时，题项标识符前缀（参见变量）可以用于从单独题项中获得正确响应。

Class : mapResponse (expression)

属性名	多重性	类型	说明
identifier	[1]	identifier	表达式查找响应变量的值，然后用已经声明的相关的映射将它转换。结果是单精度浮点数。如果响应变量具有单一多重性，那么返回值就是从映射得到的映射目标值。如果响应变量具有多值或有序多重性，那么返回值是映射目标值的和。这个表达式不能用于记录多重性的变量。

例如，如果映射将标识符 {A, B, C, D} 与值 {0, 1, 0.5, 0} 分别关联，那么 mapResponse 将映射单一值 'C' 到数字值 0.5，值的集合 {C, B} 到值 1.5。如果容器包含同一个值的多个实例，那么那个值将只被算作一次。继续上面的例子，{B, B, C} 将依然映射到 1.5，而不是 2.5。

Class : mapResponsePoint (expression)

属性名	多重性	类型	说明
identifier	[1]	identifier	表达式查找具有点基础类型的响应变量的值，并且用相关的区域映射转换它。转换与映射响应相似，是在每个区域轮流测试。映射容器时，每个区域只能被映射一次。例如，如果应试者确定落在同一个区域的两个点，那么映射值在计算出的总分里只加一次。

Class : null (expression)

Null 是返回 NuLL 值的简单表达式——Null 值被视为属于任何需要的基础类型。

Class : randomInteger (expression)

从指定的区间 [min, max] 内选择随机整数，对于某个整数 n，满足 $\text{min} + \text{step} * n$ 。例如，min=2, max=11, step=3，值 {2, 5, 8, 11} 是可以的。

属性名	多重性	类型	说明
min	[1]	integerOrTemplateRef=0	
max	[1]	integerOrTemplateRef	
step	[0..1]	integerOrTemplateRef	

Class : randomFloat (expression)

属性名	多重性	类型	说明
min	[1]	floatOrTemplateRef	
max	[1]	floatOrTemplateRef	

16.2 只用于结果处理的表达式

本节描述一些只能用在结果处理中的内嵌表达式，返回关于评估测试中引用的题项集合信息。

Abstract class : itemSubset

Derived classes:

numberCorrect, numberIncorrect, numberPresented, numberResponded, numberSelected, outcomeMaximum, outcomeMinimum, testVariables

这个类定义评估测试中选择的题项子集的概念。本属性定义必须被所有子集成员匹配的标准。它用于控制一些结果处理的表达式，以作为一个整体或任意一个子集返回关于测试的信息。

属性名	多重性	类型	说明
sectionIdentifier	[0..1]	identifier	如果指定，只有在 assessmentSection 中具有匹配标识符的题项的变量是匹配的。子节的题项包括在本定义中
includeCategory	[*]	identifier	如果指定，只包括具有匹配类别的题项的变量。
excludeCategory	[*]	identifier	如果指定，只包括没有匹配类别的题项的变量。

Class : testVariables (expression, itemSubset)

这个表达式只能用于结果处理，它同时在测试引用的题项子集中查找 itemVariable 的值。只有具有单一多重性的变量被考虑，所有 NULL 值被忽略。结果具有多值多重性和下面指定的基础类型。

属性名	多重性	类型	说明
variableIdentifier	[1]	identifier	要在每个题项中查找的变量标识符。如果测试将具有不同命名习惯的变量集合在一起，变量映射可以用于减少结果处理的复杂性，并且允许使用单一测试变量表达式。没有匹配变量的题项被忽略。
baseType	[0..1]	baseType	如果指定，只匹配声明为本基础类型的变量。如果忽略，匹配声明为 integer 或 float 基础类型的变量如果所有匹配值具有 integer 基础类型，结果的基础类型就是 integer，否则是服从类型提升的 float 和 integer 值。
weightIdentifier	[0..1]	identifier	如果指定，定义的权值用于 weightIdentifier 中描述的每个变量。这个属性的行为只有在基础类型是

float 或者忽略的条件下有效。当指定权值，表达式的结果总是具有 float 基础类型。

Class : outcomeMaximum (expression, itemSubset)

测试中引用的题项子集中的结果变量。只有具有单一多重性的变量被考虑。如果在给定子集中的任何题项没有声明最大值，结果为NULL，否则结果具有多值多重性和float基础类型。

属性名	多重性	类型	说明
outcomeIdentifier	[1]	identifier	同测试变量的 variableIdentifier 属性
weightIdentifier	[0..1]	identifier	同测试变量的 weightIdentifier 属性

Class : outcomeMinimum (expression, itemSubset)

这个表达式只能用在结果处理中，同时查找测试引用的题项子集的结果变量的normalMinimum值。结果具有多值多重性和float基础类型。

属性名	多重性	类型	说明
outcomeIdentifier	[1]	identifier	同测试变量的 variableIdentifier 属性
weightIdentifier	[0..1]	identifier	同测试变量的 weightIdentifier 属性

Class : numberCorrect (expression, itemSubset)

这个表达式只能用在结果处理中，计算给定子集中的题项的数量，对于这些子集，所有声明的响应变量具有定义的正确响应。结果是具有单一多重性的整数。

Class : numberIncorrect (expression, itemSubset)

这个表达式只能用在结果处理中，它计算给定子集中的题项的数量，对于这些子集，至少一个定义的响应变量不匹配它相关的correctResponse。只有所有声明的响应变量具有定义的正确值，并且被尝试至少一次的题项考虑。结果是具有单一多重性的整数

Class : numberResponded (expression, itemSubset)

这个表达式只能用在结果处理中，它计算给定子集中被尝试过（至少一次）并且给与响应的题项的数量。换句话说，对于这个题项，至少一个声明的响应具有一个不同于它的声明默认值（典型的NULL）的值。结果是具有单一多重性的整数。

Class : numberPresented (expression, itemSubset)

这个表达式只能用在结果处理中，它计算给定子集中被尝试过（至少一次）题项的数量。换句话说，用户进行交互的题项，不管他们是否提供了响应。结果是具有单一多重性的整数。

Class : numberSelected (expression, itemSubset)

这个表达式只能用在结果处理中，它计算给定子集中被选择呈现给应试者的题项的数量，不管应试者是否尝试。结果是具有单一多重性的整数。

16.3 操作符

操作符类的作用是按照指定方式修改单一子表达式或联合表达式或更多子表达式而获得它们值。操作符不直接影响题项变量的值。所有操作符具有基础类型和多重性，尽管可能依附于它们包含的子表达式。

Class : multiple (expression)

Contains : expression [*]

多重操作符取得 0 个或多个子表达式，它们都具有单一或多值多重性。尽管子表达式可能属于任何基础类型，但它们必须都是相同基础类型的。结果是具有多值多重性的包含子表达式值的容器。具有多值多重性的子表达式具有它们单独的值加在结果中：容器不能包含其他容器。例如将 A, B 用于 {C, D}，多值操作符结果是 {A, B, C, D}。所有具有 NULL 值的子表达式被忽略。如果没有给定子表达式（或者所有值都是 NULL），那么结果是 NULL。

Class : ordered (expression)

Contains : expression [*]

排序操作符取得 0 个或多个子表达式，都具有单一或有序多重性。虽然子表达式可以属于任何基础类型，但它们必须都是相同基础类型。结果是包含子表达式值的具有有序多重性的容器，具有有序多重性的子表达式有它们单独的值加入（按顺序）结果：容器不能包含其他容器。例如，将 A, B, {C, D}，排序操作符结果是 {A, B, C, D}。注意排序操作符结果不会是空容器。所有具有 NULL 值的子表达式被忽略。如果没有给定子表达式（或者所有值都是 NULL），那么结果是 NULL。

Class : containerSize (expression)

Contains : expression [1]

containerSize 操作符取得一个具有任何基础类型和多值或有序多重性的子表达式。结果是子表达式中的值的整数，也就是容器的大小。如果子表达式为 NULL，结果是 0。例如，这个操作符可以用于决定在多响应的选择交互中有多少选项被选择。

Class : isNull (expression)

Contains : expression [1]

isNull 操作符取得一个具有任何基础类型和多重性的子表达式。如果子表达式为 NULL，结果是单一值为真的 boolean，否则是假。注意，空容器和空字符都被视为 NULL。

Class : index (expression)

Attribute : n [1]: integer

Contains : expression [1]

index 操作符取得一个具有有序容器值和任何基础类型的子表达式。结果是容器的第 n 个值并且具有与子表达式相同的基础类型，但是单一多重性的。容器的第一个值 index 是 1，第二个是 2，以此类推。N 必须是正整数。如果 n 超过容器中值的数量，那么 index 操作符的结果为 NULL。

Class : fieldValue (expression)

Attribute : fieldIdentifier [1]: identifier

域的标识符

Contains : expression [1]

field-value 操作符取得一个具有记录容器值的子表达式。结果是具有特定 fieldIdentifier 的域值。如果没有具有 fieldIdentifier 的域，那么操作符的结果为 NULL。

Class : random (expression)

Contains : expression [1]

random 操作符取得具有多值或有序的容器值和任何基础类型的子表达式。结果是从容器中随机选取的单一值。结果与子表达式具有相同的基础类型，但是单一多重性。如果子表达式为 NULL，那么结果也是 NULL。

Class : member (expression)

Contains : expression [2]

member 操作符取得两个必须具有相同基础类型的子表达式。第一个子表达式必须具有单一多重性，第二个必须是多重或有序容器。如果第一个子表达式给定的值属于第二个子表达式定义的容器，结果是值为真的单一 boolean。如果子表达式为 NULL，那么操作符的结果为 NULL。

Member 操作符不应该用于具有 float 基础类型的子表达式，因为值比较的定义较困难。也不能用于具有 duration 基础类型的子表达式。

Class : delete (expression)

Contains : expression [2]

delete 操作符取得两个必须都是相同基础类型的子表达式。第一个子表达式必须有单一多重性，第二个必须是多值或者有序容器。结果是将第二个容器中所有第一个子表达式的实例都去除而得到的新的容器。例如，当应用 A 和 {B, A, C, A} 时，结果是容器 {B, C}。如果任何子表达式为 NULL，操作符的结果为 NULL。用于 member 操作符的限制也适用于 delete 操作符。

Class : contains (expression)

Contains : expression [2]

contains 操作符取得两个必须都具有相同基础类型和多值或有序多重性的子表达式。如果第一个表达式给定的容器包含第二个表达式给定的值，结果是单一值为真的 boolean，否则为假。注意 contains 操作符依据两个表达式多重性，以不同的方式工作。对于无序容器，不考虑顺序的比较值，例如，[A, B, C] 包含 [C, A]。注意，[A, B, C] 不包含 [B, B]，但是 [A, B, B, C] 包含。对于有序容器，第二个子表达式必须是第一个的严格的子序列。换句话说，[A, B, C] 不包含 [C, A]，但它包含 [B, C]。如果子表达式为 NULL，那么操作符的结果为 NULL。像 member 操作符一样，contains 操作符不应该用于具有 float 基础类型的子表达式，而且不能用于具有 duration 基础类型的子表达式

Class : substring (expression)

Contains : expression [2]

Substring 操作符取得两个必须都具有有效 string 基础类型和单一多重性的子表达式。如果第一个表达式是第二个表达式的子串，结果是值为真的单一 boolean 值，否则为假。如果任何一个子表达式为 NULL，那么操作符的结果是 NULL。

Attribute : caseSensitive [1]: boolean = true

GB/T×××—××××

用于控制匹配子串时是否大小写敏感。如果为真，那么匹配是大小写敏感的，例如，“Hell”不是“Shell”的子串。如果为假，那么匹配是大小写不敏感的，“Hell”是“Shell”的子串。

Class : not (expression)

Contains : expression [1]

not操作符取得具有boolean基础类型和单一多重性的单一子表达式。结果是单一boolean值，通过子表达式值的逻辑否定获得。如果子表达式为NULL，那么not操作符结果也是NULL。

Class : and (expression)

Contains : expression [1..*]

and操作符取得一个或多个具有boolean基础类型和单一多重性的子表达式。如果所有子表达式为真，结果是值为真的单一boolean值，如果它们之中任何一个为假，值为假。如果一个或多个子表达式为NULL并且所有其它的为真，那么操作符结果也为NULL。

Class : or (expression)

Contains : expression [1..*]

or操作符取得一个或多个具有boolean基础类型和单一多重性的子表达式。如果任何子表达式为真，结果是值为真的单一boolean值，如果它们全为假，值为假。如果一个或多个子表达式为NULL并且所有其它的为假，那么操作符结果也为NULL。

Class : anyN (expression)

Contains : expression [1..*]

anyN操作符取得一个或多个具有boolean基础类型和单一多重性的子表达式。如果至少min个子表达式为真，并且至多max个子表达式为真，结果是值为真的单一boolean值。如果多于n-min个子表达式为假（n是子表达式的总数），或者多于max个子表达式为真，那么值为假。如果一个或多个子表达式为NULL，那么可能这些条件都不满足，在这种情况下。操作符结果为NULL。例如，如果min为3，max为4，子表达式的值为{true, true, false, NULL}，那么操作符结果为NULL，然而{true, false, false, NULL}结果为假，{true, true, true, NULL}结果为真。结果NULL表示操作符的正确值不能被决定。

属性名	多重性	类型	说明
min	[1]	integerOrTemplateRef	必须为真的子表达式的最小数量
max	[1]	integerOrTemplateRef	必须为真的子表达式的最大数量

Class : match (expression)

Contains : expression [2]

match操作符取得两个具有相同基础类型和多重性的子表达式。如果两个代表相同值，结果是值为真的单一boolean值，如果不是，值为假。如果任何一个子表达式为NULL，操作符结果为NULL。match操作符不能和例如数值相等的广义等值概念相混淆。为了避免混淆，match操作符不应该用于比较具有float基础类型的子表达式，并且不能用于具有duration基础类型的子表达式。

Class : stringMatch (expression)

Contains : expression [2]

stringMatch操作符取得两个必须具有单一和string基础类型的子表达式。如果两个string根据下文属性定义的比较规则匹配，结果是值为真的单一boolean值，如果不是，值为假。如果任何一个子表达式为NULL，那么操作符结果为NULL。

属性名	多重性	类型	说明
caseSensitive	[1]	Boolean	匹配实现时是否大小写敏感
substring	[0..1]	boolean = false	现在不建议使用本属性，应该用substring操作符代替。如果为真，那么如果第一个string包含第二个，比较返回真，否则它返回真当且仅当它们完全匹配。如果两个string根据下文属性定义的比较规则匹配，结果是值为真的单一boolean值，如果不是，值为假。

Class : patternMatch (expression)

Contains : expression [1]

patternMatch操作符取得一个具有string基础类型和单一多重性的子表达式。如果子表达式匹配模式给定的正规表达式，结果是值为真的单一boolean值，如果不是，值为假。如果子表达式为NULL，那么操作符结果为NULL。

Attribute : pattern [1]: stringOrTemplateRef

参见正则表达式。

Class : equal (expression)

Contains : expression [2]

equal操作符取得两个都具有单一多重性和数字基础类型和的子表达式。如果两个表达式数值相等，结果是值为真的单一boolean值，如果不是，值为假。如果任何一个子表达式为NULL，那么操作符结果为NULL。

属性名	多重性	类型	说明
toleranceMode	[1]	toleranceMode	当比较两个浮点数值时，经常需要有容忍度以确保舍入错误不会在计分中引入错误。容忍度模式决定比较是否精确地完成，使用绝对区间或相对区间。
tolerance	[0..2]	floatOrTemplateRef	如果容忍度模式是绝对或相对，那么必须设定容忍度。容忍度包括两个正的数字，t0和t1，定义下界和上界。如果只给定一个值，它用于两个。如果第二个表达式的值y在由第一个值x定义的如下区间：x-t0, x+t1，在绝对模式下，比较的结果为真。在相对模式下，t0和t1被视为百分数，如下区间被替代使用： $x*(1-t0/100)$, $x*(1+t1/100)$

includeLowerBound	[0..1]	boolean = true	控制比较时下界是否被包括
includeUpperBound	[0..1]	boolean = true	控制比较时上界是否被包括

Class : equalRounded (expression)

Contains : expression [2]

equalRounded操作符取得两个都具有单一多重性和数字基础类型的子表达式。如果两个表达式在舍入之后数值相等，结果是值为真的单一boolean值，如果不是，值为假。如果任何一个子表达式为NULL，那么操作符结果为NULL。

属性名	多重性	类型	说明
roundingMode	[1]	roundingMode = significantFigures	数字被舍入到给定数字 significantFigures 或 decimalPlaces。
figures	[1]	integerOrTemplateRef	舍入后的数值。例如，如果两个值为 1.56 和 1.6，使用 significantFigures 模式， figure=2，那么结果为真。

Enumeration: roundingMode

significantFigures

decimalPlaces

Class : inside (expression)

Contains : expression [1]

inside 操作符取得一个具有基础类型和的子表达式。如果给定点在形状和坐标定义的区域，结果是值为真的单一boolean值。如果子表达式为容器，任何点在区域内，结果为真。如果任何一个子表达式为NULL，那么操作符结果为NULL。

属性名	多重性	类型	说明
shape	[1]	shape	区域的形状。
coords	[1]	coords	区域的大小和位置，与形状联合被解释。

lass : lt (expression)

Contains : expression [2]

lt 操作符取得两个都具有单一多重性和数值基础类型和的子表达式。如果第一个表达式数值上小于第二个，结果是值为真的单一boolean值，如果它大于或等于第二个，值为假。如果任何一个子表达式为NULL，那么操作符结果为NULL。

Class : gt (expression)

Contains : expression [2]

gt 操作符取得两个都具有单一多重性和数值基础类型和的子表达式。如果第一个表达式数值上大于第二个，结果是值为真的单一boolean值，如果数值上小于或等于第二个，值为假。如果任何一个子表达式为NULL，那么操作符结果为NULL。

Class : lte (expression)

Contains : expression [2]

lte 操作符取得两个都具有单一多重性和数值基础类型和的子表达式。如果第一个表达式数值上小于第二个，结果是值为真的单一 boolean 值，如果它大于或等于第二个，值为假。如果任何一个子表达式为 NULL，那么操作符结果为 NULL。

Class : gte (expression)

Contains : expression [2]

gte 操作符取得两个都具有单一多重性和数值基础类型和的子表达式。如果第一个表达式数值上小于或等于第二个，结果是值为真的单一 boolean 值，如果它大于第二个，值为假。如果任何一个子表达式为 NULL，那么操作符结果为 NULL。

Class : durationLT (expression)

Contains : expression [2]

durationLT 操作符取得两个都具有单一多重性和 duration 基础类型和的子表达式。如果第一个 durationd 短于第二个，结果是值为真的单一 boolean 值，如果它短于，或等于第二个，值为假。如果任何一个子表达式为 NULL，那么操作符结果为 NULL。没有' durationLTE' 或' durationGT'，因为 duration 的等同性在 duration 允许的变量精确度条件下没有意义。由于 duration 值通过切断获得，而不是舍入，只有小于或大于等于不等式有意义。例如，如果在报告 duration 到精度为 epsilon 秒的系统中，我们要决定应试者是否消耗小于 10 秒来完成任务，那么等于 10 的值会包括所有在 [10, 10+epsilon) 区间的 duration。

Class : durationGTE (expression)

Contains : expression [2]

durationGTE 操作符取得两个都具有单一多重性和 duration 基础类型和的子表达式。如果第一个 durationd 长于（或等于，在上文描述的切断所施加的界限内）第二个，结果是值为真的单一 boolean 值，如果它短于第二个，值为假。如果任何一个子表达式为 NULL，那么操作符结果为 NULL。参见 durationLT 以获得关于测试 duration 等值性的信息。

Class : sum (expression)

Contains : expression [1..*]

sum 操作符取得一个或多个都具有单一多重性和数值基础类型和的子表达式。结果是对应于子表达式数值的总和的单精度浮点数，或者，如果所有子表达式为 integer 类型，为单精度 integer。如果任何子表达式为 NULL，那么操作符结果为 NULL。

Class : product (expression)

Contains : expression [1..*]

product 操作符取得一个或多个都具有单一多重性和数值基础类型和的子表达式。结果是对应于子表达式数值乘积的单精度浮点数，或者，如果所有子表达式为 integer 类型，为单精度 integer。如果任何子表达式为 NULL，那么操作符结果为 NULL。

GB/T×××—××××

Class : subtract (expression)

Contains : expression [2]

subtract 操作符取得两个都具有单一多重性和数值基础类型和的子表达式。结果是对应于第一个值减去第二个的单精度浮点数，或者，如果所有子表达式为 integer 类型，为单精度 integer。如果任何子表达式为 NULL，那么操作符结果为 NULL。

Class : divide (expression)

Contains : expression [2]

divide 操作符取得两个都具有单一多重性和数值基础类型和的子表达式。结果是对应于第一个表达式除以第二个单精度浮点数，或者，如果所有子表达式为 integer 类型，为单精度 integer。如果任何子表达式为 NULL，那么操作符结果为 NULL。题项开发者应该尽最大努力来确保第二个表达式的值不是 0，然而，如果它是 0，或者结果值超出 float（不包括正无穷和负无穷）定义的值集合，那么操作符结果应该是 NULL。

Class : power (expression)

Contains : expression [2]

power 操作符取得两个都具有单一多重性和数值基础类型和的子表达式。结果是对应于第一个表达式的第二个的乘方的单精度浮点数。如果任何子表达式为 NULL，那么操作符结果为 NULL。如果结果值超出 float（不包括正无穷和负无穷）定义的值集合，那么操作符结果应该是 NULL。

Class : integerDivide (expression)

Contains : expression [2]

integer divide 操作符取得两个都具有单一多重性和 integer 基础类型和的子表达式。结果是对应于第一个表达式 (x) 除以第二个表达式 (y)，舍去小数点后的最大整数，即 $i \leq (x/y)$ 的单精度整数。如果 y 是 0，或者任何子表达式为 NULL，那么操作符结果为 NULL。

Class : integerModulus (expression)

Contains : expression [2]

integer modulus 操作符取得两个都具有单一多重性和 integer 基础类型和的子表达式。结果是对应于第一个表达式 (x) 除以第二个表达式 (y) 的余数的单精度整数。如果 z 是 integerDivide 操作符的相应结果，那么结果是 $x-z*y$ ，如果 y 是 0，或者任何子表达式为 NULL，那么操作符结果为 NULL。

Class : truncate (expression)

Contains : expression [1]

truncation 操作符取得一个具有单一多重性和 float 基础类型和的子表达式。结果是通过向 0 截取子表达式的值形成的基础类型 integer。例如，6.8 成为 6，-6.8 成为 -6。如果子表达式为 NULL，那么操作符结果为 NULL。

Class : round (expression)

Contains : expression [1]

round 操作符取得一个具有单一多重性和 float 基础类型和的子表达式。结果是通过四舍五入子表达式的值形成的基础类型 integer。对于在 $[n-0.5, n+0.5)$ 区间内的输入值，结果是 integer n。换句话说，6.8 和 6.5 都入到 7，6.40 舍为 6，-6.5 入到-6。如果子表达式为 NULL，那么操作符结果为 NULL。

Class : integerToFloat (expression)

Contains : expression [1]

该操作符取得一个具有单一多重性和 float 基础类型和的子表达式。结果是具有相同数值的基础类型 float。如果子表达式为 NULL，那么操作符结果为 NULL。

Class : customOperator (expression)

custom操作符提供用于定义本标准目前不支持的操作符的扩展机制。

属性名	多重性	类型	说明
class	[0..1]	identifier	class属性允许命名简单子类。子类的定义是工具细节的，可以在工具名和 工具版本中推断。
definition	[0..1]	uri	在全局名字域中识别定制操作符定义的 URI。除class和定义属性之外，子类可以添加任何数量的自身属性。

Contains : expression [*]

定制操作符可以取得任何数量的参数任何类型的子表达式。建议customOperator用于帮助链接本标准定义的处理规则到基于处理机制的web服务实例。鼓励执行者用本方法实验，以共享关于它们解答的信息，以帮助决定达到这类处理的最好的方式。

17 题项和测试片段

题项片断是独立于题项而操纵题项的片断。类似的，测试片断是独立于依附它的测试而操纵的测试的片断。片断被打包为单独的资源，可以被独立传输。片断被定义为任何从下列抽象类继承来的类的实例：flow, responseRule, sectionPart 或 outcomeRule。例如，题项片断可以是 div 类的实例代表的 itemBody 的一部分。

Class : include (flow, outcomeRule, responseRule, sectionPart)

用 Xinclude 机制包含片断。include 的实例被视为它是一个被 include 元素的 href 属性引用的片断根元素的实例。为了本标准的目标，不能使用 XInclude 规范定义的 xpointer 机制。同样，所有被包括的片断必须用 xml 解析。

该技术类似于媒体对象（使用对象）的包换，但是允许包含与本标准一致的数据，具体来说，它允许从外部定义片断引入交互、静态内容、处理规则，或测试级别上整个节。

当包含外部定义片断时，片段的内容必须满足包含它的题项的上下文规范的要求。例如，片断包含的交互必须正确受限于在题项中声明的响应变量。

Class : responseProcessingFragment (responseRule)

包含:

名称	多重性	说明
responseRule	[*]	

responseProcessingFragment 是一些响应规则的简单组合, 为了使这些响应规则可以作为一个单独的资源管理, 它们被分在一个组里。responseProcessingFragment 不应该用于任何其他目的。

注意响应处理模版允许系统在不解析单独相应处理规则的情况下实现响应处理。另一方面, 包含对外部定义响应处理片断的引用的 responseProcessing 元素必须被解析以决定要实现的行为。

Class : outcomeProcessingFragment (outcomeRule)

包含:

名称	多重性	说明
outcomeRule	[*]	

outcomeProcessingFragment 是一些 outcomeRules 的简单组合, 为了使这些 outcomeRules 可以作为一个单独的资源来管理, 它们被分在一个组里。outcomeProcessingFragment 不应该用于任何其他目的。

18 基本数据类型

数据类型: boolean

Boolean 值为真或假。注意类似"Yes", "TRUE", "1"的词法绑定字符串超出本文档的范围。

数据类型: coords

coords 类型提供决定由相应形状定义的区域尺寸和位置坐标。

坐标自身是有序的长度列表 (定义在[XHTML])。每个长度值的解释依赖于下文相关形状的值:

•rect: left-x, top-y, right-x, bottom-y.

•circle: center-x, center-y, radius.

注意: 当半径值是百分比值, 用户代理应该基于相关对象的宽度和长度计算最终半径值。半径应该是两个值中较小的一个。

•poly: x1, y1, x2, y2, ..., xN, yN.

第一个和最后一个 x 和 y 坐标对应该相同以闭合多边形。当这些坐标值不同时, 用户代理应该推断附加的坐标对以闭合多边形。

•ellipse: center-x, center-y, h-radius, v-radius.

注意: 不推荐椭圆形, 因为它在[XHTML]中未定义。

•default: no coordinates should be given.

数据类型: date

完全指定的历法日期, 包括从[ISO8601]定义的参考系统中得到的年、月和月中的日。合法的年的区间是 0001-9999。年的值没有时区信息。

数据类型: datetime

完全指定的从[ISO8601]定义的参考系统中得到的历法日期和时间。合法的年的区间是 0001-9999。

数据类型: duration

一段时间, 以秒为单位测量。

数据类型：float

IEEE 双精度 64 位浮点类型。

数据类型：identifier

标识符是对题项中另一个对象的逻辑引用，例如 itemVariable 或 choice。标识符是必须开始于字母或下划线('_')，并且只包含字母、下划线、连字号('-')、句号('.', a.k.a. full-stop)，数字、连接符和扩展符的一串字符。包含句号字符的标识符为将来的使用预留。字符类 Letter, Digit, CombiningChar 和 Extender 定义在 Extensible Markup Language (XML) 1.0 (Second Edition) [XML]中。特别注意标识符不能包含冒号(':')字符。为兼容版本 1，标识符应该具有不多于 32 个字符。标识符大小写敏感。

数据类型：integer

Integer 值是[-2147483648,2147483647]区间内的整数。这是二进制补码 32 位整数的区间。

数据类型：language

自然语言标识符。

数据类型：length

length 数据类型在[XHTML]中定义。

数据类型：mimeType

Mime 类型（类型和子类型）的集合，定义在[RFC2045]。

枚举：orientation

vertical

horizontal

枚举：shape

shape 的值总是伴随坐标值（参见 coords 和相关提供用于解释它们的上下文的图象）。

Default

default shape 指的是相关图象的整个区域。

Rect

矩形区域

Circle

圆形区域

Poly

任意多边形区域

Ellipse

不建议使用这个值，但是为了和 QTI 标准版本 1 兼容，所以包含了它。系统应该用 circle 或 poly 来代替。

数据类型：string

string 值是所有字符的序列。字符是定义在 Extensible Markup Language (XML) 1.0 (Second Edition)中的 Char 类的任何值。

GB/T×××—××××

数据类型: `string256`

长度限制在 256 个字符的 `string` 值 (与 `string` 相同)。

数据类型: `styleclass`

引用类定义时使用的类型, 例如在样式表中。类名不能包含空格。

数据类型: `uri`

定义在[URI]中的通用资源标识符。

数据类型: `valueType`

用于呈现任何 `baseType` 单一值的简单类型, `baseType` 由评估变量数据模型定义

枚举: `view`

`author`

`candidate`

`proctor`

有时被称为监考人

`scorer`

`testConstructor`

`tutor`
