IMS Question and Test Interoperability: ASI Outcomes Processing

Version 1.2 Final Specification

IPR and Distribution Notices

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

IMS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on IMS's procedures with respect to rights in IMS specifications can be found at the IMS Intellectual Property Rights web page: http://www.imsglobal.org/ipr/imsipr_policyFinal.pdf.

If you wish to distribute this document or use this document to implement a product or service, you must complete a valid license registration with IMS and receive an email from IMS granting the license. To register, follow the instructions on the IMS website: http://www.imsglobal.org/specificationdownload.cfm.

This document may be copied and furnished to others by Licensee organizations registered on the IMS website provided that the above copyright notice and this paragraph are included on all such copies. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to IMS, except as needed for the purpose of developing IMS specifications, under the auspices of a chartered IMS work group.

Use of this specification to develop products or services is governed by the license with IMS found on the IMS website: http://www.imsglobal.org/question/qtiv1p2/qtiv1p2speclicense.html.

The limited permissions granted above are perpetual and will not be revoked by IMS or its successors or assigns.

THIS SPECIFICATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE CONSORTIUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS SPECIFICATION.

Document Name: IMS Question and Test Interoperability: ASI Outcomes Processing
Date: 11 February 2002

Table of Contents

1. Introduction

1.1 Question & Test Interoperability Overview

The IMS Question & Test Interoperability (QTI) specification describes a basic structure for the representation of question (item) and test (assessment) data and their corresponding results reports [QTI, 02i]. Therefore, the specification enables the exchange of this item, assessment and results data between Learning Management Systems, as well as content authors and, content libraries and collections. The IMS QTI specification is defined in XML to promote the widest possible adoption. XML is a powerful, flexible, industry standard mark-up language used to encode data models for Internet-enabled and distributed applications. The QTI specification is extensible and customizable to permit immediate adoption, even in specialized or proprietary systems. Leading suppliers and consumers of learning products, services and content contributed time and expertise to produce this final specification. The IMS QTI specification, like all IMS specifications, does not limit product designs by specifying user interfaces, pedagogical paradigms, or establishing technology or policies that constrain innovation, interoperability, or reuse.

The 'Outcomes Processing' specification contains the description of how the aggregated scores at the Assessment and Section levels can be derived. These scoring outcomes are based upon the child Sections and/or Items. A wide range of scoring algorithms is supported through the usage of a predefined set of parameterized instructions; these avoid the realization of the algorithms within the XML. This document contains the relevant information model, XML binding and best practices guidance but it should be read in the context of the core ASI documents.

1.2 Scope & Context

This document is the IMS Question & Test Interoperability ASI Outcomes Processing Specification and must be read in conjunction with the core documents:

- IMS QTI: ASI Information Model [QTI, 02a];
- IMS QTI: ASI XML Binding [QTI, 02b];
- IMS QTI: ASI Best Practice & Implementation Guide [QTI, 02c].

It defines the outcomes processing features that are to be applied to the Section and/or Assessments. These new Assessment and Section features are **not** backwards compatible with the other versions of the IMS QTI specifications. These new features have no effect on the IMS QTILite specification [QTI, 02h] because that refers only to the Item data structure.

1.3 Structure of this Document

The structure of this document is:

2. Information Model	The description of the information model of the outcomes processing component for the full IMS QTI ASI;
3. XML Binding	The description of the XML binding of the outcomes processing component for the full IMS QTI ASI;
4. Best Practice & Implementation Guide	The description of the best practices and implementation guide for the outcomes processing component of the full IMS QTI ASI;
Appendix A – Glossary of Terms	A brief description of all of the elements and attributes that have been used to support outcomes processing;
Appendix B – In-Built Scoring Algorithms	A description of the operation of each of the default scoring algorithms;
Appendix C – Logic Rules	The logic rules that are supported by the *or_object*, *and_object* and *not_object* elements.

1.4 Nomenclature

API	Application Programming Interface
ASI	Assessment, Section, Item
CBT	Computer Based Training
DTD	Document Type Definition
QTI	Question & Test Interoperability
VLE	Virtual Learning Environment
W3C	World Wide Web Consortium
XML	Extensible Mark-up Language

1.5 References

[IMS, 01] *IMS Persistent, Location-Independent Resource Identifier Implementation Handbook*, M.McKell, Version 1.0, IMS, April 2001.

[QTI, 02a] *IMS Question & Test Interoperability: ASI Information Model Specification*, C.Smythe, E.Shepherd, L.Brewer and S.Lay, Final Specification, Version 1.2, IMS, February 2002.

[QTI, 02b] *IMS Question & Test Interoperability: ASI XML Binding Specification*, C.Smythe, E.Shepherd, L.Brewer and S.Lay, Final Specification, Version 1.2, IMS, February 2002.

[QTI, 02c] *IMS Question & Test Interoperability: ASI Best Practice & Implementation Guide*, C.Smythe, E.Shepherd, L.Brewer and S.Lay, Final Specification, Version 1.2, IMS, February 2002.

[QTI, 02d] *IMS Question & Test Interoperability: ASI Selection & Ordering Specification*, C.Smythe, L.Brewer and S.Lay, Final Specification, Version 1.2, IMS, February 2002.

[QTI, 02e] *IMS Question & Test Interoperability: Results Reporting Information Model*, C.Smythe, L.Brewer and S.Lay, Final Specification, Version 1.2, IMS, February 2002.

[QTI, 02f] *IMS Question & Test Interoperability: Results Reporting XML Binding*, C.Smythe, L.Brewer and S.Lay, Final Specification, Version 1.2, IMS, February 2002.

[QTI, 02g] *IMS Question & Test Interoperability: Results Reporting Best Practice & Implementation Guide*, C.Smythe, L.Brewer and S.Lay, Final Specification, Version 1.2, IMS, February 2002.

[QTI, 02h] *IMS Question & Test Interoperability: QTILite Specification*, C.Smythe, E.Shepherd, L.Brewer and S.Lay, Final Specification, Version 1.2, IMS, February 2002.

[QTI, 02i] *IMS Question & Test Interoperability: Overview*, C.Smythe, E.Shepherd, L.Brewer and S.Lay, Final Specification, Version 1.2, IMS, February 2002.

2. Information Model

2.1 Exchanging ASI Objects

The possible advanced Assessment structures that can be exchanged using QTI are shown in Figure 2.1:

- The Assessment could contain a single Section block (Figure 2.1a);
- The Assessment could consist of a reference to an external Section block (Figure 2.1b);
- The Assessment could consist of a mixture of contained and externally referenced Section blocks (Figure 2.1c). There is no preferential order in the blocks and references.

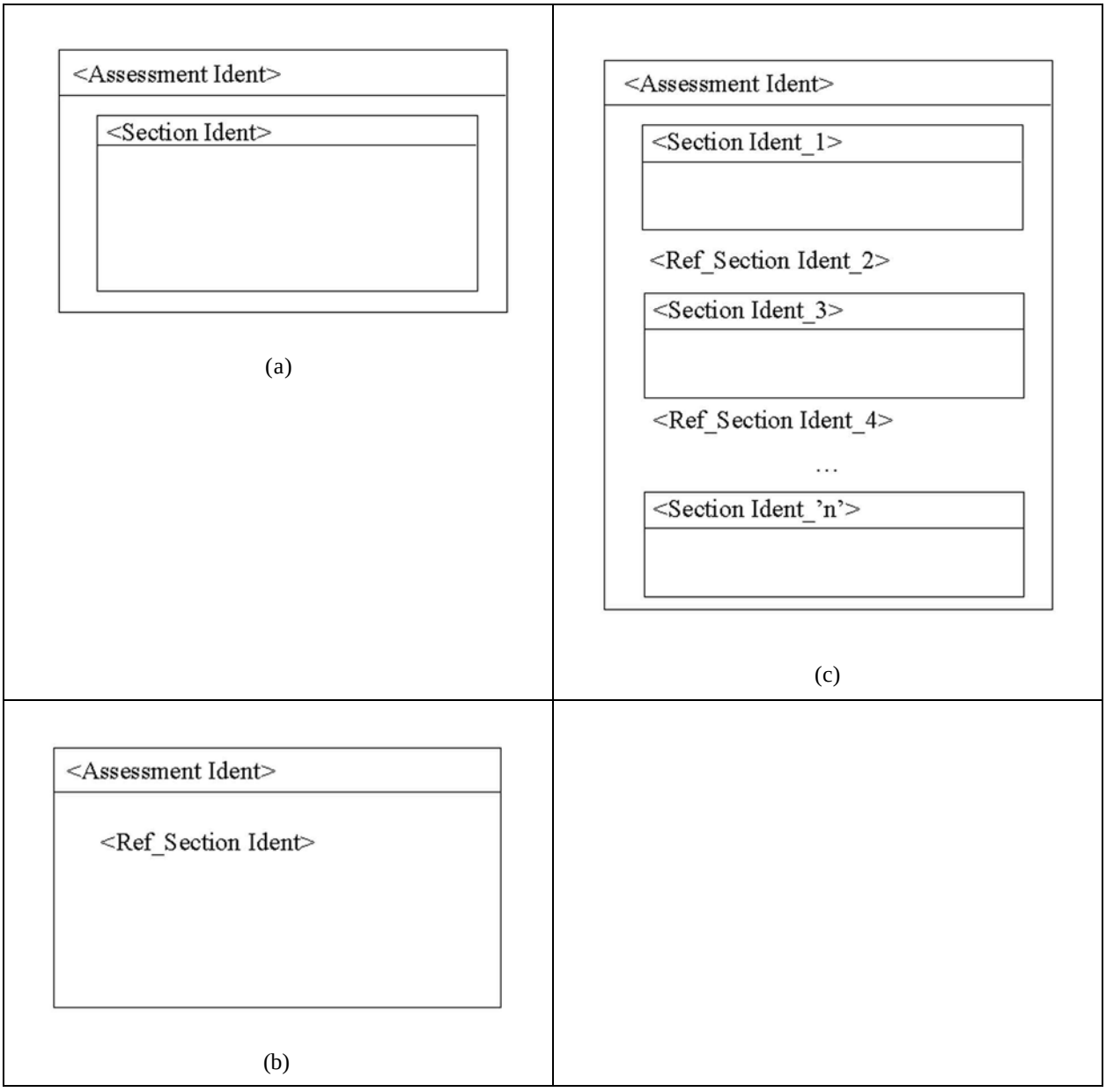
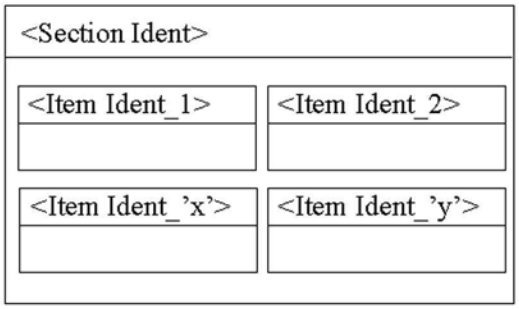


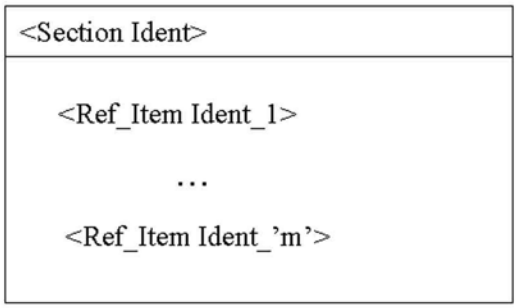
Figure 2.1 Possible <assessment> structures.

In the case of Sections, some of the possible data structures that can be exchanged are shown in Figure 2.2:

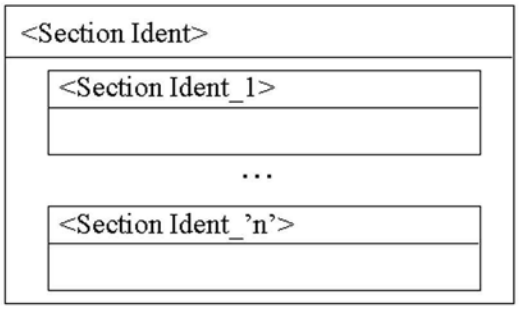
- The Section could contain one or more Items (Figure 2.2a);
- The Section could contain one or more references to external Items (Figure 2.2b);
- The Section could contain one or more Sections (Figure 2.2c);
- The Section could contain one or more references to external Sections (Figure 2.2d);
- The Section could contain a mixture of blocks and references to Section and Items. There is no constraint on the order in which the Sections and Items are declared/referenced i.e. interleaving is supported.



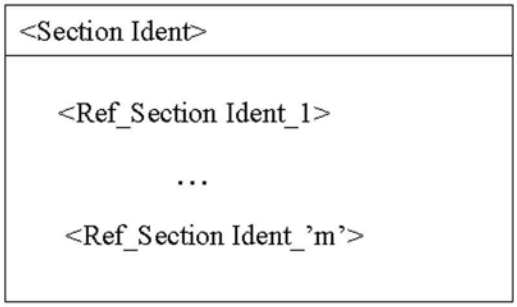
(a)



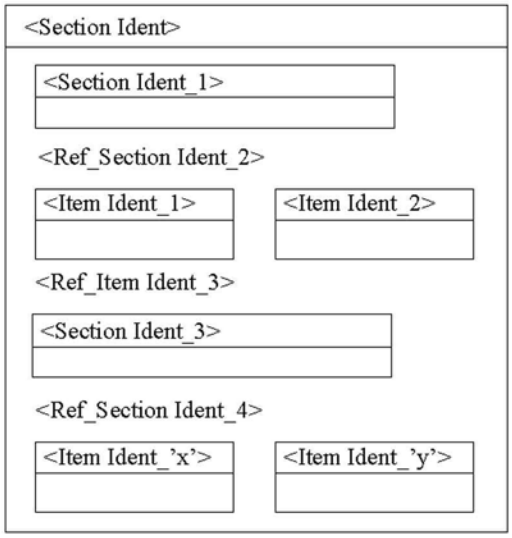
(b)



(c)



(d)



(e)

Figure 2.2 Possible <section> structures.

The wide range of different data structures that can be exchanged is especially significant when developing mechanisms that support 'Outcomes Processing'. 'Outcomes Processing' is responsible for:

- The aggregation of scores assigned to the Items and/or Sections to create a score or set of scores for the parent Section. The ways in which this roll-up is achieved reflects the different scoring algorithms;
- The aggregation of scores assigned to Sections to create a score or set of scores for the parent Assessment. The ways in which this roll-up is achieved reflects the different scoring algorithms;
- Any assignment of scores based upon more than one Item and/or Section.

It is important to note that the type of scoring algorithm that is to be used is not mandated by the IMS specification. Instead, the specification is capable of supporting most of the available scoring algorithms. Also, the algorithms themselves are not encoded within the specification itself. Instead the type of algorithm is specified and the necessary parametric information supplied to enable the assessment engine to execute the corresponding algorithm.

2.2 Use Cases

The following representative use-cases have been identified for support within QTI ASI outcomes processing V1.2:

a) Multiple-choice quiz – this is a simple "end of chapter" quiz given by an LMS. The scores are reported back to the student for self-evaluation, so there is no particular need for high reliability. All of the questions are discrete (single response) questions whose outcomes are a dichotomous variable e.g. multiple-choice questions. For the sake of being definite we assume there are 10 Items and that we have a defined outcome, "CORRECT" for each Item. This takes on the value "True" if the response was correct and "False' if the response was not correct. All items are given equal weight. We wish to know, (1) the number of correct items, (2) the percentage of total score obtained by a candidate and (3) the percentage of items actually attempted which were correct;

b) True/false quiz – this is a simple "end of chapter" quiz given by an LMS. The scores are reported back to the student for self-evaluation, so there is no particular need for high reliability. All of the questions are discrete true/false. For the sake of being definite we assume there are 10 Items and that we have a defined outcome,

"isCorrect" for each Item. This takes on the value '1' if the response was correct and "-1' if the response was not correct. All items are given equal weight. We wish to know, (1) the number of correct items, (2) the total score and (3) the weighted total score;

c) Multiple-response quiz – this is a simple "end of chapter" quiz given by an LMS. The scores are reported back to the student for self-evaluation, so there is no particular need for high reliability. All of the questions are multiple-response questions (two answers for each) with partial scores being awarded when at only one correct answer is given. For the sake of being definite we assume there are 5 Items and that we have a defined outcome, "SCORE" for each Item. This takes on the value "+1" for each correct response "-1' for each incorrect response. All items are given equal weight. We wish to know, (1) the number of correct items, (2) the total score and (3) the percentage of items actually attempted which were correct;

d) End of chapter test – this is the same as use-case (a) except that all of the items have different weights. We wish to see (1) the number of correct items, (2) the percentage of total weight answered correctly by the candidate and (3) the "weighted" percentage of questions attempted correctly of those questions answered;

e) English comprehension/composition test with essay – this is a placement test designed to assess English language ability. The test consists of two sections: (1) Short answer items (multiple-choice) and (2) the Essay. Short answer items come in two kinds. The first are 15 discrete vocabulary, grammar and usage Items that all produce a single dichotomous outcome "isCorrect". The second are the reading comprehension Items (although they may actually test other skills than just reading) that consist of a stimulus followed by five discrete questions. We assume that these are done as Items with five responses and hence five outcomes: these are Boolean variables whose meanings are as above. All of these Items can be given different weights (as can the 5 outcomes of the two reading comprehension items). The essay question is graded "A,B,C,D,E,F" by either a human or computer rater. The divisions do not necessarily represent equal spacings. The essay represents 1/3 of the total grade. Out of ten points, the mapping of the grades is as follows A=10, B=8, C=7, D=5, E=2, F=0. The total score for the assessment is required;

f) Diagnostic scoring – this is an additional score report which has been retrofitted onto the 'English comprehension test ((e) above). In addition to the overall score, "diagnostic" scores are going to be given in four different sub-areas: "Reading", "Writing", "Vocabulary" and "Grammar" (the dependencies among these sub-scores and the overall score are ignored for computational simplicity). The 15 discrete items and each of the 10 outcomes from the reading items all load onto these sub-scores differently. Some outcomes are completely irrelevant to some of the sub-scores. An Item/outcome can have a high "vocabulary" load and a low "grammar" load, or vice-versa. The essay is also added onto each of the sub-scores, but with a different loading;

g) Diagnostics complex response biology lite – this is a classroom assessment designed to assess the students understanding of the scientific method as it applies to Biology. The assessment returns not a single score but three scores reflecting "Domain Knowledge" (knowledge of the biological subjects in the assessment), "Methodological Knowledge" (understanding of the scientific method) and "Integrated Knowledge" (the ability to apply the scientific method to a given problem). The items are a series of complex "tasks". Each task gives a piece of scenario of a scientific investigation and has the students work through several steps of the process, answering questions about what they are doing at each step. Each task has many complex responses, and hence many observable outcomes. The outcomes are related to the number of steps in the task. Each is a variable given one of the values "Low", "Medium", or "High" depending on how the student did on one part of one step. The outcomes are labelled "OUT***" where '***' is a sequence number unique to the item. For more definite, assume that there are five tasks: Task1 has 7 outcomes, Task2 has 5, Task3 has 8, Task4 has 6 and Task5 has 13. For each of the reporting variables, the scoring model should add an appropriate "weight" if the outcome has reached an appropriate level (either "Medium" and above or "High"). The weight and the level will, in general, be different for each of the reporting variables. Not all outcomes are used for all of the reporting variables;

2.3 ASI Issues

The core features within the ASI structures that are related to the process of selection and ordering are shown in Figure 2.3.

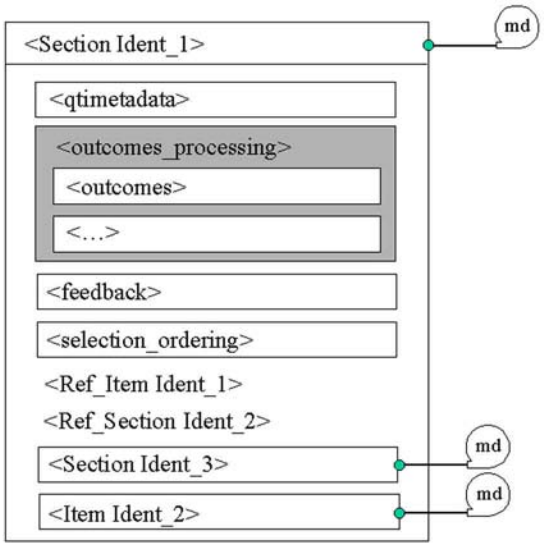


Figure 2.3 An ASI outcomes processing structure.

The relationship of these features to outcomes processing are:

- <qtimetadata> – contains the QTI-specific meta-data about the object. This meta-data may be used to decide which objects are selected to support a particular aggregation mechanism;
- <outcomes_processing> – the set of aggregation processing instructions that are to be applied to the child objects. The variables to contain the aggregated scores are declared using the <outcomes> element;
- <feedback> – the feedback that is to be presented if the conditions defined within the <outcomes_processing> occur;
- <selection_ordering> – the selection and ordering rules that are to be applied to the set of child objects contained within the parent. Only those objects that are presented to the participant can contribute to the scoring algorithm but these may or may not be attempted by the participant;
- Internal and externally referenced Section and Item objects – the set of contained and referenced objects which can be selected and ordered;
- External metadata – the external meta-data descriptions that are linked to the object and which conform to the IMS Meta-data specification.

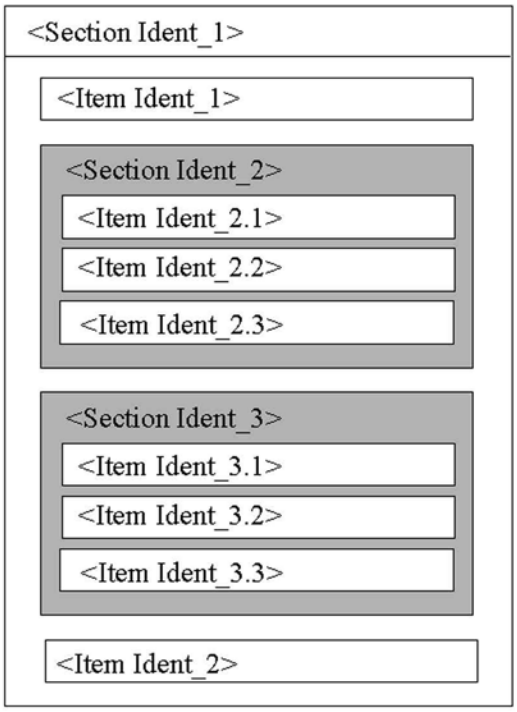


Figure 2.4 An ASI outcomes processing example.

The scoping of the scoring algorithms that are supported is summarized using the example Section/Item combination shown in Figure 2.4. The scope of the scoring algorithm is:

- For Section 'Section Ident_1' the set of children objects is - 'Item Ident_1', 'Section Ident_2', 'Section Ident-3' and 'Item Ident_2'. This means that the scoring algorithm identified in Section 'Section Ident_1' is applied to the default variables associated with each child object;
- For Section 'Section Ident_2' the set of children objects is – 'Item Ident_2.1', 'Item Ident_2.2' and 'Item Ident_2.3'. The Section-level default variables are formed from the aggregation of these child objects;
- For Section 'Section Ident_3' the set of children objects is – 'Item Ident_3.1', 'Item Ident_3.2' and 'Item Ident_3.3'. The Section-level default variables are formed from the aggregation of these child objects;
- If for all of these three Sections only some of the objects are selected then the corresponding aggregation is based upon those objects presented.

It is the responsibility of the scoring engine that is implementing the scoring algorithm to maintain the scoping of the variables. If the default variable names are used then all of the variables will be named 'SCORE'. For instance, the scoping should ensure that:

Section_Ident_2.SCORE = Item_Ident_2.1.SCORE + Item_Ident_2.1.SCORE + Item_Ident_2.1.SCORE

At the level of the 'Section_Ident_1' then the variables 'Item_Ident_2.1.SCORE' etc. are only perceived through the variable 'Section_Ident_2.SCORE'.

2.4 Outcomes Processing Solution

The outcomes processing capability is based upon two complimentary mechanisms:

- In-built scoring algorithms – in which the appropriate scoring algorithm is just named and so the corresponding assessment engine is responsible for applying the algorithm to the default input and output variables;

- Defined proprietary algorithms – in which the relationship between the input and output scoring variables is defined using a parameterized approach.

2.4.1 The In-built Scoring Algorithms

The algorithms that are to be supported are:

a) Number of right answers (including multivariate responses) – this is identified by the algorithm name "NumberCorrect" and has associated with it the default integer variable 'COUNT'. This is the store for the number of correctly answered objects. An object is defined as correctly answered if the value assigned to the default Boolean variable 'COUNT' for the object is 'True'. The number correct algorithm can only operate on Boolean scoring variables. The default integer scoring variables COUNT.max (equal to the number of objects attempted), COUNT.min (set to zero) and COUNT.normalized (a value of between zero and one) are also available. The value of COUNT is determined with respect to the number of objects that have been selected and presented to the participant irrespective if an attempt is made to provide and answer;

b) Number of right answers normalized with respect to those objects that have been attempted and not just selected and presented. This algorithm is named 'NumberCorrectAttempted';

c) Weighted number right (including multivariate responses) – this is similar to the 'number of rights answers' with the addition of the weighting of the individual components and is identified by the algorithm name "WeightedNumberCorrect". The same set of variables are used but their values reflect the object weighting as defined in the associated meta-data field 'qmd_weighting'. The value of COUNT is determined with respect to the number of objects that have been selected and presented to the participant irrespective if an attempt is made to provide and answer;

d) The weighted number of right answers normalized with respect to those objects that have been attempted and not just selected and presented. This algorithm is named 'WeightedNumberCorrectAttempted';

e) Parameterized weighted number correct – name 'ParameterWeightedNumberCorrect' this is similar to the 'WeightedNumberCorrect' algorithm but the weighting value is passed using the <objects_parameter> element in preference to the meta-data field. The value of CORRECT is determined with respect to the number of objects that have been selected and presented to the participant irrespective if an attempt is made to provide and answer;

f) The parameterized weighted number of right answers normalized with respect to those objects that have been attempted and not just selected and presented. This algorithm is named 'ParameterWeightedNumberCorrectAttempted';

g) Percentage correct (including multivariate responses) – this score is derived from both the "NumberCorrect" and "NumberCorrectAttempted" algorithms in which the percentage correct is defined by the equation-

$$\frac{\text{COUNT}*100}{\text{COUNT.max}}$$

h) Sum-of-scores – this is identified by the algorithm name "SumofScores" and has associated with it four default integer variables SCORE, SCORE.min, SCORE.max and SCORE.normalized. These variables store the sum of the scores, the minimum possible score (set to the sum of the 'minvalue' attributes for all of the corresponding objects), the maximum possible score (set to the sum of the 'maxvalue' attributes for all of the corresponding objects) and the normalized score (a value of between zero and one) respectively for the set of objects (the minimum, maximum and normalized values are only valid if they have been explicitly defined for all of the corresponding objects). The value of 'SCORE' is determined with respect to the number of objects that have been selected and presented to the participant irrespective if an attempt is made to provide and answer;

i) Sum of scores normalized with respect to those objects that have been attempted and not just selected and presented. This algorithm is named 'SumofScoresAttempted';

j) Weighted sum of scores (including multivariate responses) – this is similar to the 'sum of scores' with the addition of the weighting of the individual components and is identified by the algorithm name "WeightedSumofScores". The same set of variables are used but their values reflect the object weighting as defined in the associated meta-data field 'qmd_weighting'. The value of SCORE is determined with respect to the number of objects that have been selected and presented to the participant irrespective if an attempt is made to provide and answer;

k) The weighted sum of scores normalized with respect to those objects that have been attempted and not just selected and presented. This algorithm is named 'WeightedSumofScoresAttempted';

l) Parameterized weighted sum of scores – name 'ParameterWeightedSumofScores' this is similar to the 'WeightedSumofScores' algorithm but the weighting value is passed using the <objects_parameter> element in preference to the meta-data field. The value of SCORE is determined with respect to the number of objects that have been selected and presented to the participant irrespective if an attempt is made to provide and answer;

m) The parameterized weighted sum of scores normalized with respect to those objects that have been attempted and not just selected and presented. This algorithm is named 'ParameterWeightedSumofScoresAttempted';

n) Best 'K' out of 'N' (including multivariate responses) – this is identified by the algorithm name "BestKofN" and is used to calculate the outcome by using the highest 'K' scores from the set of presented objects generically defined as 'N' objects. The values are placed in the default variables 'SCORE', 'SCORE.min', 'SCORE.max' and 'SCORE.normalized'. The value of SCORE is determined with respect to the number of objects that have been selected and presented to the participant irrespective if an attempt is made to provide and answer;

o) Negative scores (including multivariate responses) – called 'GuessingPenalty'. Each object has three associated variables for the values of the number of correct answers ('COUNT.correct'), the number of incorrect answers ('COUNT.incorrect') and the number of unattempted answers ('COUNT.unattempted');

p) Item Response Theory (IRT) – supported using the processing parameter mechanism;

q) Time remaining – **for further study in V2.0**;

r) Par scores – **for further study in V2.0**;

s) Cut scores (pass/fail, right/wrong and including multivariate responses) – **for further study in V2.0**;

t) Multiple forms/equating (table based) – **for further study in V2.0**;

u) Partial credit/Graded response – **for further study**;

v) Critical Item – **for further study in V2.0**;

w) Critical Outcome – **for further study in V2.0**;

x) Factor analysis/ Multivariate IRT (research) –supported through extensions. **For further study in V2.0**;

y) Bayes net (research) – supported through extensions. **For further study in V2.0**;

z) Computer Adaptive Testing – **for further study in V2.0**.

2.4.2 Defined Proprietary Algorithms

Proprietary algorithms can be used but these must be characterized using the parameter definition mechanisms supported in the specification i.e. the <objects_parameter> and <processing_parameter> elements. Any number of parameters can be passed using these elements and there is no restriction on the naming convention used for these parameters – all parameters are specific to an algorithm. The names of the proprietary algorithms (passed in the *scoremodel* attribute) should not clash with the intrinsic names and some form of naming convention based upon the creating organization should be adopted.

2.5 Tabular Representation

The tables in this Section provide a conceptual, informative description of the elements in the data objects. The columns in these tables refer to:

No: The number of the data element. An element may be composed of sub-elements. The numbering scheme reflects these relationships.

Name: The descriptive name of the element.

Explanation: A brief functional description of the element.

Required: Indicates if the element is required:

- **M** = Mandatory Element that must be included in the data object, if the element at the higher level is included;
- **C** = Conditional Element. Existence is dependent on values of other Elements;
- **O** = Optional Element.

Multi: Multiplicity of the element:

- Blank = single instance;
- Number = maximum number of times the element is repeatable;
- **n** = multiple occurrences allowed, no limit;
- Repeatability of an element implies that all sub-elements repeat with the same element.

Type: A description of formatting rules for the data element. Type includes the maximum length of the element:

- **ID** = element used to uniquely identify an object;
- **Code** = element value from a list of codes;
- **Description** = descriptive element, human language
- **Flag** = binary flag
- **Enumerated** = list of predefined non-numeric options i.e. the definitive list of objects
- The international character set specified by ISO 10646 will be used for all fields.

The type will also include a description of the set of valid values for the sub-element:

- Coding schemes using numerical values;
- The set of values as defined in the Domain i.e. making it closed. The list of values cannot be extended to include values not defined in the specification. If there is a need for values not included in the domain set of values then the extension should be done defining a new element under the Extension element that is a part of each data object definition.

Note: Additional descriptive information about the element.

2.5.1 QTI Outcomes Processing Data Objects

Table 2.1 describes the data objects that are used in the construction of the QTI outcomes processing elements.

Table 2.1 QTI outcomes processing data objects detailed description.

No	Name	Explanation	Reqd	Mult	Type	Note
1	outcomes_processing	Accumulated outcomes processing and feedback applied within Sections and Assessments.	O	n		
1.1	scoremodel	The type of scoring model being adopted.	O		CDATA string describing the model. String 32 chars.	Default string is “SumofScores”.
1.2	qticomment	The comments used to annotate the XML file.	O			Comments should be used to aid human readability of the XML file itself.
1.2.1	xml:lang	The language that is being used for the information.	O		String 32 chars.	The language entries will be defined as per the ISO639 and ISO3166 standards.
1.3	outcomes	To create the variables required for the assessment accumulated scores.	M			The assessment accumulated processing variables group.
1.3.1	qticomment	As per structure 1.2.				
1.3.2	decvar	Declaration of a variable to be used for scoring.	M	n		Each type of variable must be declared before it is used.
1.3.2.1	varname	The name of the variable to be declared.	M		String 256 chars.	Default is set as ‘SCORE”.
1.3.2.2	vartype	The type of variable.	M		Enumerated: String Integer (default) Decimal Scientific Boolean Enumerated Set	Default is set to ‘Integer’.
1.3.2.3	defaultval	The default value for the variable.	O		Numerical 32 chars String 32 chars True/False	Can be set to any value. Default is set to ‘0’.
1.3.2.4	minvalue	The minimum value permitted for a numeric score.	O		String 32chars.	Applies to the value of the score after all of the item processing has been completed.
1.3.2.5	maxvalue	The maximum score permitted for a numeric score.	O		String 32chars.	Applies to the value of the score after all of the item processing has been completed.
1.3.2.6	cutvalue	The value above which the participant will have been defined to have mastery of the subject.	O		String 32chars.	The type of this cut value is set by the variable type.
1.3.2.7	members	The set of enumerated values.	O		String 1024chars.	This is a comma separated list without enclosing parentheses.

1.3.3	interpretvar	The interpretation to be applied to the variable in terms relevant to an actor.	O	n		At present this element will be a comment string however it will be further developed in version 1.2.
1.3.3.1	varname	The name of the input variable being described.	O		As per structure 1.3.2.1.	
1.3.3.2	view	The view to which the interpretation is applied.	O	n	Enumerated: All (default) Administrator AdminAuthority Assessor Author Candidate Invigilator Proctor Psychometrician Scorer Tutor	The 'All' view is the default value.
1.4	objects_condition	This contains the conditions that are applied to define the ways in which the outcomes variables are combined to create the aggregated value.	O	n		
1.4.1	qticomment	As per structure 1.2.				
1.4.2	outcomes_metadata	Contains the rules that are applied to the IMS QTI-specific and/or IMS Meta-data meta-data fields of the object to decide if the object scoring is to be aggregated.	C		String 1-256 chars.	
1.4.2.1	mdname	Identifies the IMS QTI-specific or IMS Meta-data field that is to be used for the aggregation rule.	M		String 1-64 chars.	
1.4.2.2	mdoperator	Identifies the nature of the meta-data field comparison that is to be applied.	M		Enumerated list: EQ NEQ LT LTE GT GTE	
1.4.3	and_objects	Contains the construction of complex score condition rules to be built based upon the logical 'AND' operator. The object is selected for aggregation if all of the contained rules are 'True'.	C			See Appendix B for the logic rules.
1.4.3.1	outcomes_metadata	As per structure 1.4.2				
1.4.3.2	and_objects	As per structure 1.4.3				
1.4.3.3	or_objects	As per structure 1.4.4				
1.4.3.4	not_objects	As per structure 1.4.5				

1.4.4	or_objects	Contains the construction of complex score condition rules to be built based upon the logical 'OR' operator. The object is selected for aggregation if at least one of the contained rules is 'True'.	C			See Appendix B for the logic rules.
1.4.4.1	outcomes_metadata	As per structure 1.4.2				
1.4.4.2	and_objects	As per structure 1.4.3				
1.4.4.3	or_objects	As per structure 1.4.4				
1.4.4.4	not_objects	As per structure 1.4.5				
1.4.5	not_objects	Contains the construction of complex rules to be built based upon the logical 'NOT' operator. The object is selected for aggregation if the contained rule(s) is 'False'.	C			This element contains only ONE of the sub-elements. See Appendix B for the logic rules.
1.4.5.1	outcomes_metadata	As per structure 1.4.2				
1.4.5.2	and_objects	As per structure 1.4.3				
1.4.5.3	or_objects	As per structure 1.4.4				
1.4.5.4	not_objects	As per structure 1.4.5				
1.4.6	objects_parameter	Contains the value of a particular parameter that is to be used by the corresponding scoring algorithm variable selection. Each parameter has a particular meaning to each scoring algorithm i.e. there is no established vocabulary for these parameters.	O	n		These parameters only apply to the objects within the containing <objects_condition> element.
1.4.6.1	pname	The name of the parameter that is being defined.	M		String 256 chars.	Parameters of the same name may have different significance to different algorithms.
1.4.7	map_input	This element is used to re-map the default input variable to another variable.	O	n		
1.4.7.1	varname	The name of the input variable whose name is to be remapped.	O		As per structure 1.3.2.1.	
1.4.8	objectscond_extension	The objects condition extension facility.	O	n	ANY	
1.5	processing_parameter	The value of a parameter that is passed to the scoring algorithm that is being used.	O	n	String 128 chars.	These parameters apply to the algorithm as a whole and not to specific objects operated on by the algorithm.
1.5.1	pname	The name of the parameter that is being defined.	M		String 256 chars.	Parameters of the same name may have different significance to different algorithms.

1.6	map_output	This is used to remap the result of the scoring algorithm from the default/named variable to another named variable.	O	n	String 256 chars.	The new variable name should have been declared in the <outcomes> element of the parent object. This remapping operates for all of the associated variable names e.g. the *.min , *.max and *.normalized variables.
1.6.1	varname	The name of the output variable whose name is to be remapped.	O		As per structure 1.3.2.1.	
1.7	outcomes_feedback_test	Contains the tests to be applied to determine if any and the type of feedback to be presented.	O	n		
1.7.1	title	The title of the feedback test.	O		String 256 chars.	
1.7.2	test_variable	The conditional test that is to be applied to the aggregated score variables. A wide range of separate and combinatorial tests can be applied.	M			
1.7.2.1	variable_test	The conditional test that is to be applied to the aggregated score variables. A wide range of separate and combinatorial tests can be applied.	C			
1.7.2.1.1	varname	The name of the variable whose state is to be tested.	O		As per structure 1.3.2.1.	
1.7.2.1.2	testoperator	Identifies the nature of the variable comparison that is to be applied.	M		Enumerated list: EQ NEQ LT LTE GT GTE	
1.7.2.2	and_test	This element allows the construction of complex variable test rules to be built based upon the logical ‘AND’ operator.	C			
1.7.2.2.1	variable_test	As per structure 1.7.2.1.				
1.7.2.2.2	and_test	As per structure 1.7.2.2.				
1.7.2.2.3	or_test	As per structure 1.7.2.3.				
1.7.2.2.4	not_test	As per structure 1.7.2.4.				
1.7.2.3	or_test	This element allows the construction of complex variable test rules to be built based upon the logical ‘OR’ operator.	C			
1.7.2.3.1	variable_test	As per structure 1.7.2.1.				
1.7.2.3.2	and_test	As per structure 1.7.2.2.				
1.7.2.3.3	or_test	As per structure 1.7.2.3.				
1.7.2.3.4	not_test	As per structure 1.7.2.4.				

1.7.2.4	not_test	This element allows the construction of complex variable test rules to be built based upon the logical 'NOT' operator.	C			Only one of the contained sub-elements can be used within each usage of the NOT operator.
1.7.2.4.1	variable_test	As per structure 1.7.2.1.				
1.7.2.4.2	and_test	As per structure 1.7.2.2.				
1.7.2.4.3	or_test	As per structure 1.7.2.3.				
1.7.2.4.4	not_test	As per structure 1.7.2.4.				
1.7.3	displayfeedback	The trigger for displaying feedback.	M	n		
1.7.3.1	feedbacktype	The type of feedback to be displayed.	M		Enumerated: Response (default) Solution Hint	The default value is 'Response'.
1.7.3.2	linkrefid	The identifier of the material to be referenced.	M		String 32 chars.	Consistency checking is beyond the scope of this specification. Usage rules are given in the Q&TI Best Practice Guide.

3. XML Binding

3.1 outcomes_processing> Elements

Description: The <outcomes_processing> element is the container for all of the outcomes processing instructions for Assessments and Sections. Multiple outcomes processing containers can be use when multiple scoring algorithms are to be applied to produce the aggregated outcomes. If multiple outcomes_processing elements are supplied, it is the intention that all should be run and the outcomes from all of them together should be reported as the outcomes of the enclosing Section or Assessment. The outcome variables defined by each <outcomes_processing> element should be unique across all outcomes_processing elements defined by a section. In particular, it is an error for multiple <outcomes_processing> elements to set the same outcome variable and the results will be undefined.

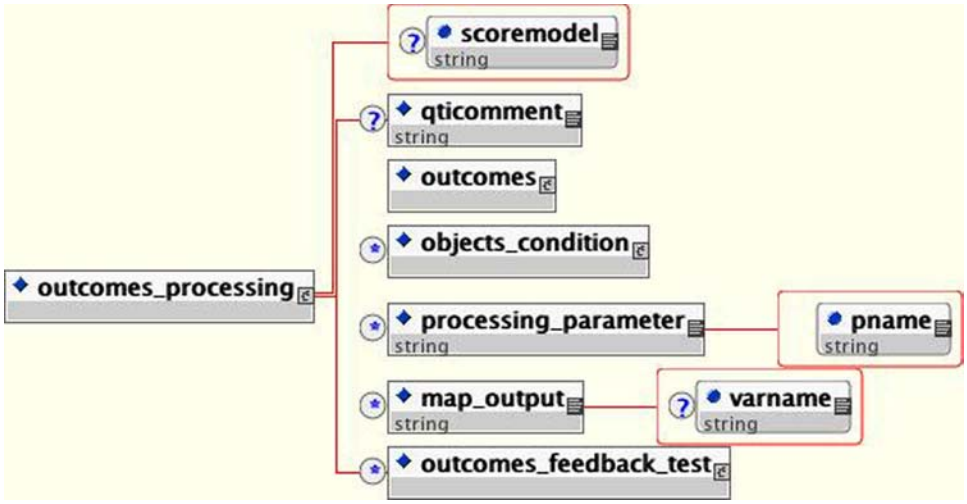


Figure 3.1 The <outcomes_processing> element structure.

Multiplicity: The <outcomes_processing> occurs zero or more times within the <assessment> and <section> elements.

Attributes:

- **scoremodel (optional. Default='SumofScores').** The type of the scoring algorithm that is to be used. Data-type = String (max of 32 chars).

Elements:

- qticomment
- outcomes
- objects_condition
- processing_parameter
- map_outcome
- outcomes_feedback_test

Example:

```
<outcomes_processing scoremodel="SumofScores">
   <outcomes>
      <decvar/>
   </outcomes>
```

```
</outcomes_processing>
```

3.1.1 <qticomment> Element

Description: This element contains the comments that are relevant to the outcomes processing structure as a whole.

Multiplicity: Occurs zero or once within the <outcomes_processing> element.

Attributes:

- **xml:lang (optional).** Identifies the language that is to be used within the instance. The vocabulary is defined as per the XML W3C specification.
  Data-type = string.

3.1.2 <outcomes> Element

Description: The <outcomes> element contains all of the variable declarations that are to be made available to the scoring algorithm. Each variable is declared using the <decvar> element apart from the default variable called 'SCORE' that is an integer and has a default value of zero (0).

Multiplicity: This occurs once within the <outcomes_processing> element.

Attributes: See sub-section 3.2.

3.1.3 <objects_condition> Element

Description: This element contains the conditions that are applied to define the ways in which the outcomes variables are combined to create the aggregated value using the in-built set of algorithm definitions.

Multiplicity: Occurs zero or more times within the <outcomes_processing> element.

Attributes: See sub-section 3.3.

3.1.4 <processing_parameter> Element

Description: This element contains the value of a particular parameter that is to be used by the corresponding scoring algorithm. Each parameter has a particular meaning to each scoring algorithm i.e. there is no established vocabulary for these parameters.

Multiplicity: Occurs zero or more times within the <outcomes_processing> element.

Attributes:

- **pname (mandatory).** The name of the proprietary parameterized value that is to be used by the scoring algorithm.
  Data-type = string (1-64 chars).

3.1.5 <map_output> Element

Description: This element is used to re-map the named variable to another named variable (given in the body of the element). The target variable name must have been declared using <decvar> in the <outcomes> element of the enclosing <outcomes_processing> element. When a variable is remapped, all of its derived variables are to be remapped as well. Thus if remapping 'SCORE' to 'myScore', 'SCORE.min', 'SCORE.max' and 'SCORE.normalized' would be remapped to 'myScore.min', 'myScore.max' and 'myScore.normalized' respectively. Data-type = string (1-256 chars).

Multiplicity: This occurs zero or many times within the <outcomes_condition> element.

Attributes:

- **varname (optional. Default = 'SCORE').** The name of the variable that is normally used by the scoring algorithm e.g. 'SCORE', 'CORRECT', etc. The default name is 'SCORE' but the variables SCORE.min, SCORE.max and SCORE.normalized are also to be remapped.
  Data-type = String (max of 256 chars).

3.1.6 <outcomes_feedback_test> Element

Description: The <outcomes_feedback> element contains the tests to be applied to determine if any and the type of feedback to be presented. This feedback could include information about passing the assessment etc.

Multiplicity: This occurs zero or many times within the <outcomes_processing> element.

Attributes: See sub-section 3.4.

3.2 <outcomes> Element

Description: The <outcomes> element contains all of the variable declarations that are to be made available to the scoring algorithm. Each variable is declared using the <decvar> element apart from the default variable called 'SCORE' that is an integer and has a default value of zero (0). The declaration of each variable is accompanied by the implicit declaration of the <vname>.min, <vname>.max and <vname>.normalized variables. These are used to store the minimum value, maximum value and normalized value (in the range zero to one) of the scoring variable.

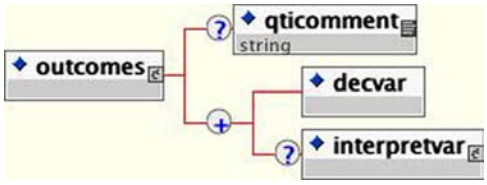


Figure 3.2 <outcomes> elements.

Multiplicity: This occurs once within the <outcomes_processing> element.

Attributes: None.

Elements:

- qticomment
- decvar
- interpretvar

3.2.1 <qticomment> Element

Description: This element contains the comments that are relevant to the outcomes.

Multiplicity: Occurs zero or once within the <outcomes> element.

Attributes:

- **xml:lang (optional).** Identifies the language that is to be used within the instance. The vocabulary is defined as per the XML W3C specification.
  Data-type = string (1-2048 chars).

3.2.2 <decvar> Element

Description: The <decvar> element permits the declaration of the scoring variables. The default name is 'SCORE' but the variables SCORE.min, SCORE.max and SCORE.normalized are also declared and made available from the scoring algorithms. A <decvar> with no attributes is assumed to define the integer variable 'SCORE' with a default value of zero.

Multiplicity: This occurs one or more times within the <outcomes> element.

Attributes:

- **varname (optional. Default = 'SCORE').** The name of the variable that is to be declared.
  Data-type = String (max of 256 chars).
- **vartype (required. Enumerated list: String, Decimal, Scientific, Boolean, Integer, Enumerated, Set. Default=Integer).** The type of the variable declared.
  Data-type = Enumerated list.
- **defaultval (optional).** The default value to which the variable is to be initialized.
  Data-type = String (max of 16 chars).
- **cutvalue (optional).** The value/grade above which the subject is considered to have been mastered.
  Data-type = String (max of 16 chars).
- **minvalue (optional).** The minimum value permitted for a numeric score.
  Data-type = String (max of 32 chars).
- **maxvalue (optional).** The maximum value permitted for a numeric score.
  Data-type = String (max of 32 chars).
- **members (optional).** The set of enumerated values that constitute the member.
  Data-type = String (max of 1024 chars).

3.2.3 <interpretvar> Element

Description: The <interpretvar> element is used to provide statistical interpretation information about the associated variables.

Multiplicity: This occurs zero or more times within the <outcomes> element.

Attributes:

- **varname (optional. Default = 'SCORE').** The name of the variable whose interpretation details are to be described. The default name is 'SCORE'.
  Data-type = String (max of 256 chars).
- **view (optional with selection from the enumerated list of: All, Administrator, AdminAuthority, Assessor, Author, Candidate, InvigilatorProctor, Psychometrician, Scorer, Tutor. Default=All).** The view defines the scope for the display of the associated information i.e. to whom the material can be presented.
  Data-type = Enumerated list.

Another description of this element and its sub-components is given in IMS QTI: ASI XML Binding V1.2 [QTI, 02b].

3.3 <objects_condition>

Description: Each <objects_condition> element defines a subset of the objects (Items and Sections) selected by the selection algorithm that are to be used in this score. This allows for the construction of subscores. If no <outcomes_metadata> element is present within the <objects_conditions> element to select a subset of objects, then the <objects_conditions> applies to all objects selected by the selection algorithm. If multiple <objects_conditions> are given within a single <outcomes_processing> element, then the algorithm is applied to the union of all objects selected. The <objects_parameter>, <map_input> and <objectscond_extension> elements within the

<objects_condition> element apply only to those objects selected in the condition. If an object is selected by more than one <objects_condition> element, it should receive parameter assignments and input mapping as if it were the first element so selected. These conditions include the identification and mapping of the variables for input.

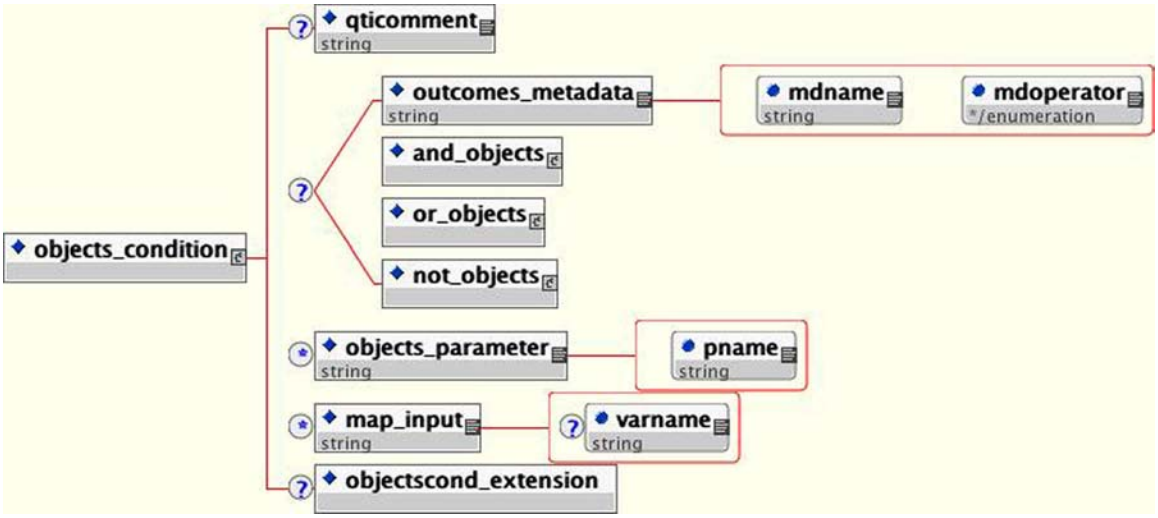


Figure 3.3 <objects_condition> elements.

Multiplicity: This occurs zero or many times within the <outcomes_processing> element.

Attributes: None.

Elements:

- qticomment
- outcomes_metadata
- and_objects
- or_objects
- not_objects
- objects_parameter
- map_input
- objectscond_extension

3.3.1 <qticomment> Element

Description: This element contains the comments that are relevant to the presentation of the Item.

Multiplicity: Occurs zero or once within the <objects_condition> element.

Attributes:

- **xml:lang (optional).** Identifies the language that is to be used within the instance. The vocabulary is defined as per the XML W3C specification.
  Data-type = string (1-2048 chars).

3.3.2 <outcomes_metadata> Element

Description: This element defines the rule that is applied to the QTI-specific and/or IMS Meta-data meta-data fields of the object to decide if the object scoring is to be aggregated. The content contains the value of the meta-data field that is being tested for within the rule. This data is a string of up to 64 characters length.

Multiplicity: Occurs zero or once within the <objects_condition> element.

Attributes: See Sub-section 3.5.

Elements: None.

3.3.3 <and_out> Element

Description: This element allows the construction of complex score condition rules to be built based upon the logical 'AND' operator. The object is selected for aggregation if all of the contained rules are 'True'.

Multiplicity: This occurs zero or once within the <objects_condition> element.

Attributes: See Sub-section 3.6.

3.3.4 <or_out> Element

Description: This element allows the construction of complex score condition rules to be built based upon the logical 'OR' operator. The object is selected for aggregation if at least one of the contained rules is 'True'.

Multiplicity: This occurs zero or once within the <objects_condition> element.

Attributes: See Sub-section 3.7.

3.3.5 <not_out> Element

Description: This element allows the construction of complex rules to be built based upon the logical 'NOT' operator. The object is selected for aggregation if the contained rule(s) is 'False'.

Multiplicity: This occurs zero or once within the <objects_condition> element.

Attributes: See Sub-section 3.8.

3.3.6 <objects_parameter> Element

Description: This element contains the value of a particular parameter that is to be used by the corresponding scoring algorithm variable selection. Each parameter has a particular meaning to each scoring algorithm i.e. there is no established vocabulary for these parameters. These parameters are applied only to the objects selected by the enclosing <outcomes_condition> element. If multiple <outomes_condition> select the same object, the parameters are taken from the first <outcomes_condition> that selects the element.

Multiplicity: Occurs zero or more times within the <objects_condition> element.

Attributes:

- **pname (mandatory).** The name of the proprietary parameterized value that is to be used by the scoring algorithm.
  Data-type = string (1-64 chars).

3.3.7 <map_input> Element

Description: This element is used to re-map the named input variable to another variable. The default variable names are derived from the type of scoring algorithm identified in <outcomes_processing> element. The target variable name must have been declared in the evaluation objects that undergoing aggregation and must be of the same type as the default variable. When a variable is remapped, all of its derived variables are to be remapped as well. The input mapping is applied only to the objects selected by the enclosing <outcomes_condition> element. If multiple <outcomes_condition> select the same object, the input mapping used is the one contained in the first <outcomes_condition> that selects the element.
Data-type = String (1-256 chars).

Multiplicity: This occurs zero or many times within the <objects_condition> element.

Attributes:

- **varname (optional. Default = 'SCORE').** The name of the variable that is normally used in the scoring algorithm e.g. 'SCORE', 'COUNT', etc.
  Data-type = String (max of 256 chars).

3.3.8 <objectscond_extension> Element

Description: This element contains the proprietary extensions that can be used to extend the functionally capabilities of the <outcomes_condition> element.

Multiplicity: Occurs zero or once within the <objects_condition> element.

Attributes: None.

3.4 <outcomes_feedback_test> Element

Description: The <outcomes_feedback> element contains the tests to be applied to determine if any and the type of feedback to be presented. This feedback could include information about passing the assessment etc.

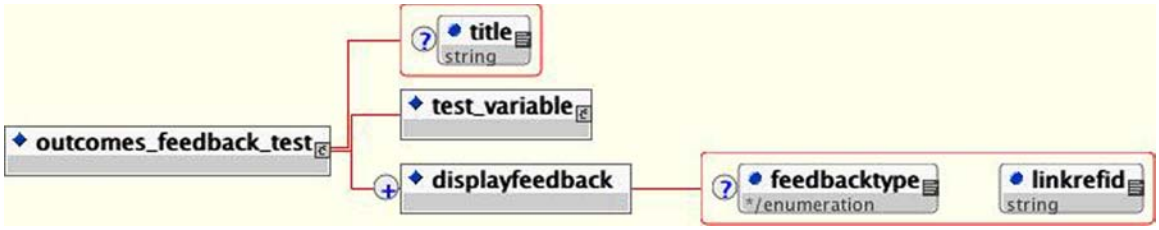


Figure 3.4 <outcomes_feedback_test> elements.

Multiplicity: This occurs zero or many times within the <outcomes_processing> element.

Attributes:

- **title (optional).** The title of the feedback test.
  Data-type = String (max of 256 chars).

Elements:

- test_variable
- displayfeedback

3.4.1 <test_variable> Elements

Description: The conditional test that is to be applied to the aggregated score variables. A wide range of separate and combinatorial tests can be applied.

Multiplicity: This occurs once within the <outcomes_feedback_test> element.

Attributes: See Sub-section 3.9.

3.4.2 <displayfeedback> Element

Description: The <displayfeedback> element is responsible for assigning an associated feedback to the aggregated scoring if the 'True' state results.

Multiplicity: This occurs zero or more times within the <outcomes_feedback_test> element.

Attributes:

- **feedbacktype (optional – enumerated list: Response, Solution, Hint. Default = Response).** The type of feedback that has been triggered by the associated outcomes condition. Within the Assessment and Section feedback elements only the 'Response' type of feedback is supported.
  Data-type = Enumerated list.
- **linkrefid (required).** The identifier of the associated feedback. An <assessfeedback> or <sectionfeedback> element must exist with this identifier as defined by the 'ident' attribute.
  Data-type = String (max of 256 chars).

3.5 <outcomes_metadata> Element

Description: This element defines the rule that is applied to the IMS QTI-specific and/or IMS Meta-data fields of the object to decide if the object scoring is to be aggregated. The content contains the value of the meta-data field that is being tested for within the rule.
Data-type = String (1-64 chars).

Multiplicity: Occurs zero or once within the <objects_condition> element.

Attributes:

- **mdname (mandatory).** Identifies the QTI-specific or IMS Meta-data field that is to be used for the aggregation rule. No validation check is made in the instance on the existence or otherwise of this field.
  Data-type = string (1-256 chars).
- **mdoperator (mandatory with selection from the enumerated list of: EQ, NEQ, LT, LTE, GT, GTE).** Identifies the nature of the meta-data field comparison that is to be applied.
  Data-type = string (1-16 chars).

Elements: None.

Example:

```
<objects_condition>
   <outcomes_metadata mdname="qmd_topic" mdoperator="EQ">Trigonometry</outcomes_metadata>
</objects_condition>
```

3.6 <and_objects> Element

Description: The <and_objects> element supports the construction of complex score condition rules to be built based upon the logical 'AND' operator. The object is selected for aggregation if all of the contained rules are 'True'. The logic rules for this element are described in Appendix B.

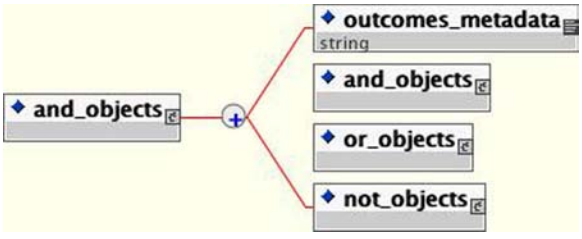


Figure 3.5 <and_objects> elements.

Multiplicity: This occurs zero or once within the <objects_condition>, and zero or many times within the <and_objects>, <or_objects> and <not_objects> elements.

Attributes: None.

Elements:

- outcomes_metadata
- and_objects
- or_objects
- not_objects

3.6.1 <outcomes_metadata> Element

Description: This element defines the rule that is applied to the IMS QTI-specific and/or IMS Meta-data fields of the object to decide if the object scoring is to be aggregated. The content contains the value of the meta-data field that is being tested for within the rule.
Data-type = string (1-64 chars).

Multiplicity: Occurs zero or many times within the <and_objects> element.

Attributes:

- **mdname (mandatory).** Identifies the IMS QTI-specific or IMS Meta-data field that is to be used for the aggregation rule. No validation check is made in the instance on the existence or otherwise of this field. Data-type = string (1-64 chars).
- **mdoperator (mandatory with selection from the enumerated list of: EQ, NEQ, LT, LTE, GT, GTE).** Identifies the nature of the meta-data field comparison that is to be applied.
  Data-type = string (1-16 chars).

Elements: None.

3.6.2 <and_objects> Element

Description: This element allows the construction of complex score condition rules to be built based upon the logical 'AND' operator. The object is selected for aggregation if all of the contained rules are 'True'.

Multiplicity: This occurs zero or many times within the <and_objects> element.

Attributes: See Sub-section 3.6.

3.6.3 <or_objects> Element

Description: This element allows the construction of complex score condition rules to be built based upon the logical 'OR' operator. The object is selected for aggregation if at least one of the contained rules is 'True'.

Multiplicity: This occurs zero or many times within the <and_objects> element.

Attributes: See Sub-section 3.7.

3.6.4 <not_objects> Element

Description: This element allows the construction of complex rules to be built based upon the logical 'NOT' operator. The object is selected for aggregation if the contained rule(s) is 'False'.

Multiplicity: This occurs zero or many times within the <and_objects> element.

Attributes: See Sub-section 3.8.

3.7 <or_objects> Element

Description: The <or_objects> element supports the construction of complex score condition rules to be built based upon the logical 'OR' operator. The object is selected for aggregation if at least one of the contained rules is 'True'. The logic rules are described in Appendix B.

Multiplicity: This occurs zero or once within the <objects_condition>, and zero or more times within the <and_objects>, <or_objects> and <not_objects> element.

Attributes: None.

Elements:

- outcomes_metadata
- and_objects
- or_objects
- not_objects

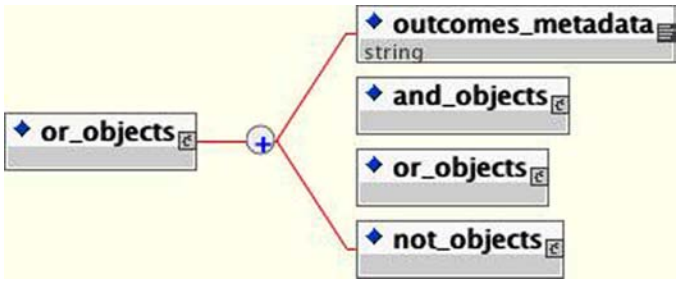


Figure 3.6 <or_objects> elements.

3.7.1 <outcomes_metadata> Element

Description: This element defines the rule that is applied to the IMS QTI-specific and/or IMS Meta-data fields of the object to decide if the object scoring is to be aggregated. The content contains the value of the meta-data field that is being tested for within the rule.
Data-type = string (1-64 chars).

Multiplicity: Occurs zero or many times within the <or_objects> element.

Attributes:

- **mdname (mandatory).** Identifies the QTI-specific or IMS Meta-data field that is to be used for the aggregation rule. No validation check is made in the instance on the existence or otherwise of this field.
  Data-type = string (1-64 chars).
- **mdoperator (mandatory with selection from the enumerated list of: EQ, NEQ, LT, LTE, GT, GTE).** Identifies the nature of the meta-data field comparison that is to be applied.
  Data-type = string (1-16 chars).

Elements: None.

3.7.2 <and_objects> Element

Description: This element allows the construction of complex score condition rules to be built based upon the logical 'AND' operator. The object is selected for aggregation if all of the contained rules are 'True'.

Multiplicity: This occurs zero or many times within the <or_objects> element.

Attributes: See Sub-section 3.6.

3.7.3 <or_objects> Element

Description: This element allows the construction of complex score condition rules to be built based upon the logical 'OR' operator. The object is selected for aggregation if at least one of the contained rules is 'True'.

Multiplicity: This occurs zero or many times within the <or_objects> element.

Attributes: See Sub-section 3.7.

3.7.4 <not_objects> Element

Description: This element allows the construction of complex rules to be built based upon the logical 'NOT' operator. The object is selected for aggregation if the contained rule(s) is 'False'.

Multiplicity: This occurs zero or many times within the <or_objects> element.

Attributes: See Sub-section 3.8.

3.8 <not_objects> Element

Description: The <not_objects> element supports the construction of complex rules to be built based upon the logical 'NOT' operator. The object is selected for aggregation if the contained rule(s) is 'False'. The logic rules are listed in Appendix B.

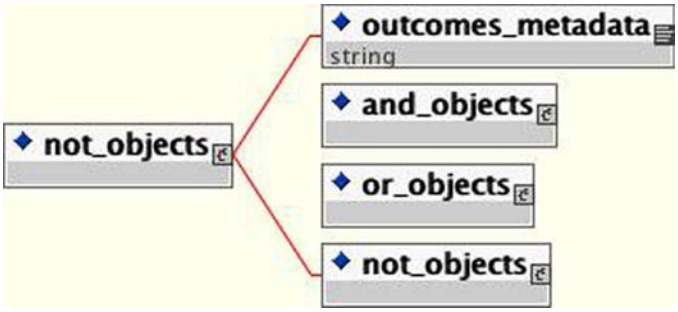


Figure 3.7 <not_objects> elements.

Multiplicity: This occurs zero or once within the <objects_condition>, and zero or many times within the <and_objects>, <or_objects> and <not_objects> element.

Attributes: None.

Elements:

- outcomes_metadata
- and_objects
- or_objects
- not_objects

3.8.1 <outcomes_metadata> Element

Description: This element defines the rule that is applied to the QTI-specific and/or IMS Meta-data fields of the object to decide if the object scoring is to be aggregated. The content contains the value of the meta-data field that is being tested for within the rule.
Data-type = string (1-64 chars).

Multiplicity: Occurs zero or once within the <not_objects> element.

Attributes:

- **mdname (mandatory).** Identifies the IMS QTI-specific or IMS Meta-data field that is to be used for the aggregation rule. No validation check is made in the instance on the existence or otherwise of this field. Data-type = string (1-64 chars).
- **mdoperator (mandatory with selection from the enumerated list of: EQ, NEQ, LT, LTE, GT, GTE).** Identifies the nature of the meta-data field comparison that is to be applied. Data-type = string (1-16 chars).

Elements: None.

3.8.2 <and_objects> Element

Description: This element allows the construction of complex score condition rules to be built based upon the logical 'AND' operator. The object is selected for aggregation if all of the contained rules are 'True'.

Multiplicity: This occurs zero or once within the <not_objects> element.

Attributes: See Sub-section 3.6.

3.8.3 <or_objects> Element

Description: This element allows the construction of complex score condition rules to be built based upon the logical 'OR' operator. The object is selected for aggregation if at least one of the contained rules is 'True'.

Multiplicity: This occurs zero or many times within the <not_objects> element.

Attributes: See Sub-section 3.7.

3.8.4 <not_objects> Element

Description: This element allows the construction of complex rules to be built based upon the logical 'NOT' operator. The object is selected for aggregation if the contained rule(s) is 'False'.

Multiplicity: This occurs zero or many times within the <not_objects> element.

Attributes: See Sub-section 3.8.

3.9 <test_variable> Element

Description: The <test_variable> element contains the conditions that are applied to a defined set of input scoring variables to determine if feedback is to be presented. Complex test structures can be constructed using the associated logic elements.

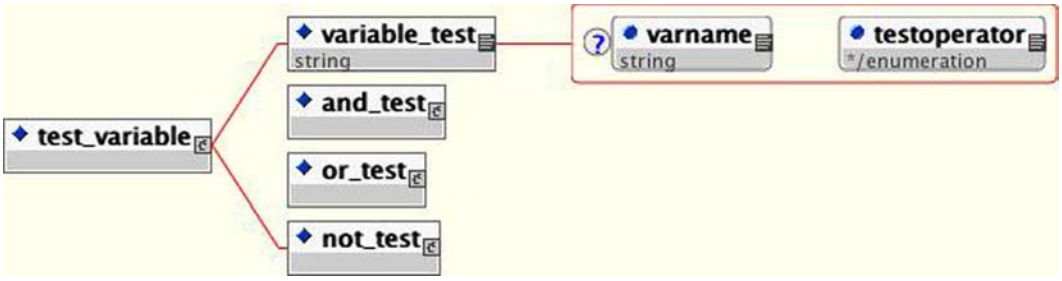


Figure 3.8 <test_variable> elements.

Multiplicity: This occurs once within the <outcomes_feedback_test> element.

Attributes: None.

Elements:

- variable_test
- and_test
- or_test
- not_test

3.9.1 <variable_test> Element

Description: The conditional test that is to be applied to the aggregated score variables. A wide range of separate and combinatorial tests can be applied.

Multiplicity: This occurs zero or once within the <test_variable> element.

Attributes:

- **varname (optional. Default = 'SCORE').** The name of the variable whose condition is to be tested. The default name is 'SCORE'.
  Data-type = String (max of 256 chars).
- **testoperator (mandatory with selection from the enumerated list of: EQ, NEQ, LT, LTE, GT, GTE).** Identifies the nature of the variable comparison that is to be applied.
  Data-type = string (1-16 chars).

3.9.2 <and_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'AND' operator.

Multiplicity: This occurs zero or once within the <test_variable> element.

Attributes: See Sub-section 3.10.

3.9.3 <or_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'OR' operator.

Multiplicity: This occurs zero or many times within the <test_variable> element.

Attributes: See Sub-section 3.11.

3.9.4 <not_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'NOT' operator.

Multiplicity: This occurs zero or once within the <test_variable> element.

Attributes: See Sub-section 3.12.

3.10 <and_test> Element

Description: The <and_test> element is used to define variable test conditions all of which must be defined as 'True' to result in the output variables being assigned the corresponding values.

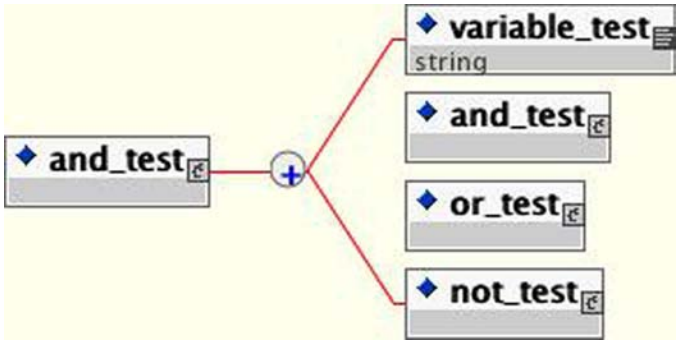


Figure 3.9 <and_test> elements.

Multiplicity: This occurs zero or once within the <test_variable> element, and zero or more times in the <and_test>, <or_test> and <not_test> elements.

Attributes: None.

Elements:

- variable_test
- and_test
- or_test
- not_test

3.10.1 <variable_test> Element

Description: The conditional test that is to be applied to the aggregated score variables. A wide range of separate and combinatorial tests can be applied.

Multiplicity: This occurs zero or more times within the <and_test> element.

Attributes:

- **varname (optional. Default = 'SCORE').** The name of the variable that is to be tested. The default name is 'SCORE'.
  Data-type = String (max of 256 chars).
- **testoperator (mandatory with selection from the enumerated list of: EQ, NEQ, LT, LTE, GT, GTE).**
  Identifies the nature of the variable comparison that is to be applied.
  Data-type = string (1-16 chars).

3.10.2 <and_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'AND' operator.

Multiplicity: This occurs zero or many times within the <and_test> element.

Attributes: See Sub-section 3.10.

3.10.3 <or_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'OR' operator.

Multiplicity: This occurs zero or many times within the <and_test> element.

Attributes: See Sub-section 3.11.

3.10.4 <not_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'NOT' operator.

Multiplicity: This occurs zero or many times within the <and_test> element.

Attributes: See Sub-section 3.12.

3.11 <or_test> Element

Description: The <or_test> element is used to define variable test conditions at least one of which must be defined as 'True' to result in the output variables being assigned the corresponding values.

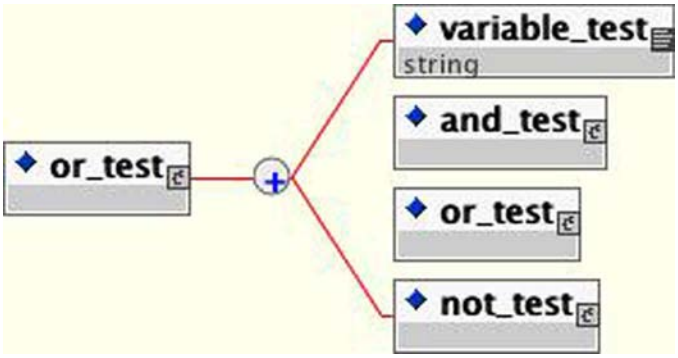


Figure 3.10 <or_test> elements.

Multiplicity: This occurs zero or more times within the <test_variable> element.

Attributes: None.

Elements:

- variable_test
- and_test
- or_test
- not_test

3.11.1 <variable_test> Element

Description: The conditional test that is to be applied to the aggregated score variables. A wide range of separate and combinatorial tests can be applied.

Multiplicity: This occurs zero or more times within the <or_test> element.

Attributes:

- **varname (optional. Default = 'SCORE').** The name of the variable that is to be tested. The default name is 'SCORE'.
  Data-type = String (max of 256 chars).
- **testoperator (mandatory with selection from the enumerated list of: EQ, NEQ, LT, LTE, GT, GTE).**
  Identifies the nature of the variable comparison that is to be applied.
  Data-type = string (1-16 chars).

3.11.2 <and_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'AND' operator.

Multiplicity: This occurs zero or many times within the <or_test> element.

Attributes: See Sub-section 3.10.

3.11.3 <or_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'OR' operator.

Multiplicity: This occurs zero or many times within the <or_test> element.

Attributes: See Sub-section 3.11.

3.11.4 <not_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'NOT' operator.

Multiplicity: This occurs zero or many times within the <or_test> element.

Attributes: See Sub-section 3.12.

3.12 <not_test> Element

Description: The <not_test> element is used to define variable test conditions that result in the condition being defined as 'False' to result in the output variables being assigned the corresponding values.

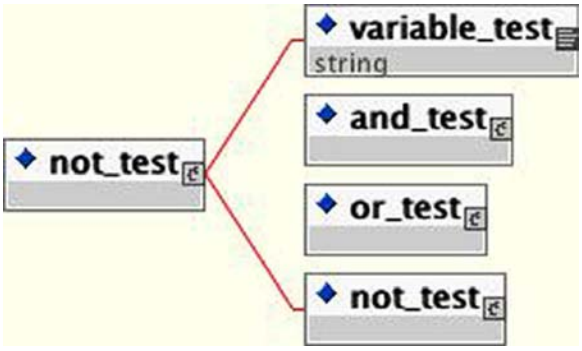


Figure 3.11 <not_test> elements.

Multiplicity: This occurs zero or more times within the <test_var> element.

Attributes: None.

Elements:

- variable_test
- and_test
- or_test
- not_test

3.12.1 <variable_test> Element

Description: The conditional test that is to be applied to the aggregated score variables. A wide range of separate and combinatorial tests can be applied.

Multiplicity: This occurs zero or once within the <not_test> element.

Attributes:

- **varname (optional. Default = 'SCORE').** The name of the variable that is to be tested. The default name is 'SCORE'.
  Data-type = String (max of 256 chars).
- **testoperator (mandatory with selection from the enumerated list of: EQ, NEQ, LT, LTE, GT, GTE).** Identifies the nature of the variable comparison that is to be applied.
  Data-type = string (1-16 chars).

3.12.2 <and_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'AND' operator.

Multiplicity: This occurs zero or once within the <not_test> element.

Attributes: See Sub-section 3.10.

3.12.3 <or_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'OR' operator.

Multiplicity: This occurs zero or once within the <not_test> element.

Attributes: See Sub-section 3.11.

3.12.4 <not_test> Element

Description: This element allows the construction of complex variable test rules to be built based upon the logical 'NOT' operator.

Multiplicity: This occurs zero or many times within the <not_test> element.

Attributes: See Sub-section 3.12.

4. Best Practice & Implementation Guide

4.1 Overall Data Model

The overall IMS QTI ASI Outcomes Processing data model is shown in Figure 4.1.

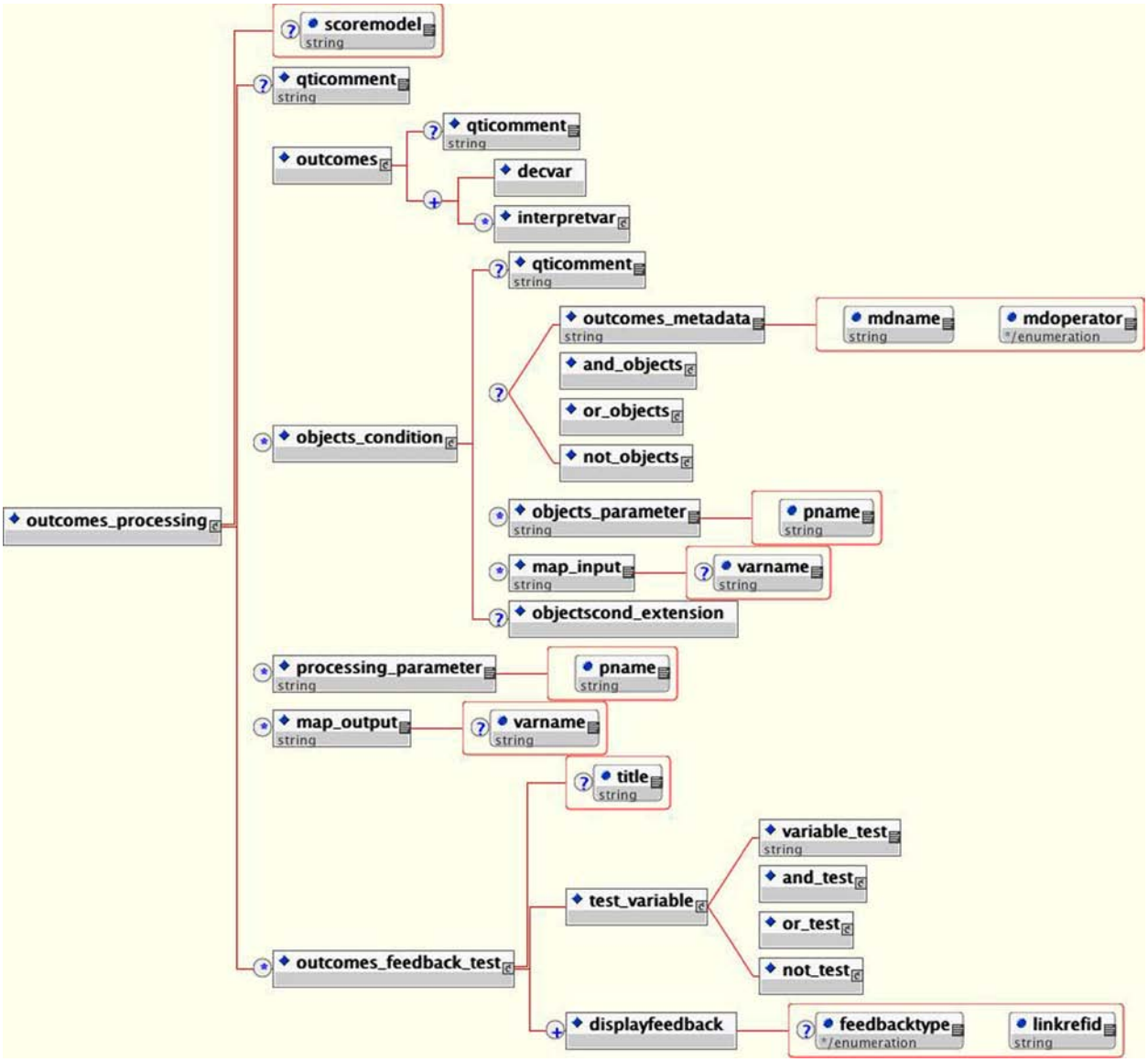


Figure 4.1 Overall ASI outcomes processing data model.

4.2 Relationship to the Other IMS Specifications

The relationship of this 'Outcomes Processing' specification to the other parts of the IMS QTI specifications is:

- The 'Outcomes Processing' root element is a part of the Assessment and Section core data objects in the IMS QTI ASI: Information Model and IMS QTI ASI: XML Binding. The <response_processing> element (defined within the QTI ASI Information model and XML Binding documents) defines the variables that are used as inputs to the response processing when the algorithm operates on items. Similarly, the <outcomes_processing> element

(defined in this document) defines the variables that are used as inputs when the outcomes processing operates on sections. Note that in predefined algorithms, these operate on either an integer variable 'SCORE' or a boolean variable 'CORRECT'. To be used with the maximum number of outcomes processing algorithms an item's <response_processing> element should set both variables. Also, some algorithms treat items that were not attempted or not presented differently from those which were attempted so there must be some way of querying the items attempted or presented status. Finally, other algorithms assume that a default value for the outcome variable (usually a null score) is available even if the item was never presented. All of these data can be extracted from a IMS QTI results report, however, most implementations will probably use equivalent internal data structures;

- The 'Selection & Ordering' component of the IMS QTI specification [QTI, 02d] interacts with the 'Outcomes Processing' part. This interaction is a result of the fact that the aggregated score for a Section and/or Assessment depends upon which Items and Section were selected for presentation to the participant;
- The IMS QTI Results Reporting specification contains the description of how the results are recorded for a participant that has undertaken an evaluation. These results will depend upon the aggregated scores derived from the outcomes processing mechanism. The QTI Results Reporting specification describes how the results from any IMS QTI-compliant Assessment, Section and Item can be reported.

4.3 Basic Example XML Instances

The following set of basic examples demonstrates how the different use-cases are supported. Table 4.1 lists the examples used to show the operation of each of the outcomes processing algorithms.

Table 4.1 The examples demonstrating each outcomes processing algorithm.

Algorithm	Example								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Number Correct	*								
Number Correct Attempted	*								
Weighted Number Correct		*							
Weighted Number Correct Attempted		*							
Parameter Weighted Number Correct		*							
Parameter Weighted Number Correct Attempted		*							
Sum of Scores			*				*	*	*
Sum of Scores Attempted			*						
Weighted Sum of Scores				*					
Weighted Sum of Scores Attempted				*					
Parameter Weighted Sum of Scores				*					
Parameter Weighted Sum of Scores Attempted				*					
Best K from N					*				
Guessing Penalty						*			

4.3.1 Example ('Number Correct' and 'Number Correct Attempted')

A Section consists of 10 Items each of which has a 'score' assigned to its default scoring boolean variable (CORRECT) that is used to denote if the Item has been answered correctly i.e. 'True'. The number of correct responses is assigned to the default score variable for the Section (COUNT). The outcomes processing algorithms for calculating the number of correct responses both in terms of the Items presented and the Items presented and attempted are given below.

```
<questestinterop>
   <section title = "IMSQTI_Outcomes_Processing_S01" ident = "IMS_QTIV1p2_S_OUT_01">
      <outcomes_processing scoremodel = "NumberCorrect">
         <outcomes>
            <decvar defaultval = "0" varname = "COUNT" vartype = "Integer"
                    minvalue = "0" maxvalue = "10" cutvalue = "5"/>
         </outcomes>
      </outcomes_processing>
      <outcomes_processing scoremodel = "NumberCorrectAttempted">
         <outcomes>
            <decvar defaultval = "0" varname = "COUNT_Attempted" vartype = "Integer"
                    minvalue = "0" maxvalue = "10" cutvalue = "5"/>
         </outcomes>
         <map_output varname="COUNT">COUNT_Attempted</map_output>
      </outcomes_processing>
      <selection_ordering>
         <selection/>
         <order order_type = "Sequential"/>
      </selection_ordering>
      <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
```

```
51          …
52          <resprocessing>
53             <outcomes>
54                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
55             </outcomes>
56             …
57          </resprocessing>
58          …
59       </item>
60       <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
61          …
62          <resprocessing>
63             <outcomes>
64                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
65             </outcomes>
66             …
67          </resprocessing>
68          …
69       </item>
70       <item ident = "IMS_QTIV1p2_I_OUT_06" maxattempts = "2">
71          …
72          <resprocessing>
73             <outcomes>
74                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
75             </outcomes>
76             …
77          </resprocessing>
78          …
79       </item>
80       <item ident = "IMS_QTIV1p2_I_OUT_07" maxattempts = "2">
81          …
82          <resprocessing>
83             <outcomes>
84                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
85             </outcomes>
86             …
87          </resprocessing>
88          …
89       </item>
90       <item ident = "IMS_QTIV1p2_I_OUT_08" maxattempts = "2">
91          …
92          <resprocessing>
93             <outcomes>
94                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
95             </outcomes>
96             …
97          </resprocessing>
98          …
99       </item>
100      <item ident = "IMS_QTIV1p2_I_OUT_09" maxattempts = "2">
101         …
102         <resprocessing>
103            <outcomes>
104               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
105            </outcomes>
106            …
107         </resprocessing>
108         …
109      </item>
110      <item ident = "IMS_QTIV1p2_I_OUT_10" maxattempts = "2">
```

```
111          …
112          <resprocessing>
113             <outcomes>
114                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
115             </outcomes>
116             …
117          </resprocessing>
118          …
119       </item>
120    </section>
121 </questestinterop>
```

The ‘NumberCorrect’ scoring algorithm that is supported for the Section is:

```
COUNT = 0
COUNT.min = 0
COUNT.max = 0
COUNT.normalized = 0.0
for ALL_PRESENTED_ITEMS do
   if ITEM_HAS_VARIABLE(CORRECT) = True
   then
      begin
         COUNT.max = COUNT.max + 1
         if item.CORRECT = True
         then COUNT = COUNT + 1
      endif
   enddo
COUNT.normalized = COUNT / COUNT.max
```

In the case where eight Items have been attempted and seven have been answered correctly then the final set of Section variable states will be:

COUNT = 7

COUNT.min = 0

COUNT.max = 10

COUNT.normalized = 0.7

The ‘NumberCorrectAttempted’ scoring algorithm that is supported for the Section is (the variable ‘COUNT_Attempted’ must be used to ensure that there is no scoping clash when two algorithms attempt to operate on the COUNT variable):

```
COUNT_Attempted = 0
COUNT_Attempted.min = 0
COUNT_Attempted.max = 0
COUNT_Attempted.normalized = 0
for ALL_ATTEMPTED_ITEMS do
   if ITEM_HAS_VARIABLE(CORRECT) = True
   then
      begin
         COUNT_Attempted.max = COUNT_Attempted.max + 1
         if item.CORRECT = True
         then COUNT_Attempted = COUNT_Attempted + 1
      endif
   enddo
COUNT_Attempted.normalized = COUNT_Attempted / COUNT_Attempted.max
```

In the case where eight Items have been attempted and seven have been answered correctly then the final set of Section variable states will be:

COUNT_Attempted = 7

COUNT_Attempted.min = 0

COUNT_Attempted.max = 8

COUNT_Attempted.normalized = 0.875

This example is available in the file: 'ims_qtiasiv1p2/outcome/basicoutexample01.xml'. All more complete description of the 'Number Correct' and 'NumberCorrectAttempted' scoring algorithms is given in Appendices B1 and B2.

4.3.2 Example ('Weighted Number Correct', 'Weighted Number Correct Attempted', 'Parameter Weighted Number Correct' and 'Parameter Weighted Number Correct Attempted')

A Section consists of 10 Items (a mixture of implicit and referenced) and each Item has a 'score' assigned to its default scoring boolean variable (CORRECT) that is used to denote if the Item has been answered correctly i.e. 'True'. The four different weighting algorithms are presented using the appropriate weighting calculation mechanism. The number of correct responses is assigned to the default score variable for the Section (COUNT).

```
1   <questestinterop>
2      <section title = "IMSQTI_Outcomes_Processing_S02" ident = "IMS_QTIV1p2_S_OUT_02">
3         <outcomes_processing scoremodel = "WeightedNumberCorrect">
4            <outcomes>
5               <decvar defaultval = "0" varname = "COUNT_WNC" vartype = "Integer"
6                     minvalue = "0" maxvalue = "15" cutvalue = "10"/>
7            </outcomes>
8            <map_output varname="COUNT">COUNT_WNC</map_output>
9         </outcomes_processing>
10        <outcomes_processing scoremodel = "WeightedNumberCorrectAttempted">
11           <outcomes>
12              <decvar defaultval = "0" varname = "COUNT_WNCA" vartype = "Integer"
13                    minvalue = "0" maxvalue = "15" cutvalue = "10"/>
14           </outcomes>
15           <map_output varname="COUNT">COUNT_WNCA</map_output>
16        </outcomes_processing>
17        <outcomes_processing scoremodel = "ParameterWeightedNumberCorrect">
18           <outcomes>
19              <decvar defaultval = "0" varname = "COUNT_PWNC" vartype = "Integer"
20                    minvalue = "0" maxvalue = "20" cutvalue = "10"/>
21           </outcomes>
22           <objects_condition>
23              <objects_parameter pname="qmd_weighting">2<objects_parameter>
24           </objects_condition>
25           <map_output varname="COUNT">COUNT_PWNC</map_output>
26        </outcomes_processing>
27        <outcomes_processing scoremodel = "ParameterWeightedNumberCorrectAttempted">
28           <outcomes>
29              <decvar defaultval = "0" varname = "COUNT_PWNC" vartype = "Integer"
30                    minvalue = "0" maxvalue = "20" cutvalue = "10"/>
31           </outcomes>
32           <objects_condition>
33              <objects_parameter pname="qmd_weighting">2<objects_parameter>
34           </objects_condition>
35           <map_output varname="COUNT">COUNT_PWNC</map_output>
36        </outcomes_processing>
37        <selection_ordering>
38           <selection/>
39           <order order_type = "Sequential"/>
40        </selection_ordering>
41        <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
```

```
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>1</fieldentry>
            </qtimetadata>
         </itemmetadata>
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>1</fieldentry>
            </qtimetadata>
         </itemmetadata>
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>1</fieldentry>
            </qtimetadata>
         </itemmetadata>
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>2</fieldentry>
            </qtimetadata>
         </itemmetadata>
         …
```

```
101          <resprocessing>
102             <outcomes>
103                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
104             </outcomes>
105             …
106          </resprocessing>
107          …
108       </item>
109       <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
110          <itemmetadata>
111             <qtimetadata>
112                <qtimetadatafield>
113                   <fieldlabel>qmd_weighting</fieldlabel>
114                   <fieldentry>1</fieldentry>
115             </qtimetadata>
116          </itemmetadata>
117          …
118          <resprocessing>
119             <outcomes>
120                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
121             </outcomes>
122             …
123          </resprocessing>
124          …
125       </item>
126       <item ident = "IMS_QTIV1p2_I_OUT_06" maxattempts = "2">
127          <itemmetadata>
128             <qtimetadata>
129                <qtimetadatafield>
130                   <fieldlabel>qmd_weighting</fieldlabel>
131                   <fieldentry>2</fieldentry>
132             </qtimetadata>
133          </itemmetadata>
134          …
135          <resprocessing>
136             <outcomes>
137                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
138             </outcomes>
139             …
140          </resprocessing>
141          …
142       </item>
143       <item ident = "IMS_QTIV1p2_I_OUT_07" maxattempts = "2">
144          <itemmetadata>
145             <qtimetadata>
146                <qtimetadatafield>
147                   <fieldlabel>qmd_weighting</fieldlabel>
148                   <fieldentry>1</fieldentry>
149             </qtimetadata>
150          </itemmetadata>
151          …
152          <resprocessing>
153             <outcomes>
154                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
155             </outcomes>
156             …
157          </resprocessing>
158          …
159       </item>
160       <item ident = "IMS_QTIV1p2_I_OUT_08" maxattempts = "2">
```

```
          <itemmetadata>
             <qtimetadata>
                <qtimetadatafield>
                   <fieldlabel>qmd_weighting</fieldlabel>
                   <fieldentry>2</fieldentry>
             </qtimetadata>
          </itemmetadata>
          …
          <resprocessing>
             <outcomes>
                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
             </outcomes>
             …
          </resprocessing>
          …
       </item>
       <item ident = "IMS_QTIV1p2_I_OUT_09" maxattempts = "2">
          <itemmetadata>
             <qtimetadata>
                <qtimetadatafield>
                   <fieldlabel>qmd_weighting</fieldlabel>
                   <fieldentry>1</fieldentry>
             </qtimetadata>
          </itemmetadata>
          …
          <resprocessing>
             <outcomes>
                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
             </outcomes>
             …
          </resprocessing>
          …
       </item>
       <item ident = "IMS_QTIV1p2_I_OUT_10" maxattempts = "2">
          <itemmetadata>
             <qtimetadata>
                <qtimetadatafield>
                   <fieldlabel>qmd_weighting</fieldlabel>
                   <fieldentry>2</fieldentry>
             </qtimetadata>
          </itemmetadata>
          …
          <resprocessing>
             <outcomes>
                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
             </outcomes>
             …
          </resprocessing>
          …
       </item>
    </section>
</questestinterop>
```

The ‘WeightedNumberCorrect’ scoring algorithm that is supported for the Section is:

```
COUNT_WNC = 0
COUNT_WNC.min = 0
COUNT_WNC.max = 0
COUNT_WNC.normalized = 0
for ALL_SELECTED_ITEMS do
```

```
      if ITEM_HAS_VARIABLE(COUNT) = True
      then
         begin
            item.weighting = item(qmd_weighting)
            COUNT_WNC.max = COUNT_WNC.max + item.weighting
            if item.CORRECT = True
            then COUNT_WNC = COUNT_WNC + item.weighting
         endif
      enddo
   COUNT_WNC.normalized = COUNT_WNC / COUNT_WNC.max
```

In the case where the first eight Items have been attempted and the first seven have been answered correctly then the final set of Section variable states will be:

COUNT_WNC = 10

COUNT_WNC.min = 0

COUNT_WNC.max = 15

COUNT_WNC.normalized = 0.667

The ‘WeightedNumberCorrectAttempted’ scoring algorithm that is supported for the Section is:

```
COUNT_WNCA = 0
COUNT_WNCA.min = 0
COUNT_WNCA.max = 0
COUNT_WNCA.normalized = 0
for ALL_ATTEMPTED_ITEMS do
   if ITEM_HAS_VARIABLE(COUNT) = True
   then
      begin
         item.weighting = item(qmd_weighting)
         COUNT_WNCA.max = COUNT_WNCA.max + item.weighting
         if item.CORRECT = True
         then COUNT_WNCA = COUNT_WNCA + item.weighting
      endif
   enddo
COUNT_WNCA.normalized = COUNT_WNCA / COUNT_WNCA.max
```

In the case where the first eight Items have been attempted and the first seven have been answered correctly then the final set of Section variable states will be:

COUNT_WCNA = 10

COUNT_WCNA.min = 0

COUNT_WCNA.max = 12

COUNT_WCNA.normalized = 0.833

The ‘ParameterWeightedNumberCorrect’ scoring algorithm that is supported for the Section is:

```
COUNT_PWNC = 0
COUNT_PWNC.min = 0
COUNT_PWNC.max = 0
COUNT_PWNC.normalized = 0
for ALL_SELECTED_ITEMS do
   if ITEM_HAS_VARIABLE(COUNT) = True
   then
      begin
         item.weighting = objects_parameter(qmd_weighting)
         COUNT_PWNC.max = COUNT.max + item.weighting
         if item.CORRECT = True
         then COUNT_PWNC = COUNT_PWNC + item.weighting
      endif
```

```
   enddo
COUNT_PWNC.normalized = COUNT_PWNC / COUNT_PWNC.max
```

In the case where the first eight Items have been attempted and the first seven have been answered correctly then the final set of Section variable states will be:

COUNT_PWNC = 14

COUNT_PWNC.min = 0

COUNT_PWNC.max = 20

COUNT_PWNC.normalized = 0.7

The 'ParameterWeightedNumberCorrectAttempted' scoring algorithm that is supported for the Section is:

```
COUNT_PWNCA = 0
COUNT_PWNCA.min = 0
COUNT_PWNCA.max = 0
COUNT_PWNCA.normalized = 0
for ALL_ATTEMPTED_ITEMS do
   if ITEM_HAS_VARIABLE(COUNT) = True
   then
      begin
         item.weighting = objects_parameter(qmd_weighting)
         COUNT_PWNCA.max = COUNT_PWNCA.max + item.weighting
         if item.CORRECT = True
         then COUNT_PWNCA = COUNT_PWNCA + item.weighting
      endif
   enddo
COUNT_PWNCA.normalized = COUNT_PWNCA / COUNT_PWNCA.max
```

In the case where the first eight Items have been attempted and the first seven have been answered correctly then the final set of Section variable states will be:

COUNT_PWNCA = 14

COUNT_PWNCA.min = 0

COUNT_PWNCA.max = 16

COUNT_PWNCA.normalized = 0.875

This example is available in the file: 'ims_qtiasiv1p2/outcome/basicoutexample02.xml'. A more complete description of the weighted number correct' scoring algorithms is given in Appendices B3, B4, B5 and B6.

4.3.3 Example ('SumofScores' and 'SumofScoresAttempted')

A Section consists of 10 Items each of which has a score assigned to its default scoring integer variable (SCORE). A score of 1 is given for a correct answer and 0 for an incorrect answer. The sum of these scores is assigned to the default score variable for the Section (SCORE). The outcomes processing algorithms for calculating the sum of scores both in terms of the Items presented and the Items presented and attempted are given below.

```
1   <questestinterop>
2      <section title = "IMSQTI_Outcomes_Processing_S03" ident = "IMS_QTIV1p2_S_OUT_03">
3         <outcomes_processing scoremodel = "SumofScores">
4            <outcomes>
5               <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
6                       minvalue = "0" maxvalue = "10" cutvalue = "5"/>
7            </outcomes>
8         </outcomes_processing>
9         <outcomes_processing scoremodel = "SumofScoresAttempted">
10           <outcomes>
```

```
            <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
                    minvalue = "0" maxvalue = "10" cutvalue = "5"/>
         </outcomes>
         <map_output varname="SCORE">SCORE_Attempted</map_output>
      </outcomes_processing>
      <selection_ordering>
         <selection/>
         <order order_type = "Sequential"/>
      </selection_ordering>
      <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/></outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_06" maxattempts = "2">
         …
```

```
71          <resprocessing>
72             <outcomes>
73                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
74             </outcomes>
75             …
76          </resprocessing>
77          …
78       </item>
79       <item ident = "IMS_QTIV1p2_I_OUT_07" maxattempts = "2">
80          …
81          <resprocessing>
82             <outcomes>
83                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
84             </outcomes>
85             …
86          </resprocessing>
87          …
88       </item>
89       <item ident = "IMS_QTIV1p2_I_OUT_08" maxattempts = "2">
90          …
91          <resprocessing>
92             <outcomes>
93                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
94             </outcomes>
95             …
96          </resprocessing>
97          …
98       </item>
99       <item ident = "IMS_QTIV1p2_I_OUT_09" maxattempts = "2">
100         …
101         <resprocessing>
102            <outcomes>
103               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
104            </outcomes>
105            …
106         </resprocessing>
107         …
108      </item>
109      <item ident = "IMS_QTIV1p2_I_OUT_10" maxattempts = "2">
110         …
111         <resprocessing>
112            <outcomes>
113               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
114            </outcomes>
115            …
116         </resprocessing>
117         …
118      </item>
119   </section>
120 </questestinterop>
```

The ‘SumOfScores’ scoring algorithm that is supported for the Section is:

```
SCORE = 0
SCORE.min = 0
SCORE.max = 0
SCORE.normalized = 0.0
for ALL_PRESENTED_ITEMS do
   if ITEM_HAS_VARIABLE(SCORE) = True
   then
      begin
         SCORE.min = SCORE.min + item.SCORE.minvalue
```

```
            SCORE.max = SCORE.max + item.SCORE.maxvalue
            SCORE = SCORE + item.SCORE
         endif
      enddo
   SCORE.normalized = (SCORE-SCORE.min) / (SCORE.max-SCORE.min)
```

In the case where eight Items have been attempted and seven have been answered correctly then the final set of Section variable states will be:

SCORE = 7

SCORE.min = 0

SCORE.max = 10

SCORE.normalized = 0.7

The 'SumofScoresAttempted' scoring algorithm that is supported for the Section is (the default output variable name has been remapped to avoid overloading the 'SCORE' variable during the concurrent application of the two scoring algorithms):

```
SCORE_Attempted = 0
SCORE_Attempted.min = 0
SCORE_Attempted.max = 0
SCORE_Attempted.normalized = 0.0
for ALL_ATTEMPTED_ITEMS do
   if ITEM_HAS_VARIABLE(SCORE) = True
   then
      begin
         SCORE_Attempted.min = SCORE_Attempted.min + item.SCORE.minvalue
         SCORE_Attempted.max = SCORE_Attempted.max + item.SCORE.maxvalue
         SCORE_Attempted = SCORE_Attempted + item.SCORE
      endif
   enddo
SCORE_Attempted.normalized = (SCORE_Attempted - SCORE_Attempted.min) /
                             (SCORE_Attempted.max - SCORE_Attempted.min)
```

In the case where eight Items have been attempted and seven have been answered correctly then the final set of Section variable states will be:

SCORE_Attempted = 7

SCORE_Attempted.min = 0

SCORE_Attempted.max = 8

SCORE_Attempted.normalized = 0.875

This example is available in the file: 'ims_qtiasiv1p2/outcome/basicoutexample03.xml'. All more complete description of the 'Sum of Scores' and 'Sum of Scores Attempted' scoring algorithms are given in Appendix B7 and B8.

4.3.4 Example ('Weighted Sum of Scores', 'Weighted Sum of Scores Attempted', 'Parameter Weighted Sum of Scores' and 'Parameter Weighted Sum of Scores Attempted')

A Section consists of 10 Items each of which has a score assigned to its default scoring integer variable (SCORE). A score of '1' is given for a correct answer and '0' for an incorrect answer. The four different weighting algorithms are presented using the appropriate weighting calculation mechanism. The sum of these scores is assigned to the default score variable for the Section (SCORE).

```
1  <questestinterop>
2     <section title = "IMSQTI_Outcomes_Processing_S04" ident = "IMS_QTIV1p2_S_OUT_04">
```

```
3        <outcomes_processing scoremodel = "WeightedSumofScores">
4           <outcomes>
5              <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
6                     minvalue = "0" maxvalue = "19" cutvalue = "12"/>
7           </outcomes>
8           <map_output varname="SCORE">SCORE_WSOS</map_output>
9        </outcomes_processing>
10       <outcomes_processing scoremodel = "WeightedSumofScoresAttempted">
11          <outcomes>
12             <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
13                    minvalue = "0" maxvalue = "15" cutvalue = "10"/>
14          </outcomes>
15          <map_output varname="SCORE">SCORE_WSOSA</map_output>
16       </outcomes_processing>
17       <outcomes_processing scoremodel = "ParameterWeightedSumofScores">
18          <outcomes>
19             <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
20                    minvalue = "0" maxvalue = "20" cutvalue = "10"/>
21          </outcomes>
22          <objects_condition>
23             <objects_parameter pname="qmd_weighting">2<objects_parameter>
24          </objects_condition>
25          <map_output varname="SCORE">SCORE_PWSOS</map_output>
26       </outcomes_processing>
27       <outcomes_processing scoremodel = "ParameterWeightedSumofScoresAttempted">
28          <outcomes>
29             <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
30                    minvalue = "0" maxvalue = "20" cutvalue = "10"/>
31          </outcomes>
32          <objects_condition>
33             <objects_parameter pname="qmd_weighting">2<objects_parameter>
34          </objects_condition>
35          <map_output varname="SCORE">SCORE_PWSOSA</map_output>
36       </outcomes_processing>
37       <selection_ordering>
38          <selection/>
39          <order order_type = "Sequential"/>
40       </selection_ordering>
41       <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
42          <itemmetadata>
43             <qtimetadata>
44                <qtimetadatafield>
45                   <fieldlabel>qmd_weighting</fieldlabel>
46                   <fieldentry>3</fieldentry>
47             </qtimetadata>
48          </itemmetadata>
49          …
50          <resprocessing>
```

```
51              <outcomes>
52                 <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
53              </outcomes>
54              …
55           </resprocessing>
56           …
57        </item>
58        <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
59           <itemmetadata>
60              <qtimetadata>
61                 <qtimetadatafield>
62                    <fieldlabel>qmd_weighting</fieldlabel>
63                    <fieldentry>2</fieldentry>
64              </qtimetadata>
65           </itemmetadata>
66           …
67           <resprocessing>
68              <outcomes>
69                 <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
70              </outcomes>
71              …
72           </resprocessing>
73           …
74        </item>
75        <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
76           <itemmetadata>
77              <qtimetadata>
78                 <qtimetadatafield>
79                    <fieldlabel>qmd_weighting</fieldlabel>
80                    <fieldentry>1</fieldentry>
81              </qtimetadata>
82           </itemmetadata>
83           …
84           <resprocessing>
85              <outcomes>
86                 <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
87              </outcomes>
88              …
89           </resprocessing>
90           …
91        </item>
92        <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
93           <itemmetadata>
94              <qtimetadata>
95                 <qtimetadatafield>
96                    <fieldlabel>qmd_weighting</fieldlabel>
97                    <fieldentry>1</fieldentry>
98              </qtimetadata>
99           </itemmetadata>
100          …
101          <resprocessing>
102             <outcomes>
103                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
104             </outcomes>
105             …
106          </resprocessing>
107          …
108       </item>
109       <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
110          <itemmetadata>
```

```
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>1</fieldentry>
               </qtimetadata>
            </itemmetadata>
            …
            <resprocessing>
               <outcomes>
                  <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
               </outcomes>
               …
            </resprocessing>
            …
         </item>
         <item ident = "IMS_QTIV1p2_I_OUT_06" maxattempts = "2">
            <itemmetadata>
               <qtimetadata>
                  <qtimetadatafield>
                     <fieldlabel>qmd_weighting</fieldlabel>
                     <fieldentry>6</fieldentry>
               </qtimetadata>
            </itemmetadata>
            …
            <resprocessing>
               <outcomes>
                  <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
               </outcomes>
               …
            </resprocessing>
            …
         </item>
         <item ident = "IMS_QTIV1p2_I_OUT_07" maxattempts = "2">
            <itemmetadata>
               <qtimetadata>
                  <qtimetadatafield>
                     <fieldlabel>qmd_weighting</fieldlabel>
                     <fieldentry>2</fieldentry>
               </qtimetadata>
            </itemmetadata>
            …
            <resprocessing>
               <outcomes>
                  <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
               </outcomes>
               …
            </resprocessing>
            …
         </item>
         <item ident = "IMS_QTIV1p2_I_OUT_08" maxattempts = "2">
            <itemmetadata>
               <qtimetadata>
                  <qtimetadatafield>
                     <fieldlabel>qmd_weighting</fieldlabel>
                     <fieldentry>1</fieldentry>
               </qtimetadata>
            </itemmetadata>
            …
            <resprocessing>
               <outcomes>
```

```
171                  <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
172               </outcomes>
173               …
174            </resprocessing>
175            …
176         </item>
177         <item ident = "IMS_QTIV1p2_I_OUT_09" maxattempts = "2">
178            <itemmetadata>
179               <qtimetadata>
180                  <qtimetadatafield>
181                     <fieldlabel>qmd_weighting</fieldlabel>
182                     <fieldentry>1</fieldentry>
183               </qtimetadata>
184            </itemmetadata>
185            …
186            <resprocessing>
187               <outcomes>
188                  <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
189               </outcomes>
190               …
191            </resprocessing>
192            …
193         </item>
194         <item ident = "IMS_QTIV1p2_I_OUT_10" maxattempts = "2">
195            <itemmetadata>
196               <qtimetadata>
197                  <qtimetadatafield>
198                     <fieldlabel>qmd_weighting</fieldlabel>
199                     <fieldentry>1</fieldentry>
200               </qtimetadata>
201            </itemmetadata>
201            …
203            <resprocessing>
204               <outcomes>
205                  <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
206               </outcomes>
207               …
208            </resprocessing>
209            …
210         </item>
211      </section>
212   </questestinterop>
```

The ‘WeightedSumOfScores’ scoring algorithm that is supported for the Section is:

```
SCORE_WSOS = 0
SCORE_WSOS.min = 0
SCORE_WSOS.max = 0
SCORE_WSOS.normalized = 0.0
for ALL_PRESENTED_ITEMS do
   if ITEM_HAS_VARIABLE(SCORE) = True
      begin
         SCORE_WSOS.min = SCORE_WSOS.min + (item.SCORE.minvalue * item(qmd_weighting))
         SCORE_WSOS.max = SCORE_WSOS.max + (item.SCORE.maxvalue * item(qmd_weighting))
         SCORE_WSOS = SCORE_WSOS + (item.SCORE * item.qmd_weighting)
      endif
   enddo
SCORE_WSOS.normalized = (SCORE_WSOS – SCORE_WSOS.min) / (SCORE_WSOS.max – SCORE_WSOS.min)
```

In the case where the first eight Items have been attempted and the first seven have been answered correctly then the final set of Section variable states will be:

SCORE_WSOS = 16

SCORE_WSOS.min = 0

SCORE_WSOS.max = 19

SCORE_WSOS.normalized = 0.842

The ‘WeightedSumOfScoresAttempted’ scoring algorithm that is supported for the Section is:

```
SCORE_WSOSA = 0
SCORE_WSOSA.min = 0
SCORE_WSOSA.max = 0
SCORE_WSOSA.normalized = 0.0
for ALL_ATTEMPTED_ITEMS do
   if ITEM_HAS_VARIABLE(SCORE) = True
      begin
         SCORE_WSOSA.min = SCORE_WSOSA.min + (item.SCORE.minvalue * item(qmd_weighting))
         SCORE_WSOSA.max = SCORE_WSOSA.max + (item.SCORE.maxvalue * item(qmd_weighting))
         SCORE_WSOSA = SCORE_WSOSA + (item.SCORE * item(qmd_weighting))
      endif
   enddo
SCORE_WSOSA.normalized = (SCORE_WSOSA–SCORE_WSOSA.min)/(SCORE_WSOSA.max–SCORE_WSOSA.min)
```

In the case where the first eight Items have been attempted and the first seven have been answered correctly then the final set of Section variable states will be:

SCORE_WSOSA = 16

SCORE_WSOSA.min = 0

SCORE_WSOSA.max = 17

SCORE_WSOSA.normalized = 0.941

The ‘ParameterWeightedSumOfScores’ scoring algorithm that is supported for the Section is:

```
SCORE_PWSOS = 0
SCORE_PWSOS.min = 0
SCORE_PWSOS.max = 0
SCORE_PWSOS.normalized = 0.0
for ALL_PRESENTED_ITEMS do
   if ITEM_HAS_VARIABLE(SCORE) = True
      begin
         SCORE_PWSOS.min = SCORE_PWSOS.min +
                        (item.SCORE.minvalue * objects_parameter(qmd_weighting))
         SCORE_PWSOS.max = SCORE_PWSOS.max +
                        (item.SCORE.maxvalue * objects_parameter(qmd_weighting))
         SCORE_PWSOS = SCORE_PWSOS + (item.SCORE * objects_parameter(qmd_weighting))
      endif
   enddo
SCORE_PWSOS.normalized = (SCORE_PWSOS–SCORE_PWSOS.min)/(SCORE_PWSOS.max–SCORE_PWSOS.min)
```

In the case where the first eight Items have been attempted and the first seven have been answered correctly then the final set of Section variable states will be:

SCORE_PWSOS = 14

SCORE_PWSOS.min = 0

SCORE_PWSOS.max = 20

SCORE_PWSOS.normalized = 0.7

The ‘ParameterWeightedSumOfScoresAttempted’ scoring algorithm that is supported for the Section is:

```
SCORE_PWSOSA = 0
SCORE_PWSOSA.min = 0
SCORE_PWSOSA.max = 0
SCORE_PWSOSA.normalized = 0.0
for ALL_PRESENTED_ITEMS do
   if ITEM_HAS_VARIABLE(SCORE) = True
      begin
         SCORE_PWSOSA.min = SCORE_PWSOSA.min +
                        (item.SCORE.minvalue * objects_parameter(qmd_weighting))
         SCORE_PWSOSA.max= SCORE_PWSOSA.max +
                        (item.SCORE.maxvalue * objects_parameter(qmd_weighting))
         SCORE_PWSOSA = SCORE_PWSOSA + (item.SCORE * objects_parameter(qmd_weighting))
      endif
   enddo
SCORE_PWSOSA.normalized = (SCORE_PWSOSA–SCORE_PWSOSA.min) /
                        (SCORE_PWSOSA.max–SCORE_PWSOSA.min)
```

In the case where the first eight Items have been attempted and the first seven have been answered correctly then the final set of Section variable states will be:

SCORE_PWSOSA = 14

SCORE_PWSOSA.min = 0

SCORE_PWSOSA.max = 16

SCORE_PWSOSA.normalized = 0.875

This example is available in the file: ‘ims_qtiasiv1p2/outcome/basicoutexample04.xml’. All more complete description of the ‘Sum of Scores’ scoring algorithm is given in Appendices B9, B10, B11 and B12.

4.3.5 Example (‘Best K from N’)

A Section consists of 10 Items and each has a score assigned to its default scoring integer variable (SCORE). The ‘maxvalue’ score is given for a correct answer and the ‘minvalue’ for an incorrect answer. The sum of these scores is assigned to the default score variable for the Section (SCORE). The aggregated score is set as the best seven scores.

```
1   <questestinterop>
2      <section title = "IMSQTI_Outcomes_Processing_S05" ident = "IMS_QTIV1p2_S_OUT_05">
3         <outcomes_processing scoremodel = "BestKofN">
4            <outcomes>
5               <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
6                       minvalue = "0" maxvalue = "10" cutvalue = "5"/>
7            </outcomes>
8            <processing_parameter pname = "BestK">7</processing_parameter>
9         </outcomes_processing>
10        <selection_ordering>
11           <selection/>
12           <order order_type = "Sequential"/>
13        </selection_ordering>
14        <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
15           …
16           <resprocessing>
17              <outcomes>
18                 <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
19              </outcomes>
20              …
21           </resprocessing>
22           …
23        </item>
24        <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
```

```
25          …
26          <resprocessing>
27             <outcomes>
28                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
29             </outcomes>
30             …
31          </resprocessing>
32          …
33       </item>
34       <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
35          …
36          <resprocessing>
37             <outcomes>
38                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
39             </outcomes>
40             …
41          </resprocessing>
42          …
43       </item>
44       <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
45          …
46          <resprocessing>
47             <outcomes>
48                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
49             </outcomes>
50             …
51          </resprocessing>
52          …
53       </item>
54       <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
55          …
56          <resprocessing>
57             <outcomes>
58                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/></outcomes>
59             …
60          </resprocessing>
61          …
62       </item>
63       <item ident = "IMS_QTIV1p2_I_OUT_06" maxattempts = "2">
64          …
65          <resprocessing>
66             <outcomes>
67                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
68             </outcomes>
69             …
70          </resprocessing>
71          …
72       </item>
73       <item ident = "IMS_QTIV1p2_I_OUT_07" maxattempts = "2">
74          …
75          <resprocessing>
76             <outcomes>
77                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
78             </outcomes>
79             …
80          </resprocessing>
81          …
82       </item>
83       <item ident = "IMS_QTIV1p2_I_OUT_08" maxattempts = "2">
```

```
84          …
85          <resprocessing>
86             <outcomes>
87                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
88             </outcomes>
89             …
90          </resprocessing>
91          …
92       </item>
93       <item ident = "IMS_QTIV1p2_I_OUT_09" maxattempts = "2">
94          …
95          <resprocessing>
96             <outcomes>
97                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
98             </outcomes>
99             …
100         </resprocessing>
101         …
102      </item>
103      <item ident = "IMS_QTIV1p2_I_OUT_10" maxattempts = "2">
104         …
105         <resprocessing>
106            <outcomes>
107               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
108            </outcomes>
109            …
110         </resprocessing>
111         …
112      </item>
113   </section>
114</questestinterop>
```

The ‘BestKfromN’ scoring algorithm that is supported for the Section is:

```
SCORE = 0
SCORE.min = 0
SCORE.max = 0
SCORE.normalized = 0.0
For HIGHEST_SCORES_FROM_PRESENTED_ITEMS(7) and ITEM_HAS_VARIABLE(SCORE) do
   begin
      SCORE.min = SCORE.min + item.SCORE.minvalue
      SCORE.max = SCORE.max + item.SCORE.maxvalue
      SCORE = SCORE + item.SCORE
   Enddo
SCORE.normalized = (SCORE – SCORE.min) / (SCORE.max – SCORE.min)
```

In the case where all ten Items have been attempted and seven have been answered correctly then the final set of Section variable states will be:

SCORE = 7

SCORE.min = 0

SCORE.max = 7

SCORE.normalized = 1

This example is available in the file: ‘ims_qtiasiv1p2/outcome/basicoutexample05.xml’. A more detailed description of these scoring algorithms are presented in Appendices B13.

4.3.6 Example ('Guessing Penalty' and 'WeightedGuessingPenalty')

A Section consists of 10 Items each of which has a 'score' assigned to its default scoring boolean variable (CORRECT) that is used to denote if the Item has been answered correctly i.e. 'True'. The number of correct responses is assigned to the default score variable for the Section (COUNT). The outcomes processing algorithms for calculating the number of correct responses both in terms of the Items presented and the Items presented and attempted are given below.

```
<questestinterop>
   <section title = "IMSQTI_Outcomes_Processing_S01" ident = "IMS_QTIV1p2_S_OUT_01">
      <outcomes_processing scoremodel = "GuessingPenalty">
         <outcomes>
            <decvar defaultval = "0" varname = "COUNT" vartype = "Decimal"
                    minvalue = "0" maxvalue = "10" cutvalue = "5"/>
         </outcomes>
      </outcomes_processing>
      <outcomes_processing scoremodel = "WeightedGuessingPenalty">
         <outcomes>
            <decvar defaultval = "0" varname = "COUNT_Weighted" vartype = "Integer"
                    minvalue = "0" maxvalue = "10" cutvalue = "5"/>
         </outcomes>
         <map_output varname="COUNT">COUNT_Weighted</map_output>
      </outcomes_processing>
      <selection_ordering>
         <selection/>
         <order order_type = "Sequential"/>
      </selection_ordering>
      <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>1</fieldentry>
            </qtimetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_penaltyvalue</fieldlabel>
                  <fieldentry>0.2</fieldentry>
            </qtimetadata>
         </itemmetadata>
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>1</fieldentry>
            </qtimetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_penaltyvalue</fieldlabel>
                  <fieldentry>0.2</fieldentry>
```

```
            </qtimetadata>
         </itemmetadata>
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>1</fieldentry>
            </qtimetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_penaltyvalue</fieldlabel>
                  <fieldentry>0.2</fieldentry>
            </qtimetadata>
         </itemmetadata>
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>2</fieldentry>
            </qtimetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_penaltyvalue</fieldlabel>
                  <fieldentry>0.2</fieldentry>
            </qtimetadata>
         </itemmetadata>
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
```

```
                  <fieldentry>1</fieldentry>
               </qtimetadata>
               <qtimetadata>
                  <qtimetadatafield>
                     <fieldlabel>qmd_penaltyvalue</fieldlabel>
                     <fieldentry>0.2</fieldentry>
               </qtimetadata>
            </itemmetadata>
            …
            <resprocessing>
               <outcomes>
                  <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
               </outcomes>
               …
            </resprocessing>
            …
         </item>
         <item ident = "IMS_QTIV1p2_I_OUT_06" maxattempts = "2">
            <itemmetadata>
               <qtimetadata>
                  <qtimetadatafield>
                     <fieldlabel>qmd_weighting</fieldlabel>
                     <fieldentry>2</fieldentry>
               </qtimetadata>
               <qtimetadata>
                  <qtimetadatafield>
                     <fieldlabel>qmd_penaltyvalue</fieldlabel>
                     <fieldentry>0.2</fieldentry>
               </qtimetadata>
            </itemmetadata>
            …
            <resprocessing>
               <outcomes>
                  <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
               </outcomes>
               …
            </resprocessing>
            …
         </item>
         <item ident = "IMS_QTIV1p2_I_OUT_07" maxattempts = "2">
            <itemmetadata>
               <qtimetadata>
                  <qtimetadatafield>
                     <fieldlabel>qmd_weighting</fieldlabel>
                     <fieldentry>1</fieldentry>
               </qtimetadata>
               <qtimetadata>
                  <qtimetadatafield>
                     <fieldlabel>qmd_penaltyvalue</fieldlabel>
                     <fieldentry>0.2</fieldentry>
               </qtimetadata>
            </itemmetadata>
            …
            <resprocessing>
               <outcomes>
                  <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
               </outcomes>
               …
            </resprocessing>
```

```
172            …
173         </item>
174         <item ident = "IMS_QTIV1p2_I_OUT_08" maxattempts = "2">
175            <itemmetadata>
176               <qtimetadata>
177                  <qtimetadatafield>
178                     <fieldlabel>qmd_weighting</fieldlabel>
179                     <fieldentry>2</fieldentry>
180               </qtimetadata>
181               <qtimetadata>
182                  <qtimetadatafield>
183                     <fieldlabel>qmd_penaltyvalue</fieldlabel>
184                     <fieldentry>0.2</fieldentry>
185               </qtimetadata>
186            </itemmetadata>
187            …
188            <resprocessing>
189               <outcomes>
190                  <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
191               </outcomes>
192               …
193            </resprocessing>
194            …
195         </item>
196         <item ident = "IMS_QTIV1p2_I_OUT_09" maxattempts = "2">
197            <itemmetadata>
198               <qtimetadata>
199                  <qtimetadatafield>
200                     <fieldlabel>qmd_weighting</fieldlabel>
201                     <fieldentry>1</fieldentry>
202               </qtimetadata>
203               <qtimetadata>
204                  <qtimetadatafield>
205                     <fieldlabel>qmd_penaltyvalue</fieldlabel>
206                     <fieldentry>0.2</fieldentry>
207               </qtimetadata>
208            </itemmetadata>
209            …
210            <resprocessing>
211               <outcomes>
212                  <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
213               </outcomes>
214               …
215            </resprocessing>
216            …
217         </item>
218         <item ident = "IMS_QTIV1p2_I_OUT_10" maxattempts = "2">
219            <itemmetadata>
220               <qtimetadata>
221                  <qtimetadatafield>
222                     <fieldlabel>qmd_weighting</fieldlabel>
223                     <fieldentry>2</fieldentry>
224               </qtimetadata>
225               <qtimetadata>
226                  <qtimetadatafield>
227                     <fieldlabel>qmd_penaltyvalue</fieldlabel>
228                     <fieldentry>0.2</fieldentry>
229               </qtimetadata>
230            </itemmetadata>
```

```
231          …
232          <resprocessing>
233             <outcomes>
234                <decvar defaultval="False" vartype="Boolean" varname="CORRECT"/>
235             </outcomes>
236             …
237          </resprocessing>
238          …
239       </item>
240    </section>
241 </questestinterop>
```

The ‘GuessingPenalty’ scoring algorithm that is supported for the Section is:

```
COUNT = 0
COUNT.correct = 0
COUNT.incorrect = 0
COUNT.unattempted = 0
for ALL_PRESENTED(childobjects)and (HAS_VARIABLE(CORRECT)  do
   begin
      if child.object(ATTEMPTED) = True
      then
         begin
            if childobject.CORRECT = True
            then
               begin
                  parentobject.COUNT.correct = parentobject.COUNT.correct + 1
                  parentobject.COUNT = parentobject.COUNT + 1
               end
            else
               begin
                  parentobject.COUNT.incorrect = parentobject.COUNT.incorrect + 1
                  parentobject.COUNT = parentobject.COUNT – item(qmd_penaltyvalue)
               end
         endif
      else childobject.COUNT.unattempted = childobject.COUNT.unattempted + 1
   enddo
```

In the case where eight Items have been attempted and seven have been answered correctly then the final set of Section variable states will be:

COUNT = 6.8

COUNT.correct = 7

COUNT.correct = 7

COUNT.incorrect = 1

COUNT.unattempted = 2

The ‘WeightedGuessingPenalty’ scoring algorithm that is supported for the Section is:

```
COUNT_Weighted = 0
COUNT_Weighted.correct = 0
COUNT_Weighted.incorrect = 0
COUNT_Weighted.unattempted = 0
for ALL_PRESENTED(childobjects)and (HAS_VARIABLE(CORRECT)  do
   begin
      if child.object(ATTEMPTED) = True
      then
         begin
```

```
            if childobject.CORRECT = True
            then
               begin
                  parentobject.COUNT_Weighted.correct =
                     parentobject.COUNT_Weighted.correct + 1
                  parentobject.COUNT_Weighted = parentobject.COUNT_Weighted + 1
               end
            else
               begin
                  parentobject.COUNT_Weighted.incorrect =
                     parentobject.COUNT_Weighted.incorrect + 1
                  parentobject.COUNT_Weighted =
                  (parentobject.COUNT_Weighted * item(qmd_weighting) –
                  item(qmd_penaltyvalue)
               end
         endif
      else childobject.COUNT_Weighted.unattempted =
         childobject.COUNT_Weighted.unattempted + 1
   enddo
```

In the case where eight Items have been attempted and seven have been answered correctly then the final set of Section variable states will be:

COUNT = 13.8

COUNT.correct = 7

COUNT.correct = 7

COUNT.incorrect = 1

COUNT.unattempted = 2

This example is available in the file: 'ims_qtiasiv1p2/outcome/basicoutexample06.xml'. All more complete description of these scoring algorithm is given in Appendices B14 and B15.

4.3.7 Example (Remapped Output 'Sum of Scores')

A Section consists of 5 Items (a mixture of implicit and referenced) and each Item has a score assigned to its default scoring integer variable (SCORE). A score of 1 is given for a correct answer and –1 for an incorrect answer. The sum of these scores is assigned to the new score variable for the Section (Section_06_SCORE).

```
<questestinterop>
   <section title = "IMSQTI_Outcomes_Processing_S07" ident = "IMS_QTIV1p2_S_OUT_07">
      <outcomes_processing scoremodel = "SumofScores">
         <outcomes>
            <decvar defaultval= "0" varname= "Section_07_SCORE" vartype= "Integer"
                    minvalue= "0" maxvalue= "10" cutvalue= "6"/>
         </outcomes>
         <map_output varname = "SCORE">Section_07_SCORE</map_output>
      </outcomes_processing>
      <selection_ordering>
         <selection/>
         <order order_type = "Sequential"/>
      </selection_ordering>
      <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
            </outcomes>
```

```
60              …
61           </resprocessing>
62           …
63        </item>
64        <item ident = "IMS_QTIV1p2_I_OUT_06" maxattempts = "2">
65           …
66           <resprocessing>
67              <outcomes>
68                 <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
69              </outcomes>
70              …
71           </resprocessing>
72           …
73        </item>
74        <item ident = "IMS_QTIV1p2_I_OUT_07" maxattempts = "2">
75           …
76           <resprocessing>
77              <outcomes>
78                 <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
79              </outcomes>
80              …
81           </resprocessing>
82           …
83        </item>
84        <item ident = "IMS_QTIV1p2_I_OUT_08" maxattempts = "2">
85           …
86           <resprocessing>
87              <outcomes>
88                 <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
89              </outcomes>
90              …
91           </resprocessing>
92           …
93        </item>
94        <item ident = "IMS_QTIV1p2_I_OUT_09" maxattempts = "2">
95           …
96           <resprocessing>
97              <outcomes>
98                 <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
99              </outcomes>
100             …
101          </resprocessing>
102          …
103       </item>
104       <item ident = "IMS_QTIV1p2_I_OUT_10" maxattempts = "2">
105          …
106          <resprocessing>
107             <outcomes>
108                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
109             </outcomes>
110             …
111          </resprocessing>
112          …
113       </item>
114    </section>
115 </questestinterop>
```

The output remapped ‘SumofScores’ scoring algorithm that is supported for the Section is:

```
Section_07_SCORE = 0
```

```
Section_07_SCORE.min = 0
Section_07_SCORE.max = 0
Section_07_SCORE.normalized = 0.0
For ALL_ATTEMPTED_ITEMS and ITEM_HAS_VARIABLE(SCORE) do
   begin
      Section_07_SCORE.min = Section_07_SCORE.min + item.SCORE.minvalue
      Section_07_SCORE.max = Section_07_SCORE.max + item.SCORE.maxvalue
      Section_07_SCORE = Section_07_SCORE + item.SCORE
   Enddo
Section_07_SCORE.normalized =
(Section_07_SCORE – Section_07_SCORE.min)/(Section_07_SCORE.max – Section_07_SCORE.min)
```

In the case where all ten Items have been attempted and eight have been answered correctly then the final set of output remapped Section variable states will be:

Section_07_SCORE = 8

Section_07_SCORE.min = 0

Section_07_SCORE.max = 10

Section_07_SCORE.normalized = 0.8

This example is available in the file: ‘ims_qtiasiv1p2/outcome/basicoutexample07.xml’.

4.3.8 Example (Remapped Input ‘Sum of Scores’)

A Section consists of 5 Items (a mixture of implicit and referenced) and each Item has a score assigned to its default scoring integer variable (SCORE). A score of 1 is given for a correct answer and –1 for an incorrect answer. The sum of these scores is assigned to the default score variable for the Section (SCORE).

```
<questestinterop>
   <section title = "IMSQTI_Outcomes_Processing_S06" ident = "IMS_QTIV1p2_S_OUT_06">
      <outcomes_processing scoremodel = "SumofScores">
         <outcomes>
            <decvar defaultval= "0" varname= "SCORE" vartype= "Integer"
                    minvalue= "0" maxvalue= "10" cutvalue= "6"/>
         </outcomes>
         <objects_condition>
            <map_input varname = "SCORE">item_SCORE</map_input>
         </objects_condition>
      </outcomes_processing>
      <selection_ordering>
         <selection seltype/>
         <order order_type = "Sequential"/>
      </selection_ordering>
      <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
```

```
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_06" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_07" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_08" maxattempts = "2">
         …
         <resprocessing>
            <outcomes>
```

```
90                <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
91             </outcomes>
92             …
93          </resprocessing>
94          …
95       </item>
96       <item ident = "IMS_QTIV1p2_I_OUT_09" maxattempts = "2">
97          …
98          <resprocessing>
99             <outcomes>
100               <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
101            </outcomes>
102            …
103         </resprocessing>
104         …
105      </item>
106      <item ident = "IMS_QTIV1p2_I_OUT_10" maxattempts = "2">
107         …
108         <resprocessing>
109            <outcomes>
110               <decvar defaultval="0" minvalue="0" maxvalue="1" varname="item_SCORE"/>
111            </outcomes>
112            …
113         </resprocessing>
114         …
115      </item>
116   </section>
117 </questestinterop>
```

The input remapped ‘SumofScores’ scoring algorithm that is supported for the Section is:

```
SCORE = 0
SCORE.min = 0
SCORE.max = 0
SCORE.normalized = 0.0
For ALL_ATTEMPTED_ITEMS and ITEM_HAS_VARIABLE(item_SCORE) do
   begin
      SCORE.min = SCORE.min + item.item_SCORE.minvalue
      SCORE.max = SCORE.max + item.item_SCORE.maxvalue
      SCORE = SCORE + item.item_Score
   Enddo
SCORE.normalized = (SCORE – SCORE.min)/(SCORE.max – SCORE.min)
```

In the case where all ten Items have been attempted and eight have been answered correctly then the final set of Section variable states will be:

SCORE = 8

SCORE.min = 0

SCORE.max = 10

SCORE.normalized = 0.8

This example is available in the file: ‘ims_qtiasiv1p2/outcome/basicoutexample08.xml’.

4.3.9 Example (Parameterized 'Sum of Scores')

A Section consists of 10 Items and each has a score assigned to its default scoring integer variable (SCORE). A score of 1 is given for a correct answer and –1 for an incorrect answer. The sum of these scores is assigned to the default score variable for the Section (SCORE). The intention is to produce a result score for all those Items on the topic that are 'Advanced' and 'Basic'.

```
<questestinterop>
   <section title = "IMSQTI_Outcomes_Processing_S08" ident = "IMS_QTIV1p2_S_OUT_08">
      <outcomes_processing scoremodel = "SumOfScores">
         <outcomes>
            <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
                    minvalue = "0" maxvalue = "10" cutvalue = "6"/>
         </outcomes>
         <objects_condition>
            <or_objects>
               <outcomes_metadatamdname="qmd_levelofdifficulty"
                                 mdoperator= "EQ">basic</outcomes_metadata>
               <outcomes_metadatamdname="qmd_levelofdifficulty"
                                 mdoperator="EQ">advanced</outcomes_metadata>
            </or_objects>
         </objects_condition>
      </outcomes_processing>
      <selection_ordering>
         <selection/>
         <order order_type = "Sequential"/>
      </selection_ordering>
      <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>3</fieldentry>
               </qtimetadatafield>
            </qtimetadata>
            <qmd_levelofdifficulty>advanced</qmd_levelofdifficulty>
            <qmd_topic>qtiv1p2test</qmd_topic>
         </itemmetadata>
         …
         <resprocessing>
            <outcomes>
               <decvar defaultval="0" minvalue="0" maxvalue="1"/>
            </outcomes>
            …
         </resprocessing>
         …
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
         <itemmetadata>
            <qtimetadata>
               <qtimetadatafield>
                  <fieldlabel>qmd_weighting</fieldlabel>
                  <fieldentry>2</fieldentry>
               </qtimetadatafield>
            </qtimetadata>
            <qmd_levelofdifficulty>basic</qmd_levelofdifficulty>
            <qmd_topic>qtiv1p2test</qmd_topic>
         </itemmetadata>
         …
```

```
53          <resprocessing>
54             <outcomes>
55                <decvar defaultval="0" minvalue="0" maxvalue="1"/>
56             </outcomes>
57             …
58          </resprocessing>
59          …
60       </item>
61       <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
62          <itemmetadata>
63             <qtimetadata>
64                <qtimetadatafield>
65                   <fieldlabel>qmd_weighting</fieldlabel>
66                   <fieldentry>1</fieldentry>
67                </qtimetadatafield>
68             </qtimetadata>
69             <qmd_levelofdifficulty>basic</qmd_levelofdifficulty>
70             <qmd_topic>qtiv1p2test</qmd_topic>
71          </itemmetadata>
72          …
73          <resprocessing>
74             <outcomes>
75                <decvar defaultval="0" minvalue="0" maxvalue="1"/>
76             </outcomes>
77             …
78          </resprocessing>
79          …
80       </item>
81       <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
82          <itemmetadata>
83             <qtimetadata>
84                <qtimetadatafield>
85                   <fieldlabel>qmd_weighting</fieldlabel>
86                   <fieldentry>1</fieldentry>
87                </qtimetadatafield>
88             </qtimetadata>
89             <qmd_levelofdifficulty>advanced</qmd_levelofdifficulty>
90             <qmd_topic>qtiv1p2test</qmd_topic>
91          </itemmetadata>
92          …
93          <resprocessing>
94             <outcomes>
95                <decvar defaultval="0" minvalue="0" maxvalue="1"/>
96             </outcomes>
97             …
98          </resprocessing>
99          …
100      </item>
101      <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
102         <itemmetadata>
103            <qtimetadata>
104               <qtimetadatafield>
105                  <fieldlabel>qmd_weighting</fieldlabel>
106                  <fieldentry>1</fieldentry>
107               </qtimetadatafield>
108            </qtimetadata>
109            <qmd_levelofdifficulty>basic</qmd_levelofdifficulty>
110            <qmd_topic>qtiv1p2test</qmd_topic>
111         </itemmetadata>
112         …
```

```
113          <resprocessing>
114             <outcomes>
115                <decvar defaultval="0" minvalue="0" maxvalue="1"/>
116             </outcomes>
117             …
118          </resprocessing>
119          …
120       </item>
121       <item ident = "IMS_QTIV1p2_I_OUT_06" maxattempts = "2">
122          <itemmetadata>
123             <qtimetadata>
124                <qtimetadatafield>
125                   <fieldlabel>qmd_weighting</fieldlabel>
126                   <fieldentry>6</fieldentry>
127                </qtimetadatafield>
128             </qtimetadata>
129             <qmd_levelofdifficulty>basic</qmd_levelofdifficulty>
130             <qmd_topic>qtiv1p2test</qmd_topic>
131          </itemmetadata>
132          …
133          <resprocessing>
134             <outcomes>
135                <decvar defaultval="0" minvalue="0" maxvalue="1"/>
136             </outcomes>
137             …
138          </resprocessing>
139          …
140       </item>
141       <item ident = "IMS_QTIV1p2_I_OUT_07" maxattempts = "2">
142          <itemmetadata>
143             <qtimetadata>
144                <qtimetadatafield>
145                   <fieldlabel>qmd_weighting</fieldlabel>
146                   <fieldentry>2</fieldentry>
147                </qtimetadatafield>
148             </qtimetadata>
150             <qmd_levelofdifficulty>basic</qmd_levelofdifficulty>
151             <qmd_topic>qtiv1p2test</qmd_topic>
152          </itemmetadata>
153          …
154          <resprocessing>
155             <outcomes>
156                <decvar defaultval="0" minvalue="0" maxvalue="1"/>
157             </outcomes>
158             …
159          </resprocessing>
160          …
161       </item>
162       <item ident = "IMS_QTIV1p2_I_OUT_08" maxattempts = "2">
163          <itemmetadata>
164             <qtimetadata>
165                <qtimetadatafield>
166                   <fieldlabel>qmd_weighting</fieldlabel>
167                   <fieldentry>1</fieldentry>
168                </qtimetadatafield>
169             </qtimetadata>
170             <qmd_levelofdifficulty>basic</qmd_levelofdifficulty>
171             <qmd_topic>qtiv1p2test</qmd_topic>
172          </itemmetadata>
```

```
173          …
174          <resprocessing>
175             <outcomes>
176                <decvar defaultval = "0" minvalue = "0" maxvalue = "1"/>
177             </outcomes>
178             …
179          </resprocessing>
180          …
181       </item>
182       <item ident = "IMS_QTIV1p2_I_OUT_09" maxattempts = "2">
183          <itemmetadata>
184             <qtimetadata>
185                <qtimetadatafield>
186                   <fieldlabel>qmd_weighting</fieldlabel>
187                   <fieldentry>1</fieldentry>
188                </qtimetadatafield>
189             </qtimetadata>
190             <qmd_levelofdifficulty>advanced</qmd_levelofdifficulty>
191             <qmd_topic>qtiv1p2test</qmd_topic>
192          </itemmetadata>
193          …
194          <resprocessing>
195             <outcomes>
196                <decvar defaultval="0" minvalue="0" maxvalue="1"/>
197             </outcomes>
198             …
199          </resprocessing>
200          …
201       </item>
202       <item ident = "IMS_QTIV1p2_I_OUT_10" maxattempts = "2">
203          <itemmetadata>
204             <qtimetadata>
205                <qtimetadatafield>
206                   <fieldlabel>qmd_weighting</fieldlabel>
207                   <fieldentry>1</fieldentry>
208                </qtimetadatafield>
209             </qtimetadata>
210             <qmd_levelofdifficulty>advanced</qmd_levelofdifficulty>
211             <qmd_topic>qtiv1p2test</qmd_topic>
212          </itemmetadata>
213          …
214          <resprocessing>
215             <outcomes>
216                <decvar defaultval="0" minvalue="0" maxvalue="1"/>
217             </outcomes>
218             …
219          </resprocessing>
220          …
221       </item>
222    </section>
223 </questestinterop>
```

The ‘SumOfScores’ scoring algorithm that is supported for the Section is:

```
SCORE = 0
SCORE.min = 0
SCORE.max = 0
SCORE.normalized = 0.0
For ALL_PRESENTED_ITEMSand ITEM_HAS_VARIABLE(SCORE)
                    and
```

```
      (item.metadata.difficulty = basic or item.metadata.difficulty = advanced)
   do
      begin
         SCORE.min = SCORE.min + item.SCORE.minvalue
         SCORE.max = SCORE.max + item.SCORE.maxvalue
         SCORE = SCORE + item.SCORE
      Enddo
   SCORE.normalized = (SCORE – SCORE.min) / (SCORE.max – SCORE.min)
```

In the case where all ten Items have been attempted and seven have been answered correctly then the final set of Section variable states will be (the meta-data selections ensure that all Items are selected to contribute to the score aggregation):

SCORE = 7

SCORE.min = 0

SCORE.max = 10

SCORE.normalized = 0.7

This example is available in the file: ‘ims_qtiasiv1p2/outcome/basicoutexample09.xml’.

4.4 Advanced Example XML Instances

The following set of advanced examples demonstrates how the different use-cases are supported.

4.4.1 Example (Multiple-choice Quiz)

Multiple-choice quiz – this is a simple “end of chapter” quiz given by an LMS. The scores are reported back to the student for self-evaluation, so there is no particular need for high reliability. All of the questions are discrete (single response) questions whose outcomes are a dichotomous variable e.g. multiple-choice questions. For the sake of being definite we assume there are 5 Items and that we have a defined outcome, “CORRECT” for each Item. This takes on the value “True” if the response was correct and “False’ if the response was not correct. All items are given equal weight. We wish to know, (1) the number of correct items, (2) the percentage of total score obtained by a candidate and (3) the percentage of items actually answered which were correct.

```
1   <questestinterop>
2      <section title = "IMSQTI_Outcomes_Processing_S01" ident = "IMS_QTIV1p2_S_OUTADV_01">
3         <outcomes_processing scoremodel = "NumberCorrect">
4            <outcomes>
5               <decvar defaultval = "0" varname = "COUNT" vartype = "Integer"
6                       minvalue = "0" maxvalue = "10" cutvalue = "5"/>
7            </outcomes>
8            <outcomes_feedback_test title = "Section Mastery Test">
9               <test_variable>
10                 <variable_test testoperator = "GTE" varname = "COUNT">5</variable_test>
11              </test_variable>
12              <displayfeedback linkrefid = "SectionMastery" feedbacktype = "Response"/>
13           </outcomes_feedback_test>
14           <outcomes_feedback_test title = "Section Fail Test">
15              <test_variable>
16                 <variable_test testoperator = "LT" varname = "COUNT">5</variable_test>
17              </test_variable>
18              <displayfeedback linkrefid = "SectionFail" feedbacktype = "Response"/>
19           </outcomes_feedback_test>
20        </outcomes_processing>
21        <outcomes_processing scoremodel = "SumofScores">
22           <outcomes>
```

```
23              <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
24                     minvalue = "0" maxvalue = "10" cutvalue = "5"/>
25           </outcomes>
26           <outcomes_feedback_test title = "Section Mastery Test">
27              <test_variable>
28                 <variable_test testoperator = "GTE" varname = "SCORE">5</variable_test>
29              </test_variable>
30              <displayfeedback linkrefid = "SectionMastery" feedbacktype = "Response"/>
31           </outcomes_feedback_test>
32           <outcomes_feedback_test title = "Section Fail Test">
33              <test_variable>
34                 <variable_test testoperator = "LT" varname = "SCORE">5</variable_test>
35              </test_variable>
36              <displayfeedback linkrefid = "SectionFail" feedbacktype = "Response"/>
37           </outcomes_feedback_test>
38        </outcomes_processing>
39        <outcomes_processing scoremodel = "NumberCorrectAttempted">
40           <outcomes>
41              <decvar defaultval = "0" varname = "COUNT" vartype = "Integer"
42                     minvalue = "0" maxvalue = "10" cutvalue = "5"/>
43           </outcomes>
44           <map_output varname="COUNT">COUNT_Attempted</map_output>
45           <outcomes_feedback_test title = "Section Mastery Test">
46              <test_variable>
47                 <variable_test testoperator = "GTE" varname = "COUNT">5</variable_test>
48              </test_variable>
49              <displayfeedback linkrefid = "SectionMastery" feedbacktype = "Response"/>
50           </outcomes_feedback_test>
51           <outcomes_feedback_test title = "Section Fail Test">
52              <test_variable>
53                 <variable_test testoperator = "LT" varname = "COUNT">5</variable_test>
54              </test_variable>
55              <displayfeedback linkrefid = "SectionFail" feedbacktype = "Response"/>
56           </outcomes_feedback_test>
57        </outcomes_processing>
58        <sectionfeedback ident = "SectionMastery" view = "All">
59           <material>
60              <mattext>Well done.  You have passed this Topic.</mattext>
61           </material>
62        </sectionfeedback>
63        <sectionfeedback ident = "SectionFail" view = "All">
64           <material>
65              <mattext>Sorry.  You have failed this topic.</mattext>
66           </material>
67        </sectionfeedback>
68        <selection_ordering>
69           <selection/>
70           <order order_type = "Sequential"/>
71        </selection_ordering>
72        <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
73           <presentation>
74              <flow>
75                 <material>
76                    <mattext>Which one of the following statements is true about the QTI
77                            specification ?</mattext>
78                 </material>
79                 <response_lid ident = "LID01" rcardinality = "Single" rtiming = "No">
80                    <render_choice shuffle = "Yes" minnumber = "1" maxnumber = "1">
81                       <flow_label>
82                          <response_label ident = "A">
```

```
83                            <material>
84                               <mattext>It is independent of pedagogy</mattext>
85                            </material>
86                         </response_label>
87                      </flow_label>
88                      <flow_label>
89                         <response_label ident = "B">
90                            <material>
91                               <mattext>It cannot be extended to support proprietary
92                                  features</mattext>
93                            </material>
94                         </response_label>
95                      </flow_label>
96                      <flow_label>
97                         <response_label ident = "C">
98                            <material>
99                               <mattext>It contains many unnecesary features
100                              </mattext>
101                           </material>
102                        </response_label>
103                     </flow_label>
104                     <flow_label>
105                        <response_label ident = "D">
106                           <material>
107                              <mattext>It provides only one view for different
108                                 types of participants</mattext>
109                           </material>
110                        </response_label>
11                      </flow_label>
112                  </render_choice>
113               </response_lid>
114            </flow>
115         </presentation>
116         <resprocessing>
117            <outcomes>
118               <decvar defaultval = "False" vartype = "Boolean" varname = "CORRECT"/>
119               <decvar defaultval = "0" vartype = "Integer" minvalue = "0"
120                  maxvalue = "1" varname = "SCORE"/>
121            </outcomes>
123            <respcondition continue = "No">
124               <qticomment>To detect the correct answer.</qticomment>
125               <conditionvar>
126                  <varequal respident = "LID01">A</varequal>
127               </conditionvar>
128               <setvar action = "Set" varname = "CORRECT">True</setvar>
129               <setvar action = "Set" varname = "SCORE">1</setvar>
130               <displayfeedback feedbacktype = "Response" linkrefid = "Correct"/>
131            </respcondition>
132            <respcondition continue = "No">
133               <qticomment>To detect if the incorrect answer is supplied.</qticomment>
134               <conditionvar>
135                  <not>
136                     <varequal respident = "LID01" case = "No">A</varequal>
137                  </not>
138               </conditionvar>
139               <displayfeedback feedbacktype = "Response" linkrefid = "InCorrect"/>
140            </respcondition>
141         </resprocessing>
142         <itemfeedback ident = "Correct">
```

```
143             <material>
144                <mattext>Correct answer is A.</mattext>
145             </material>
146          </itemfeedback>
147          <itemfeedback ident = "Incorrect" view = "All">
148             <material>
149                <mattext xml:lang="en-US" xml:space="default" texttype= "" charset= "">
150                   Answers B, C and D are incorrect.
151                </mattext>
152             </material>
153          </itemfeedback>
154       </item>
155       <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
156          <presentation>
157             <flow>
158                <material>
159                   <mattext>The QTI specification supports the exchange of which data
160                      structure ?</mattext>
161                </material>
162                <response_lid ident = "LID02" rcardinality = "Single" rtiming = "No">
163                   <render_choice shuffle = "Yes" minnumber = "1" maxnumber = "1">
164                      <flow_label>
165                         <response_label ident = "A">
166                            <material>
167                               <mattext>Response</mattext>
168                            </material>
169                         </response_label>
170                      </flow_label>
171                      <flow_label>
172                         <response_label ident = "B">
173                            <material>
174                               <mattext>Question</mattext>
175                            </material>
176                         </response_label>
177                      </flow_label>
178                      <flow_label>
179                         <response_label ident = "C">
180                            <material>
181                               <mattext>Item</mattext>
182                            </material>
183                         </response_label>
184                      </flow_label>
185                      <flow_label>
186                         <response_label ident = "D">
187                            <material>
188                               <mattext>Test</mattext>
189                            </material>
190                         </response_label>
191                      </flow_label>
192                   </render_choice>
193                </response_lid>
194             </flow>
195          </presentation>
196          <resprocessing>
197             <outcomes>
198                <decvar varname = "CORRECT" vartype = "Boolean" defaultval = "False"/>
199                <decvar defaultval = "0" vartype = "Integer" minvalue = "0"
200                   maxvalue = "1" varname = "SCORE"/>
201             </outcomes>
202             <respcondition continue = "No">
```

```
203                <qticomment>Correct response detection</qticomment>
204                <conditionvar>
205                   <varequal respident = "LID02">C</varequal>
206                </conditionvar>
207                <setvar varname = "CORRECT" action = "Set">True</setvar>
208                <setvar action = "Set" varname = "SCORE">1</setvar>
209                <displayfeedback linkrefid = "Correct" feedbacktype = "Response"/>
210             </respcondition>
211             <respcondition continue = "No">
212                <qticomment>Processing for the incorrect answer.</qticomment>
213                <conditionvar>
214                   <not>
215                      <varequal respident = "LID02">C</varequal>
216                   </not>
217                </conditionvar>
281                <displayfeedback linkrefid = "Incorrect" feedbacktype = "Response"/>
219             </respcondition>
220          </resprocessing>
221          <itemfeedback ident = "Correct" view = "All">
222             <material>
223                <mattext>Correct.  Items.</mattext>
224             </material>
225          </itemfeedback>
226          <itemfeedback ident = "Incorrect" view = "All">
227             <material>
228                <mattext>No.  The correct answer is Item.</mattext>
229             </material>
230          </itemfeedback>
231       </item>
232       <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
233          <presentation>
234             <flow>
235                <material>
236                   <mattext>What is the maximum number of Assessments can be contained
237                      in a single QTI ASI XML instance?</mattext>
238                </material>
239                <response_lid ident = "TF01" rcardinality = "Single" rtiming = "No">
240                   <render_choice shuffle = "Yes">
241                      <flow_label>
242                         <response_label ident = "A">
243                            <material>
244                               <mattext>0</mattext>
245                            </material>
246                         </response_label>
247                      </flow_label>
248                      <flow_label>
249                         <response_label ident = "B">
250                            <material>
251                               <mattext>1</mattext>
252                            </material>
253                         </response_label>
254                      </flow_label>
255                      <flow_label>
256                         <response_label ident = "C">
257                            <material>
258                               <mattext>More than one</mattext>
259                            </material>
260                         </response_label>
261                      </flow_label>
262                   </render_choice>
```

```
263                </response_lid>
264             </flow>
265          </presentation>
266          <resprocessing>
267             <outcomes>
268                <decvar defaultval = "False" vartype = "Boolean" varname = "CORRECT"/>
269                <decvar defaultval = "0" vartype = "Integer" minvalue = "0"
270                   maxvalue = "1" varname = "SCORE"/>
271             </outcomes>
272             <respcondition continue = "No">
273                <qticomment>Processing for the incorrect answer.</qticomment>
274                <conditionvar>
275                   <not>
276                      <varequal respident = "TF01">B</varequal>
277                   </not>
278                </conditionvar>
279                <displayfeedback feedbacktype = "Response" linkrefid = "Incorrect"/>
280             </respcondition>
281             <respcondition continue = "No">
282                <qticomment>Processing for the correct answer.</qticomment>
283                <conditionvar>
284                   <varequal respident = "TF01" case = "No">B</varequal>
285                </conditionvar>
286                <setvar action = "Set" varname = "CORRECT">True</setvar>
287                <setvar action = "Set" varname = "SCORE">1</setvar>
288                <displayfeedback linkrefid = "Correct" feedbacktype = "Response"/>
289             </respcondition>
290          </resprocessing>
291          <itemfeedback ident = "Correct">
292             <material>
293                <mattext>Correct answer.</mattext>
294             </material>
295          </itemfeedback>
296          <itemfeedback ident = "Incorrect">
297             <material>
298                <mattext>Sorry, you are wrong.  A QTI-XML instance CAN contain,
299                   at most, one Asssessment.</mattext>
300             </material>
301          </itemfeedback>
302       </item>
303       <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "1">
304          <presentation>
305             <flow>
306                <material>
307                   <mattext>Which one of the following data structures is used to
308                      provide hierarchically structured assessments ?</mattext>
309                </material>
310                <response_lid ident = "LID03" rcardinality = "Single" rtiming = "No">
311                   <render_choice shuffle = "Yes">
312                      <flow_label>
313                         <response_label ident = "A">
314                            <material>
315                               <mattext>Assessment</mattext>
316                            </material>
317                         </response_label>
318                      </flow_label>
319                      <flow_label>
320                         <response_label ident = "B">
321                            <material>
322                               <mattext>Section</mattext>
```

```
                           </material>
                        </response_label>
                     </flow_label>
                     <flow_label>
                        <response_label ident = "C">
                           <material>
                              <mattext>Item</mattext>
                           </material>
                        </response_label>
                     </flow_label>
                     <flow_label>
                        <response_label ident = "D">
                           <material>
                              <mattext>Question</mattext>
                           </material>
                        </response_label>
                     </flow_label>
                  </render_choice>
               </response_lid>
            </flow>
         </presentation>
         <resprocessing>
            <outcomes>
               <decvar varname = "CORRECT" vartype = "Boolean" defaultval = "False"/>
               <decvar defaultval = "0" vartype = "Integer" minvalue = "0"
                  maxvalue = "1" varname = "SCORE"/>
            </outcomes>
            <respcondition continue = "No">
               <qticomment>Correct response check</qticomment>
               <conditionvar>
                  <varequal respident = "LID03" case = "No">B</varequal>
               </conditionvar>
               <setvar action = "Set" varname = "CORRECT">True</setvar>
               <setvar action = "Set" varname = "SCORE">1</setvar>
               <displayfeedback linkrefid = "Correct" feedbacktype = "Response"/>
            </respcondition>
            <respcondition>
               <conditionvar>
                  <not>
                     <varequal respident = "LID03" case = "No">B</varequal>
                  </not>
               </conditionvar>
               <displayfeedback linkrefid = "Incorrect" feedbacktype = "Response"/>
            </respcondition>
         </resprocessing>
         <itemfeedback ident = "Correct" view = "All">
            <material>
               <mattext>Correct.</mattext>
            </material>
         </itemfeedback>
         <itemfeedback ident = "Incorrect" view = "All">
            <material>
               <mattext>No.  Only Sections support hierachical Assessments.</mattext>
            </material>
         </itemfeedback>
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "1">
         <presentation>
            <flow>
               <material>
```

```
383                     <mattext>Which question types are supported by the QTILite
384                        specification ?</mattext>
385                  </material>
386                  <response_lid ident = "LID03" rcardinality = "Multiple" rtiming = "No">
387                     <render_choice shuffle = "Yes" minnumber = "1" maxnumber = "4">
388                        <flow_label>
389                           <response_label ident = "A">
390                              <material>
391                                 <mattext>Multiple choice</mattext>
392                              </material>
393                           </response_label>
394                        </flow_label>
395                        <flow_label>
396                           <response_label ident = "B">
397                              <material>
398                                 <mattext>Multiple response</mattext>
399                              </material>
400                           </response_label>
401                        </flow_label>
403                        <flow_label>
403                           <response_label ident = "C">
404                              <material>
405                                 <mattext>Fill-in-the-blank</mattext>
406                              </material>
407                           </response_label>
408                        </flow_label>
409                        <flow_label>
410                           <response_label ident = "D">
411                              <material>
412                                 <mattext>Drag-and-drop</mattext>
313                              </material>
414                           </response_label>
415                        </flow_label>
416                     </render_choice>
417                  </response_lid>
418               </flow>
419            </presentation>
420            <resprocessing>
421               <outcomes>
422                  <decvar varname = "CORRECT" vartype = "Boolean" defaultval = "False"/>
423                  <decvar defaultval = "0" vartype = "Integer" minvalue = "0"
424                     maxvalue = "1" varname = "SCORE"/>
425               </outcomes>
426               <respcondition>
427                  <qticomment>Correct answer response processing.</qticomment>
428                  <conditionvar>
429                     <varequal respident = "LID03">A</varequal>
430                  </conditionvar>
431                  <setvar varname = "CORRECT" action = "Set">True</setvar>
432                  <setvar action = "Set" varname = "SCORE">1</setvar>
433                  <displayfeedback linkrefid = "Correct" feedbacktype = "Response"/>
434               </respcondition>
435               <respcondition>
436                  <qticomment>Incorrect answer response processing.</qticomment>
437                  <conditionvar>
438                     <not>
439                        <varequal respident = "LID03">A</varequal>
440                     </not>
441                  </conditionvar>
442                  <displayfeedback linkrefid = "Incorrect" feedbacktype = "Response"/>
```

```
443             </respcondition>
444          </resprocessing>
445          <itemfeedback ident = "Correct" view = "All">
446             <material>
447                <mattext>Correct.  Multple-choice only.</mattext>
448             </material>
448          </itemfeedback>
459          <itemfeedback ident = "Incorrect" view = "All">
460             <material>
461                <mattext>No.  Only Multiple-choice questions are supported.</mattext>
462             </material>
463          </itemfeedback>
464       </item>
465    </section>
466 </questestinterop>
```

The actual percentage total score and percentage of items actually answered will be derived values from the SCORE.normalized (SumofScores) and COUNT_Attempted.normalized (NumberCorrectAttempted) variables. If it is assumed that the candidate attempts the first four questions and gets the first three of those correct, then the set of variables are:

SCORE = 3

SCORE.min = 0

SCORE.max = 5

SCORE.normalized = 0.6

PercentageTotalScore = 60%

COUNT = 3

COUNT.min = 0

COUNT.max = 5

COUNT.normalized = 0.6

COUNT_Attempted = 3

COUNT_Attempted.min = 0

COUNT_Attempted.max = 4

COUNT_Attempted.normalized = 0.75

PercentageCorrectAttempted = 75%

This example is given as ‘ims_qtiasiv1p2/outcome/advoutexample01.xml’.

4.4.2 Example (True/false Quiz)

True/false quiz – this is a simple “end of chapter” quiz given by an LMS. The scores are reported back to the student for self-evaluation, so there is no particular need for high reliability. All of the questions are discrete true/false. For the sake of being definite we assume there are 10 Items and that we have a defined outcome, “isCorrect” for each Item. This takes on the value ‘1’ if the response was correct and “-1’ if the response was not correct. All items are given equal weight. We wish to know, (1) the number of correct items, (2) the total score and (3) the weighted total score.

```
<questestinterop>
   <section title = "IMSQTI_Outcomes_Processing_S02" ident = "IMS_QTIV1p2_S_OUTADV_02">
      <outcomes_processing scoremodel = "NumberCorrect">
         <outcomes>
            <decvar defaultval = "0" varname = "COUNT" vartype = "Integer"
               minvalue = "0" maxvalue = "10" cutvalue = "5"/>
         </outcomes>
         <outcomes_feedback_test title = "Section Mastery Test">
            <test_variable>
               <variable_test testoperator = "GTE" varname = "COUNT">5</variable_test>
            </test_variable>
            <displayfeedback linkrefid = "SectionMastery" feedbacktype = "Response"/>
         </outcomes_feedback_test>
         <outcomes_feedback_test title = "Section Fail Test">
            <test_variable>
               <variable_test testoperator = "LT" varname = "COUNT">5</variable_test>
            </test_variable>
            <displayfeedback linkrefid = "SectionFail" feedbacktype = "Response"/>
         </outcomes_feedback_test>
      </outcomes_processing>
      <outcomes_processing scoremodel = "SumofScores">
         <outcomes>
            <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
               minvalue = "-10" maxvalue = "10" cutvalue = "0"/>
         </outcomes>
         <outcomes_feedback_test title = "Section Mastery Test">
            <test_variable>
               <variable_test testoperator = "GTE" varname = "COUNT">0</variable_test>
            </test_variable>
            <displayfeedback linkrefid = "SectionMastery" feedbacktype = "Response"/>
         </outcomes_feedback_test>
         <outcomes_feedback_test title = "Section Fail Test">
            <test_variable>
               <variable_test testoperator = "LT" varname = "COUNT">0</variable_test>
            </test_variable>
            <displayfeedback linkrefid = "SectionFail" feedbacktype = "Response"/>
         </outcomes_feedback_test>
      </outcomes_processing>
      <outcomes_processing scoremodel = "ParameterSumofScoresWeighted">
         <outcomes>
            <decvar defaultval = "0" varname = "SCORE" vartype = "Integer"
               minvalue = "-20" maxvalue = "20" cutvalue = "0"/>
         </outcomes>
         <objects_condition>
            <objects_parameter pname="qmd_weighting">2</objects_parameter>
         </objects_condition>
         <map_output varname="SCORE">SCORE_PWSOS</map_output>
         <outcomes_feedback_test title = "Section Mastery Test">
            <test_variable>
               <variable_test testoperator = "GTE" varname = "COUNT">0</variable_test>
            </test_variable>
            <displayfeedback linkrefid = "SectionMastery" feedbacktype = "Response"/>
         </outcomes_feedback_test>
         <outcomes_feedback_test title = "Section Fail Test">
            <test_variable>
               <variable_test testoperator = "LT" varname = "COUNT">0</variable_test>
            </test_variable>
            <displayfeedback linkrefid = "SectionFail" feedbacktype = "Response"/>
         </outcomes_feedback_test>
      </outcomes_processing>
```

```
61        <sectionfeedback ident = "SectionMastery" view = "All">
62           <material>
63              <mattext>Well done.  You have passed this Topic.</mattext>
64           </material>
65        </sectionfeedback>
66        <sectionfeedback ident = "SectionFail" view = "All">
67           <material>
68              <mattext>Sorry.  You have failed this topic.</mattext>
69           </material>
70        </sectionfeedback>
71        <selection_ordering>
72           <selection/>
73           <order order_type = "Sequential"/>
74        </selection_ordering>
75        <item ident = "IMS_QTIV1p2_I_OUT_01" maxattempts = "2">
76           <presentation>
77              <flow>
78                 <material>
79                    <mattext>Multiple Assessments cannot be contained in a single QTI
80                       ASI XML instance.
81                    </mattext>
82                 </material>
83                 <response_lid ident = "TF01" rcardinality = "Single" rtiming = "No">
84                    <render_choice shuffle = "Yes">
85                       <flow_label>
86                          <response_label ident = "T">
87                             <material>
88                                <mattext>True</mattext>
89                             </material>
90                          </response_label>
91                       </flow_label>
92                       <flow_label>
93                          <response_label ident = "F">
94                             <material>
95                                <mattext>False</mattext>
96                             </material>
97                          </response_label>
98                       </flow_label>
99                    </render_choice>
100                </response_lid>
101             </flow>
102          </presentation>
103          <resprocessing>
104             <qticomment>Processing for the incorrect answer.</qticomment>
105             <outcomes>
106                <decvar defaultval = "False" vartype = "Boolean" varname = "CORRECT"/>
107                <decvar defaultval = "-1" vartype = "Integer"
108                   minvalue="-1" maxvalue="1" varname = "SCORE"/>
109             </outcomes>
110             <respcondition continue = "No">
11                 <conditionvar>
112                   <varequal respident = "TF01">T</varequal>
113                </conditionvar>
114                <displayfeedback feedbacktype = "Response" linkrefid = "Incorrect"/>
115             </respcondition>
116             <respcondition continue = "No">
117                <conditionvar>
118                   <varequal respident = "TF01" case = "No">F</varequal>
119                </conditionvar>
120                <setvar action = "Set" varname = "CORRECT">True</setvar>
```

```
121                <setvar action = "Set" varname = "SCORE">1</setvar>
123                <displayfeedback linkrefid = "Correct" feedbacktype = "Response"/>
124             </respcondition>
125          </resprocessing>
126          <itemfeedback ident = "Correct">
127             <material>
128                <mattext>Correct answer.</mattext>
129             </material>
130          </itemfeedback>
131          <itemfeedback ident = "Incorrect">
132             <material>
133                <mattext>Sorry, you are wrong.  A QTI-XML instance CAN contain multiple
134                   Assessments.
135                </mattext>
136             </material>
137          </itemfeedback>
138       </item>
139       <item ident = "IMS_QTIV1p2_I_OUT_02" maxattempts = "2">
140          <presentation>
141             <flow>
142                <material>
143                   <mattext>Time-based responses are supported by the QTI
144                      specification.
145                   </mattext>
146                </material>
147                <response_lid ident = "TF02" rcardinality = "Single" rtiming = "No">
148                   <render_choice shuffle = "Yes">
149                      <flow_label>
150                         <response_label ident = "T">
151                            <material>
152                               <mattext>True</mattext>
153                            </material>
154                         </response_label>
155                      </flow_label>
156                      <flow_label>
157                         <response_label ident = "F">
158                            <material>
159                               <mattext>False</mattext>
160                            </material>
161                         </response_label>
162                      </flow_label>
163                   </render_choice>
164                </response_lid>
165             </flow>
166          </presentation>
167          <resprocessing>
168             <qticomment>Processing for the correct answer.</qticomment>
169             <outcomes>
170                <decvar defaultval = "False" vartype = "Boolean" varname = "CORRECT"/>
171             <decvar defaultval = "-1" vartype = "Integer" minvalue="-1"
172                maxvalue="1" varname = "SCORE"/>
173             </outcomes>
174             <respcondition>
175                <conditionvar>
176                   <varequal respident = "TF02">T</varequal>
177                </conditionvar>
178                <setvar action = "Set" varname = "CORRECT">True</setvar>
179                <setvar action = "Set" varname = "SCORE">1</setvar>
180                <displayfeedback feedbacktype = "Response" linkrefid = "Correct"/>
```

```
            </respcondition>
            <respcondition continue = "No">
               <conditionvar>
                  <varequal respident = "TF02" case = "No">F</varequal>
               </conditionvar>
               <displayfeedback linkrefid = "Incorrect" feedbacktype = "Response"/>
            </respcondition>
         </resprocessing>
         <itemfeedback ident = "Correct">
            <material>
               <mattext>Correct answer.</mattext>
            </material>
         </itemfeedback>
         <itemfeedback ident = "Incorrect">
            <material>
               <mattext>Sorry, you are wrong.  Time-based reponses ARE supported by
                  the specification.
               </mattext>
            </material>
         </itemfeedback>
      </item>
      <item ident = "IMS_QTIV1p2_I_OUT_03" maxattempts = "2">
         <presentation>
            <flow>
               <material>
                  <mattext>Items defined according to QTI V1.0 are not compatible with
                     Items from QTI V1.2.
                  </mattext>
               </material>
               <response_lid ident = "TF03" rcardinality = "Single" rtiming = "No">
                  <render_choice shuffle = "Yes">
                     <flow_label>
                        <response_label ident = "T">
                           <material>
                              <mattext>True</mattext>
                           </material>
                        </response_label>
                     </flow_label>
                     <flow_label>
                        <response_label ident = "F">
                           <material>
                              <mattext>False</mattext>
                           </material>
                        </response_label>
                     </flow_label>
                  </render_choice>
               </response_lid>
            </flow>
         </presentation>
         <resprocessing>
            <qticomment>Processing for the correct answer.</qticomment>
            <outcomes>
               <decvar defaultval = "True" vartype = "Boolean" varname = "CORRECT"/>
               <decvar defaultval = "-1" vartype = "Integer"
                  minvalue="-1" maxvalue="1" varname = "SCORE"/>
            </outcomes>
            <respcondition>
               <conditionvar>
                  <varequal respident = "TF03">F</varequal>
               </conditionvar>
```

```
241                <setvar action = "Set" varname = "CORRECT">True</setvar>
242                <setvar action = "Set" varname = "SCORE">1</setvar>
243                <displayfeedback feedbacktype = "Response" linkrefid = "Correct"/>
244             </respcondition>
245             <respcondition continue = "No">
246                <conditionvar>
247                   <varequal respident = "TF03" case = "No">T</varequal>
248                </conditionvar>
249                <displayfeedback linkrefid = "Incorrect" feedbacktype = "Response"/>
250             </respcondition>
251          </resprocessing>
252          <itemfeedback ident = "Correct">
253             <material>
254                <mattext>Correct answer.</mattext>
255             </material>
256          </itemfeedback>
257          <itemfeedback ident = "Incorrect">
258             <material>
259                <mattext>Sorry, you are wrong.  V1.0 Items ARE compatible with the V1.2
260                   specification.
261                </mattext>
262             </material>
263          </itemfeedback>
264       </item>
265       <item ident = "IMS_QTIV1p2_I_OUT_04" maxattempts = "2">
266          <presentation>
267             <flow>
268                <material>
269                   <mattext>The QTI specification is incompatible with the IMS Content
270                      Packaging specification.
271                   </mattext>
272                </material>
273                <response_lid ident = "TF04" rcardinality = "Single" rtiming = "No">
274                   <render_choice shuffle = "Yes">
275                      <flow_label>
276                         <response_label ident = "T">
277                            <material>
278                               <mattext>True</mattext>
279                            </material>
280                         </response_label>
281                      </flow_label>
282                      <flow_label>
283                         <response_label ident = "F">
284                            <material>
285                               <mattext>False</mattext>
286                            </material>
287                         </response_label>
288                      </flow_label>
289                   </render_choice>
290                </response_lid>
291             </flow>
292          </presentation>
293          <resprocessing>
294             <qticomment>Processing for the correct answer.</qticomment>
295             <outcomes>
296                <decvar defaultval = "False" vartype = "Boolean" varname = "CORRECT"/>
297                <decvar defaultval = "-1" vartype = "Integer"
298                   minvalue="-1" maxvalue="1" varname = "SCORE"/>
299             </outcomes>
300             <respcondition>
```

```
301                <conditionvar>
302                   <varequal respident = "TF04">F</varequal>
303                </conditionvar>
304                <setvar action = "Set" varname = "CORRECT">True</setvar>
305                <setvar action = "Set" varname = "SCORE">1</setvar>
306                <displayfeedback feedbacktype = "Response" linkrefid = "Correct"/>
307             </respcondition>
308             <respcondition continue = "No">
309                <conditionvar>
310                   <varequal respident = "TF04" case = "No">T</varequal>
311                </conditionvar>
312                <displayfeedback linkrefid = "Incorrect" feedbacktype = "Response"/>
313             </respcondition>
314          </resprocessing>
315          <itemfeedback ident = "Correct">
316             <material>
317                <mattext>Correct answer.</mattext>
318             </material>
319          </itemfeedback>
320          <itemfeedback ident = "Incorrect">
321             <material>
322                <mattext>Sorry, you are wrong.  The IMS QTI and Content Packaging
323                   specifications are compatible.
324                </mattext>
325             </material>
326          </itemfeedback>
327       </item>
328       <item ident = "IMS_QTIV1p2_I_OUT_05" maxattempts = "2">
329          <presentation>
330             <flow class = "Block">
331                <material>
332                   <mattext texttype = "text/plain" charset = "ascii-us">The QTI
333                      specification is used within the ADL SCORM.
334                   </mattext>
335                </material>
336                <response_lid ident = "TF05" rcardinality = "Single" rtiming = "No">
337                   <render_choice shuffle = "Yes">
338                      <flow_label class = "Block">
339                         <response_label ident = "T">
340                            <material>
341                               <mattext>True</mattext>
342                            </material>
343                         </response_label>
344                      </flow_label>
345                      <flow_label class = "Block">
346                         <response_label ident = "F">
347                            <material>
348                               <mattext>False</mattext>
349                            </material>
350                         </response_label>
351                      </flow_label>
352                   </render_choice>
353                </response_lid>
354             </flow>
355          </presentation>
356          <resprocessing>
357             <qticomment>Processing for the correct answer.</qticomment>
358             <outcomes>
359                <decvar defaultval = "False" vartype = "Boolean" varname = "CORRECT"/>
360                <decvar defaultval = "-1" vartype = "Integer"
```

```
                  minvalue="-1" maxvalue="1" varname = "SCORE"/>
            </outcomes>
            <respcondition continue = "No">
               <conditionvar>
                  <varequal respident = "TF05" case = "No">F</varequal>
               </conditionvar>
               <setvar action = "Set" varname = "CORRECT">True</setvar>
               <setvar action = "Set" varname = "SCORE">1</setvar>
               <displayfeedback feedbacktype = "Response" linkrefid = "Correct"/>
            </respcondition>
            <respcondition continue = "No">
               <conditionvar>
                  <varequal respident = "TF05" case = "No">T</varequal>
               </conditionvar>
               <displayfeedback linkrefid = "Incorrect" feedbacktype = "Response"/>
            </respcondition>
         </resprocessing>
         <itemfeedback ident = "Correct" view = "All">
            <material>
               <mattext>Correct answer.</mattext>
            </material>
         </itemfeedback>
         <itemfeedback ident = "Incorrect" view = "All">
            <material>
               <mattext>Sorry, you are wrong.  The QTI specification is NOT used
                  within ADL SCORM.
               </mattext>
            </material>
         </itemfeedback>
      </item>
   </section>
</questestinterop>
```

If it is assumed that the candidate attempts the first four questions and gets the first three of those correct, then the set of variables are:

SCORE = 1

SCORE.min = -5

SCORE.max = 5

SCORE.normalized = 0.6

COUNT = 3

COUNT.min = 0

COUNT.max = 5

COUNT.normalized = 0.6

SCORE_Weighted = 2

SCORE_Weighted.min = -10

SCORE_Weighted.max = 10

SCORE_Weighted.normalized = 0.6

This example is given as ‘ims_qtiasiv1p2/outcome/advoutexample01.xml’.

4.5 Implementation Guidance

4.5.1 The In-built Scoring Algorithms

Whenever possible the in-built scoring algorithms should be used to ensure interoperability between systems. It should be noted that it is possible to use multiple scoring algorithms on the same set of evaluation objects – this enables multivariate scoring mechanisms. At the current time the in-built scoring algorithms are:

- Number Correct – see Appendix B1;
- Number Correct Attempted – see Appendix B2;
- Weighted Number Correct – see Appendix B3;
- Weighted Number Correct Attempted – see Appendix B4;
- Parameter Weighted Number Correct – see Appendix B5;
- Parameter Weighted Number Correct Attempted – see Appendix B6;
- Sum of Scores – see Appendix B7;
- Sum of Scores Attempted – see Appendix B8;
- Weighted Sum of Scores – see Appendix B9;
- Weighted Sum of Scores Attempted – see Appendix B10;
- Parameter Weighted Sum of Scores – see Appendix B11;
- Parameter Weighted Sum of Scores Attempted – see Appendix B12;
- Best K of N – see Appendix B13;
- Guessing Penalty – see Appendix B14;
- Weighted Guessing Penalty – see Appendix B15;

4.5.2 Defining Proprietary Scoring Algorithms

It is understood that not all of the required scoring algorithms will be supported intrinsically. Proprietary algorithms can be supported by:

- Assigning an appropriate name to the new scoring algorithm and identifying this algorithm in the ‘scoremodel’ attribute for the <outcome_processing> element. Scoring engines that do not recognize the named scoring algorithm should indicate an error stating that it will not be possible to produce the associated set of aggregated scores;
- Deciding how the default or specially named will be handled by the new scoring algorithm;
- Assigning particular parameter names and passing these to the scoring algorithm using the <objects_parameter> and <processing_parameter> elements.

In some cases it will be necessary to coerce a numeric score into a count of whether or not the object has been answered correctly. The conversion from a SCORE to a CORRECT variable will be achieved using the cutscore value for the variable. The algorithm now becomes:

```
if (object.SCORE >= object.SCORE.cutscore)
then object.CORRECT = True
else object.CORRECT = False
```

4.5.3 Using Meta-data

The meta-data of the objects is an important aggregation object selection mechanism. This selection must operate on the IMS QTI-specific meta-data and the IMS Meta-data definitions. There are three meta-data entry techniques used within IMS QTI:

- For the explicitly identified IMS QTI-specific meta-data fields (the elements named using the 'qmd_**…***' scheme) the name of the meta-data element is used as the string name in the 'mdname' attribute of the <outcomes_metadata> element. The actual meta-data field value is that content of the <qmd_***…***> element. This mechanism is deprecated and should **not** be used;
- For the IMS QTI-specific meta-data fields named using the generic vocabulary mechanism (using the element <qtimetadatafield>) the string name will be that identified using the <qtimetadatafield><fieldlabel> structure. The actual meta-data field value is the content of the structure <qtimetadatafield><fieldentry>;
- For the IMS Meta-data entries the string naming convention will adopt the '.' mechanism to denote the concatenation of nested tags e.g. the string "metametadata.contribute.role.source" is used to denote the IMS Meta-data structure of '<metametadata><contribute><role><source>'. The 'lom:' namespace should be used at the front of the string.

4.5.4 Consideration of Selection & Ordering

It is important to realize that the actual score assigned can only be determined once the evaluation has been presented. Assessments and Sections can consist of a complex of child objects that are selected and ordered according to some set of rules defined within the <selection_ordering> element. The current outcomes processing mechanism is used to define the static rules that are used to create the aggregated scores that result from the selected, ordered, presented and attempted objects. Scores can only be assigned to objects that have been presented but some of the algorithms will provide normalized information that takes into account whether or not a presented object is attempted by the participant.

4.6 Example XML Instances

The full set of example files, as referred to in Sections 4.3 and 4.4 are available as part of the IMS QTI Resource Kit. The set of example XML instances are listed in Table 4.2.

Table 4.2 The QTI outcomes processing XML example files.

Filename	Nature	Description
basicoutexample01.xml	S(1)I(10)	The 'NumberCorrect' and 'NumberCorrectAttempted' in-built scoring algorithm using the standard input and output variable names.
basicoutexample02.xml	S(1)I(10)	The 'WeightedNumberCorrect', 'WeightedNumberCorrectAttempted', 'ParameterWeightedNumberCorrect' and 'ParameterWeightedNumberCorrectAttempted' in-built scoring algorithms using the standard input and output variable names.
basicoutexample03.xml	S(1)I(10)	The 'SumofScores' and 'SumofScoresAttempted' in-built scoring algorithms using the standard input and output variable names.
basicoutexample04.xml	S(1)I(10)	The 'WeightedSumofScores', 'WeightedSumofScoresAttempted', 'ParameterWeightedSumofScores' and 'ParameterWeightedSumofScoresAttempted' in-built scoring algorithm using the standard input and output variable names.
basicoutexample05.xml	S(1)I(10)	The 'BestKfromN' in-built scoring algorithm (selecting the best 7 scores) using the standard input and output variable names.
basicoutexample06.xml	S(1)I(10)	The 'GuessingPenalty' in-built scoring algorithm using the standard input and output variable names.

basicoutexample07.xml	S(1)I(10)	The 'SumofScores' in-built scoring algorithm using the standard input variable names but using remapped output variables.
basicoutexample08.xml	S(1)I(10)	The 'SumofScores' in-built scoring algorithm using remapped input variable names but with standard output variables.
basicoutexample09.xml	S(1)I(10)	The 'SumofScores' in-built scoring algorithm using the standard input and output variable names but selecting the objects according to their meta-data properties.
advoutexample01.xml	A(1)S(1)I(10)	Multiple-choice quiz with aggregated outcomes based upon the number correct, the percentage of total score obtained by a candidate and the percentage of items actually answered that were correct.
advoutexample02.xml	A(1)S(1)I(10)	True/false quiz with aggregated outcomes based upon the number correct, the total score and the weighted total score.

These files can be found in the directory: 'ims_qtiasiv1p2/outcome/'.

4.7 Proprietary Extensions

The proprietary extensions facilities listed in Table 4.3 are supported as elements within the specifications:

Table 4.3 List of proprietary extension elements.

Extension Element Name	Host Element	Description
objectscond_extension	objects_condition	Inclusion of alternative test mechanisms on the eligibility of an object to contribute to the scoring algorithm.

Note: *These elements are only used if the suppliers of the ASIs require proprietary features that are not supported by the available range of elements. It is recommended that extensions are used sparingly. Backwards compatibility with proprietary extensions will NOT be guaranteed in future versions of these specifications.*

4.8 V1.x/V2.0 Issues & Compatibility

The 'Outcomes Processing' functionality is not backwards compatible with early versions of the specification. These functions that have been replaced in earlier specifications are listed in Table 4.4.

Table 4.4 List of elements replaced in earlier specifications.

Previous Element Name	New Element Name	Description
sectionprocessing	selection	The original outcomes processing container for Sections.
assessmentprocessing	assessment	The original outcomes processing container for Assessments.

V1.0, V1.01 and V1.1 QTI-XML instances that do not use the elements listed within Table 4.4 are backwards compatible because the <outcomes_processing> element within V1.2 is optional.

4.9 IMS Harmonization

The issues of harmonization of this part of the IMS QTI specification with other IMS specifications is addressed by:

- IMS Meta-data – the <outcomes_metadata> element should be capable of operating on any form of the entries defined within the IMS Meta-data specification.

Appendix A – Glossary of Terms

A1 – General Terms

attempted	Several of the scoring algorithms differentiate between objects that were attempted and those that were not. Exact details of what constitutes an attempt will vary according to the exact details of the presentation environment, however, it is expected that the participants will know what actions on their parts constitute attempts. A typical example would be a submit button where pressing the button constitutes an attempt, where as moving to the next object using navigations controls to skip to the next item without pressing submit would not be considered an attempt.
object	In the context of a scoring algorithm an object is an Item or Section that was selected for presentation to the participant and hence must (at least potentially) be operated on by the scoring algorithm. Item and section objects are nearly identical in that the both contain meta-data, which can be used for subset selection, and the both report outcome variables that are operated on by the items.
presented	An Item is presented if it is displayed on whatever device is used for rendering items and the participant is given an opportunity to respond to it. A Section is presented if its selection rules have been used and at least one Item selected (directly or indirectly) by those rules has been presented.
reliability	A term from psychometrics, the reliability of a score on an Assessment is defined as the correlation between the scores by the same participant taking the same Assessment twice (in an ideal world where the participant forgets all of the items and learns nothing new between administrations). As a correlation, this is a number between -1 and 1 (with negative results not seen in practice). Higher values are better. Practically, there are several methods for estimating reliability of a score and the method used should be described in the *interpretvar* for the score.
selected	Selected objects appear in two contexts. First, the selection and ordering algorithm selects a set of sections and items for presentation as part of a section. These may or may not be presented, depending on what happens in the course of the Assessment (in particular, items may be selected but not presented due to time-outs). The selected objects and not just the presented ones are the ones operated on by the selection algorithm. Second, the *outcomes_condition* element can select a subset of the objects selected by selection and ordering for use with a particular scoring algorithm. This is particularly useful for defining sub-scores.
standard-error	The “chance error” in the estimation of the score, i.e., the measurement error of the Assessment instrument. In an ideal experiment where a participant could take the same assessment multiple times, forgetting the items and not changing in between, this would be the standard deviations of the scores so obtained. Practically, there are several methods for estimating the reliability of a score and the *interpretvar* should provide information about which one is used.
sub-score	A score defined on only a portion of the objects selected by the selection and ordering conditions.

A2 – Elements & Attributes

and_objects The *and_objects* element is used to create complex logical ('AND') score processing rules. In this rule all of the clauses must be true before the object contributes to the score. This construct is commonly used to identify the set of meta-data values that must be true before the object can contribute to the outcomes aggregation. Appendix B contains a detailed description of the logic rules.

and_test The *and_test* element is used to create a complex logical ('AND') test on score variables to determine what feedback will be presented. All of the contained clauses must be 'True' for this state to be 'True'. This mechanism is used to provide the participant with feedback on the Section or Assessment.

cutvalue The *cutvalue* attribute is used to define the value for a score above which the participant is declared to have achieved mastery. This is an attribute of the *decvar* element.

decvar The *decvar* element is used to declare the variables that are to be required to support the ASI scoring models. The integer variable 'SCORE' is the default declaration with a value of zero. The name, type and default value of the variables is determined by the *varname*, *vartype* and *defaultval* attributes. Other relevant attributes are *minvalue*, *maxvalue* and *members*.

defaultval The *defaultval* attribute is used by the *decvar* element to define the start value for the variable. This value is bounded by the *minvalue* and *maxvalue* attributes, when used. The type of value is determined by the type of variable declared i.e. it should conform to the *vartype* attribute as used by the *decvar* element.

displayfeedback The *displayfeedback* element is the trigger for the presentation of feedback to the users. The type of feedback to be displayed is determined by the *feedbacktype* attribute. The *linkrefid* attribute is used to identify the element containing the feedback.

feedbacktype The *feedbacktype* attribute is used by the *displayfeedback* element to denote the type of feedback that is to be invoked. The default value for the *feedbacktype* attribute is 'Response'. This attribute is required because the element containing the feedback may have more than one type of feedback within its body. At the current time IMS QTI hints and solutions are not supported with the Assessment and Section structures.

interpretvar The *interpretvar* element is used to describe statistical features about the associated variable. The variable that is associated is identified by the *varname* attribute. The primary usage for this element is to enable descriptions about the significance of the variables to be associated with their declaration - statistical parameters will be of particular importance.

linkrefid The *linkrefid* attribute is used to associate the feedback trigger, *displayfeedback*, with the outcomes feedback in the Section and/or Assessment. This means that only a single copy of the data structure is required with multiple references to it.

map_input The *map_input* element is used to change the algorithm specific default name of the input variable (e.g. 'SCORE', 'CORRECT') that is used as the source variable for the evaluation of objects (items and section) in a scoring algorithm. This enables the scoring algorithm to use different item and section level outcome variables in the scoring algorithm.

map_output The *map_output* element is used to change the default name of the result variable reported out by an outcome_processing algorithm (e.g., 'SCORE', 'COUNT'). This enables the scoring algorithm to provide more descriptive output names and to use names to distinguish various subscores.

maxvalue The *maxvalue* attribute is used with the *decvar* element to define the maximum value that can be assigned to the declared variable. This also limits the maximum value that can be assigned as part of the processing. This value does not take into account any weighting that may be applied to the object. This **must** be declared if the '*.maximum' and '*.normalized' variables are to be correctly defined.

mdname The *mdname* operator is used with the *outcomes_metadata* element to identify the meta-data field that is to be used. The meta-data field may be contained within either the IMS QTI-specific fields or the externally linked IMS Meta-data field of the object. It is the responsibility of the host system to determine the existence or otherwise of the meta-data field.

mdoperator The *mdoperator* is used with the *outcomes_metadata* element to identify the comparison operation to be applied on the meta-data field. The set of possible operators is defined as and enumerated set and consists of: EQ, NEQ, LT, LTE, GT, GTE. This set is derived from the operators normally supported by database access languages. Operations on strings should be limited to the EQ and NEQ otherwise the outcome is undefined.

members The *members* attribute is used by the *decvar* element to define the members of the set being defined. The members of the set are defined as a comma separated list without any enclosing parentheses.

minvalue The *minvalue* attribute is used with the *decvar* element to define the minimum value that can be assigned to the declared variable. This also limits the minimum value that can be assigned as part of the processing. This value does not take into account any weighting that may be applied to the object. This **must** be declared if the '*.minimum' and '*.normalized' variables are to be correctly defined.

not_objects The *not_objects* element is used to create complex logical ('NOT') score processing rules. In this rule the contained clause must be false before the object contributes to the score. This construct is commonly used to identify the set of meta-data values that must be true before the object can contribute to the outcomes aggregation. Appendix B contains a detailed description of the logic rules.

not_test The *not_test* element is used to create a complex logical ('NOT') test on score variables to determine the type of feedback that will be presented. The contained clauses must be 'False' for this state to be 'True'. This mechanism is used to provide the participant with feedback on the Section or Assessment.

objects_condition This element is the container for all of the conditions that are to be applied to each of the aggregation objects (Sections and/or Items). The scoring variables for all of the objects that obey these conditions must be used by the scoring engine to produce the corresponding aggregated score.

objects_parameter The *objects_parameter* element is used within the *objects_condition* element to pass key parameters to the system that is used to test the eligibility of the objects that are to contribute to the aggregated scoring. The name of the associated parameter is defined using the *pname* attribute.

objectscond_extension This element is the extension facility within the *objects_condition* element. It should be used when the current conditions that can be applied to determine which objects are to contribute to the aggregated outcomes processing have insufficient functionality.

or_objects The *or_objects* element is used to create complex logical ('OR') score processing rules. In this rule at least one of the clauses must be true before the object contributes to the score. This construct is commonly used to identify the set of meta-data values that must be true before the object can contribute to the outcomes aggregation. Appendix B contains a detailed description of the logic rules.

or_test	The *or_test* element is used to create a complex logical ('OR') test on score variables to determine the type of feedback that will be presented. At least one of the contained clauses must be 'True' for this state to be 'True'. This mechanism is used to provide the participant with feedback on the Section or Assessment.
outcomes	The *outcomes* element is used to contain the declaration of the variables for the outcomes processing. The *outcomes* element contains the *decvar* and *interpretvar* sub-elements to define the scoring variables.
outcomes_feedback_test	The *outcomes_feedback_test* element is used to contain the tests that are applied to the aggregated scores to determine the corresponding participant feedback. The actual feedback is contained in the *sectionfeedback* or *assessfeedback* elements as appropriate. The *title* attribute is used to contain the name of the feedback test set.
outcomes_metadata	The *outcomes_metadata* element is used to contain the rules that are applied to the meta-data fields of the object. This element uses the *mdname* and *mdoperator* attributes to define the meta-data label and the type of comparison respectively. Complex combinations of the meta-data values can be constructed using the *and_object*, *or_object* and *not_object* elements.
outcomes_processing	The *outcomes_processing* element is used to contain all of the aggregated outcomes processing instructions that are used by the parent Section or Assessment. The type of scoring algorithm that is assigned to the object is defined using the *scoremodel* attribute.
pname	The *pname* attribute is used by the *objects_parameter* and *processing_parameter* elements to identify the name of the parameter that is to be passed to the appropriate component of the scoring algorithm. There is no naming convention for these parameters and each parameter will, in general, be used differently by each scoring algorithm.
processing_parameter	The *processing_parameter* element is used within the *outcomes_processing* element to pass key parameters to the scoring algorithm itself e.g. this is used for the 'BestKofN' algorithm to define the value of 'K'. The name of the associated parameter is defined using the *pname* attribute.
qticomment	This is the commenting facility within the XML schemas. The comments can take any form supported as #PCDATA. The key difference between this comment style and the standard '<!-- *** -->' is that the former is passed through the XML parser to the host system.
scoremodel	The *scoremodel* attribute is used by the *outcomes_processing* element to identify that type of scoring model being used. The current implied default value is 'SumofScores'. This attribute allows vendors to clearly identify when they are using proprietary scoring models.
test_variable	The *test_variable* element is used within the *outcomes_feedback_test* element to contain the actual tests that are applied to the scoring variables. Complex test rules are constructed using the *and_test*, *or_test* and *not_test* elements.
testoperator	The *testoperator* attribute is used by the *variable_test* element to define the type of test being applied to the scoring variable. The possible tests are enumerated as: EQ, NEQ, LT, LTE, GT, GTE i.e. equal to, not equal to, less than, than or equal to, greater than, greater than or equal to.
title	The *title* attribute is an optional naming of the associated element. The title is used to help readability of the XML files. The title is usually 1-256 characters in length.

variable_test	The variable_test element is the actual test rule that is to be applied to an identified scoring variable. The variable is identified using the *varname* attribute and the type of test is defined using the *testoperator* attribute. Complex tests are constructed using this element within the *and_test*, *or_test* and *not_test* elements.
varname	The *varname* attribute is used to define the name of the variable. This attribute is used by the *decvar* and *interpretvar* elements. The name can consist of any lowercase and uppercase characters i.e. a-z, A-Z, 0-9 and underscore (0-9 should not be used for the first character). The default variable name is 'SCORE'.
vartype	The *vartype* attribute is used by the *decvar* element to define the type of variable being declared. The type of variable is: String, Boolean, Integer, Decimal, Scientific and Enumerated. The 'Enumerated' option enables the declaration of typed entries and the range of types is defined within the *members* attribute that must accompany the declaration.
view	The view attribute is used to define the 'actors' permitted to see the associated information e.g. feedback, objectives, etc. The supported actors are All (used to indicate access to all), Administrating Authority, Administrator, Assessor, Author, Candidate, Invigilator/Proctor, Psychometrician, Scorer and Tutor.
xml:lang	The *xml:lang* attribute is used wherever the language of the entry text can be varied. This attribute is used to define the language of the associated text. The format of the attribute shows that it is one of the core attributes provided by XML itself.

Appendix B – In-Built Scoring Algorithms

B1 – 'Number Correct' Scoring Algorithm

Name: NumberCorrect

Description: This is a count of the number of objects that have been presented and scored as correct. An object is included within this scoring algorithm if it has a variable that has been declared as 'CORRECT' and with type Boolean. The object is defined as correct when CORRECT=True.

Default Variables:

Name	*Type*	*Usage*
COUNT	Integer	The total number of attempted objects with CORRECT=True.
COUNT.min	Integer	Set to 0 (zero).
COUNT.max	Integer	Set to the number of objects that were attempted.
COUNT.normalized	Real	The normalized value (range 0-1) for the number of correct answers.

Constraints: The variable 'CORRECT' must have been declared as a Boolean type otherwise the <map_input> instruction must identify the new variable name.

Algorithm:

```
parentobject.COUNT = 0
parentobject.COUNT.min = 0
parentobject.COUNT.max = 0
parentobject.COUNT.normalized = 0.0
for ALL_SELECTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(CORRECT)) = True
      then
         begin
            parentobject.COUNT.max = parentobject.COUNT.max + 1
            if childobject.CORRECT = True
            then parentobject.COUNT = parentobject.COUNT + 1
         endif
   enddo
parentobject.COUNT.normalized =
         parentobject.COUNT/parentobject.COUNT.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.COUNT' and 'parentobject.COUNT.max'.

Derive Parameters: Percentage Correct = parentobject.COUNT.normalized * 100
(assuming parentobject.COUNT.max ?0)

Notes: If this algorithm is used concurrently with another that operates on the 'COUNT' variables then the <map_output> element should be used to ensure that 'COUNT' variable is not overloaded and therefore provides unreliable results.

B2 – 'Number Correct (Attempted)' Scoring Algorithm

Name: NumberCorrectAttempted

Description: This is a count of the number of objects that have been attempted (this may be a subset of those selected by the system) and scored as correct. An object is included within this scoring algorithm if it has a variable that has been declared as 'CORRECT' and with type Boolean. The object is defined as correct when CORRECT=True.

Default Variables:

Name	*Type*	*Usage*
COUNT	Integer	The total number of attempted objects with CORRECT=True.
COUNT.min	Integer	Set to 0 (zero).
COUNT.max	Integer	Set to the number of objects that were attempted.
COUNT.normalized	Real	The normalized value (range 0-1) for the number of correct answers.

Constraints: The variable 'CORRECT' must have been declared as a Boolean type otherwise the <map_input> instruction must identify the new variable name.

Algorithm:

```
parentobject.COUNT = 0
parentobject.COUNT.min = 0
parentobject.COUNT.max = 0
parentobject.COUNT.normalized = 0.0
for ALL_ATTEMPTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(CORRECT)) = True
      then
         begin
            parentobject.COUNT.max = parentobject.COUNT.max + 1
            if childobject.CORRECT = True
            then parentobject.COUNT = parentobject.COUNT + 1
         endif
   enddo
parentobject.COUNT.normalized =
         parentobject.COUNT/parentobject.COUNT.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.COUNT' and 'parentobject.COUNT.max'.

Derive Parameters: Percentage Correct = parentobject.COUNT.normalized * 100
(assuming parentobject.COUNT.max ?0)

Notes: If this algorithm is used concurrently with another that operates on the 'COUNT' variables then the <map_output> element should be used to ensure that 'COUNT' variable is not overloaded and therefore provides unreliable results.

B3 – 'Weighted Number Correct' Scoring Algorithm

Name: WeightedNumberCorrect

Description: This is a weighted count of the number of objects that have been presented and scored as correct. An object is included within this scoring algorithm if it has a variable that has been declared as 'CORRECT' and with type Boolean. The object is defined as correct when CORRECT=True.

Default Variables:

Name	*Type*	*Usage*
COUNT	Integer	The total weighted number of attempted objects with CORRECT=True.

	COUNT.min	Integer	Set to 0 (zero).
	COUNT.max	Integer	Set to the sum of the weights of the number of objects attempted.
	COUNT.normalized	Real	The normalized value (range 0-1) for the number of correct answers.

Constraints: The variable 'CORRECT' must have been declared as a Boolean type otherwise the <map_input> instruction must identify the new variable name.
The child objects must have meta-data entries that contain the meta-data field 'qmd_weighting'. In the case where no weighting has been defined then it is assumed to be one (1).

Algorithm:

```
parentobject.COUNT = 0
parentobject.COUNT.min = 0
parentobject.COUNT.max = 0
parentobject.COUNT.normalized=0.0
for ALL_SELECTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(CORRECT)) = True
      then
         begin
            parentobject.COUNT.max = parentobject.COUNT.max
                                   + childobject.weighting
            if childobject.CORRECT = True
            then parentobject.COUNT = parentobject.COUNT
                                   + childobject.weighting
         endif
   enddo
parentobject.COUNT.normalized =
         parentobject.COUNT/parentobject.COUNT.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.COUNT' and 'parentobject.COUNT.max'.

Derive Parameters: Percentage Correct = parentobject.COUNT.normalized * 100
(assuming parentobject.COUNT.max ?0)

Notes: If this algorithm is used concurrently with another that operates on the 'COUNT' variables then the <map_output> element should be used to ensure that 'COUNT' variable is not overloaded and therefore provides unreliable results.

B4 – 'Weighted Number Correct (Attempted)' Scoring Algorithm

Name: WeightedNumberCorrectAttempted

Description: This is a weighted count of the number of objects that have been presented and attempted, and scored as correct. An object is included within this scoring algorithm if it has a variable that has been declared as 'CORRECT' and with type Boolean. The object is defined as correct when CORRECT=True.

Default Variables:

Name	*Type*	*Usage*
COUNT	Integer	The total weighted number of attempted objects with CORRECT=True.
COUNT.min	Integer	Set to 0 (zero).
COUNT.max	Integer	Set to the sum of the weights of the number of objects attempted.
COUNT.normalized	Real	The normalized value (range 0-1) for the number of correct answers.

Constraints: The variable 'CORRECT' must have been declared as a Boolean type otherwise the <map_input> instruction must identify the new variable name.
The child objects must have meta-data entries that contain the meta-data field 'qmd_weighting'. In the case where no weighting has been defined then it is assumed to be one (1).

Algorithm:

```
parentobject.COUNT = 0
parentobject.COUNT.min = 0
parentobject.COUNT.max = 0
parentobject.COUNT.normalized=0.0
for ALL_ATTEMPTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(CORRECT)) = True
      then
         begin
            parentobject.COUNT.max = parentobject.COUNT.max
                                   + childobject.weighting
            if childobject.CORRECT = True
            then parentobject.COUNT = parentobject.COUNT
                                   + childobject.weighting
         endif
   enddo
parentobject.COUNT.normalized =
         parentobject.COUNT/parentobject.COUNT.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.COUNT' and 'parentobject.COUNT.max'.

Derive Parameters: Percentage Correct = parentobject.COUNT.normalized * 100
(assuming parentobject.COUNT.max ?0)

Notes: If this algorithm is used concurrently with another that operates on the 'COUNT' variables then the <map_output> element should be used to ensure that 'COUNT' variable is not overloaded and therefore provides unreliable results.

B5 – 'Parameter Weighted Number Correct' Scoring Algorithm

Name: ParameterWeightedNumberCorrect

Description: This is a weighted count of the number of objects that have been presented and scored as correct. An object is included within this scoring algorithm if it has a variable that has been declared as 'CORRECT' and with type Boolean. The object is defined as correct when CORRECT=True.

Default Variables:

Name	*Type*	*Usage*
COUNT	Integer	The total weighted number of attempted objects with CORRECT=True.
COUNT.min	Integer	Set to 0 (zero).
COUNT.max	Integer	Set to the sum of the weights of the number of objects attempted.
COUNT.normalized	Real	The normalized value (range 0-1) for the number of correct answers.

Constraints: The variable 'CORRECT' must have been declared as a Boolean type otherwise the <map_input> instruction must identify the new variable name.
The weighting that is to be applied is defined using the values assigned in the outcomes processing element statement <objects_parameter pname="qmd_weighting">. In the case where no weighting has been defined then it is assumed to be one (1).

Algorithm:

```
parentobject.COUNT = 0
parentobject.COUNT.min = 0
parentobject.COUNT.max = 0
parentobject.COUNT.normalized=0.0
for ALL_SELECTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(CORRECT)) = True
      then
         begin
            parentobject.COUNT.max = parentobject.COUNT.max
                                   + childobject.weighting
            if childobject.CORRECT = True
            then parentobject.COUNT = parentobject.COUNT
                                   + childobject.weighting
         endif
   enddo
parentobject.COUNT.normalized =
         parentobject.COUNT/parentobject.COUNT.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.COUNT' and 'parentobject.COUNT.max'.

Derive Parameters: Percentage Correct = parentobject.COUNT.normalized * 100
(assuming parentobject.COUNT.max ?0)

Notes: If this algorithm is used concurrently with another that operates on the 'COUNT' variables then the <map_output> element should be used to ensure that 'COUNT' variable is not overloaded and therefore provides unreliable results.

B6 – 'Parameter Weighted Number Correct (Attempted)' Scoring Algorithm

Name: ParameterWeightedNumberCorrectAttempted

Description: This is a weighted count of the number of objects that have been presented, attempted and scored as correct. An object is included within this scoring algorithm if it has a variable that has been declared as 'CORRECT' and with type Boolean. The object is defined as correct when CORRECT=True.

Default Variables:

Name	*Type*	*Usage*
COUNT	Integer	The total weighted number of attempted objects with CORRECT=True.
COUNT.min	Integer	Set to 0 (zero).
COUNT.max	Integer	Set to the sum of the weights of the number of objects attempted.
COUNT.normalized	Real	The normalized value (range 0-1) for the number of correct answers.

Constraints: The variable 'CORRECT' must have been declared as a Boolean type otherwise the <map_input> instruction must identify the new variable name.
The weighting that is to be applied is defined using the values assigned in the outcomes processing element statement <objects_parameter pname="qmd_weighting">. In the case where no weighting has been defined then it is assumed to be one (1).

Algorithm:

```
parentobject.COUNT = 0
parentobject.COUNT.min = 0
parentobject.COUNT.max = 0
parentobject.COUNT.normalized=0.0
for ALL_ATTEMPTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(CORRECT)) = True
      then
         begin
            parentobject.COUNT.max = parentobject.COUNT.max
                                   + childobject.weighting
            if childobject.CORRECT = True
            then parentobject.COUNT = parentobject.COUNT
                                   + childobject.weighting
         endif
   enddo
parentobject.COUNT.normalized =
         parentobject.COUNT/parentobject.COUNT.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.COUNT' and 'parentobject.COUNT.max'.

Derive Parameters: Percentage Correct = parentobject.COUNT.normalized * 100
(assuming parentobject.COUNT.max ?0)

Notes: If this algorithm is used concurrently with another that operates on the 'COUNT' variables then the <map_output> element should be used to ensure that 'COUNT' variable is not overloaded and therefore provides unreliable results.

B7 – 'Sum of Scores' Scoring Algorithm

Name: SumofScores

Description: This is the sum of the scores assigned to the individual child objects selected for presentation to the participant. An object is included within the scoring algorithm if it has a variable that has been declared as 'SCORE' and with type integer.

Default Variables:

Name	*Type*	*Usage*
SCORE	Integer	The sum of the individual 'SCORE; values assigned within each of the child objects.
SCORE.min	Integer	The sum of the individual 'minvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.max	Integer	The sum of the individual 'maxvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.normalized	Real	The normalized value (range 0-1) for the total score.

Constraints: The 'SCORE' integer variable must be used otherwise the <map_input> instruction must identify the new variable name.
The 'minvalue' attribute for the variable must have been defined. If undefined then the SCORE.min value is undefined.
The 'maxvalue' attribute for the variable must have been defined. If undefined then the SCORE.max value is undefined.

Algorithm:

```
parentobject.SCORE = 0
parentobject.SCORE.min = 0
parentobject.SCORE.max = 0
parentobject.SCORE.normalized = 0.0
for ALL_SELECTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(SCORE)) = True
      then
         begin
            parentobject.SCORE.min = parentobject.SCORE.min
                                   + childobject.SCORE.minvalue
            parentobject.SCORE.max = parentobject.SCORE.max
                                   + childobject.SCORE.maxvalue
            parentobject.SCORE = parentobject.SCORE
                                   + childobject.SCORE
         endif
   enddo
parentobject.SCORE.normalized  =
   parentobject.SCORE-parentobject.SCORE.min/
   parentobject.SCORE.max-parentobject.SCORE.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.SCORE' and 'parentobject.SCORE.max'.

Derive Parameters: Percentage Score = parentobject.SCORE.normalized * 100

Notes: If this algorithm is used concurrently with another that operates on the 'SCORE' variables then the <map_output> element should be used to ensure that 'SCORE' variable is not overloaded and therefore provides unreliable results.

B8 – 'Sum of Scores (Attempted)' Scoring Algorithm

Name: SumofScoresAttempted

Description: This is the sum of the scores assigned to the individual attempted (this may be a subset of those selected) child objects. An object is included within the scoring algorithm if it has a variable that has been declared as 'SCORE' and with type integer.

Default Variables:

Name	*Type*	*Usage*
SCORE	Integer	The sum of the individual 'SCORE; values assigned within each of the child objects.
SCORE.min	Integer	The sum of the individual 'minvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.max	Integer	The sum of the individual 'maxvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.normalized	Real	The normalized value (range 0-1) for the total score.

Constraints: The 'SCORE' integer variable must be used otherwise the <map_input> instruction must identify the new variable name.
The 'minvalue' attribute for the variable must have been defined. If undefined then the SCORE.min value is undefined.
The 'maxvalue' attribute for the variable must have been defined. If undefined then the SCORE.max value is undefined.

Algorithm:

```
parentobject.SCORE = 0
parentobject.SCORE.min = 0
parentobject.SCORE.max = 0
parentobject.SCORE.normalized = 0.0
for ALL_ATTEMPTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(SCORE)) = True
      then
         begin
            parentobject.SCORE.min = parentobject.SCORE.min
                                   + childobject.SCORE.minvalue
            parentobject.SCORE.max = parentobject.SCORE.max
                                   + childobject.SCORE.maxvalue
            parentobject.SCORE = parentobject.SCORE
                                   + childobject.SCORE
         endif
   enddo
parentobject.SCORE.normalized  =
   parentobject.SCORE-parentobject.SCORE.min/
   parentobject.SCORE.max-parentobject.SCORE.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.SCORE' and 'parentobject.SCORE.max'.

Derive Parameters: Percentage Score = parentobject.SCORE.normalized * 100

Notes: If this algorithm is used concurrently with another that operates on the 'SCORE' variables then the <map_output> element should be used to ensure that 'SCORE' variable is not overloaded and therefore provides unreliable results.

B9 – 'Weighted Sum of Scores' Scoring Algorithm

Name: WeightedSumofScores

Description: This is the weighted sum of the scores assigned to the individual child objects that have been presented. An object is included within the scoring algorithm if it has a variable that has been declared as 'SCORE' and with type integer.

Default Variables:

Name	*Type*	*Usage*
SCORE	Integer	The weighted sum of the individual 'SCORE; values assigned within each of the child objects.
SCORE.min	Integer	The weighted sum of the individual 'minvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.max	Integer	The weighted sum of the individual 'maxvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.normalized	Real	The normalized value (range 0-1) for the total score.

Constraints: The 'SCORE' integer variable must be used otherwise the <map_input> instruction must identify the new variable name.
The 'minvalue' attribute for the variable must have been defined. If undefined then the SCORE.min value is undefined.
The 'maxvalue' attribute for the variable must have been defined. If undefined then the SCORE.max value is undefined.
The child objects must have meta-data entries that contain the meta-data field 'qmd_weighting'. Where no weighting has been defined then it is set to be one (1).

Algorithm:

```
parentobject.SCORE = 0
parentobject.SCORE.min = 0
parentobject.SCORE.max = 0
parentobject.SCORE.normalized = 0.0
for ALL_SELECTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(SCORE)) = True
      then
         begin
            parentobject.SCORE.min = parentobject.SCORE.min
                                   + (childobject.SCORE.minvalue
                                     * childobject.weighting)
            parentobject.SCORE.max = parentobject.SCORE.max
                                   + (childobject.SCORE.maxvalue
                                     * childobject.weighting)
            parentobject.SCORE = parentobject.SCORE
                                 + (childobject.SCORE
                                   * childobject.weighting)
         endif
   enddo
parentobject.SCORE.normalized  =
   parentobject.SCORE-parentobject.SCORE.min/
   parentobject.SCORE.max-parentobject.SCORE.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.SCORE' and 'parentobject.SCORE.max'.

Derive Parameters: Percentage Score = parentobject.SCORE.normalized * 100

Notes: If this algorithm is used concurrently with another that operates on the 'SCORE' variables then the <map_output> element should be used to ensure that 'SCORE' variable is not overloaded and therefore provides unreliable results.

B10 – 'Weighted Sum of Scores (Attempted)' Scoring Algorithm

Name: WeightedSumofScoresAttempted

Description: This is the weighted sum of the scores assigned to the individual child objects that have been presented and attempted. An object is included within the scoring algorithm if it has a variable that has been declared as 'SCORE' and with type integer.

Default Variables:

Name	*Type*	*Usage*
SCORE	Integer	The weighted sum of the individual 'SCORE; values assigned within each of the child objects.
SCORE.min	Integer	The weighted sum of the individual 'minvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.max	Integer	The weighted sum of the individual 'maxvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.normalized	Real	The normalized value (range 0-1) for the total score.

Constraints: The 'SCORE' integer variable must be used otherwise the <map_input> instruction must identify the new variable name.
The 'minvalue' attribute for the variable must have been defined. If undefined then the SCORE.min value is undefined.
The 'maxvalue' attribute for the variable must have been defined. If undefined then the SCORE.max value is undefined.
The child objects must have meta-data entries that contain the meta-data field 'qmd_weighting'. Where no weighting has been defined then it is set to be one (1).

Algorithm:

```
parentobject.SCORE = 0
parentobject.SCORE.min = 0
parentobject.SCORE.max = 0
parentobject.SCORE.normalized = 0.0
for ALL_ATTEMPTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(SCORE)) = True
      then
         begin
            parentobject.SCORE.min = parentobject.SCORE.min
                                   + (childobject.SCORE.minvalue
                                     * childobject.weighting)
            parentobject.SCORE.max = parentobject.SCORE.max
                                   + (childobject.SCORE.maxvalue
                                     * childobject.weighting)
            parentobject.SCORE = parentobject.SCORE
                               + (childobject.SCORE
                                 * childobject.weighting)
         endif
   enddo
parentobject.SCORE.normalized  =
   parentobject.SCORE-parentobject.SCORE.min/
   parentobject.SCORE.max-parentobject.SCORE.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.SCORE' and 'parentobject.SCORE.max'.

Derive Parameters: Percentage Score = parentobject.SCORE.normalized * 100

Notes: If this algorithm is used concurrently with another that operates on the 'SCORE' variables then the <map_output> element should be used to ensure that 'SCORE' variable is not overloaded and therefore provides unreliable results.

B11 – 'Parameter Weighted Sum of Scores' Scoring Algorithm

Name: ParameterWeightedSumofScores

Description: This is the weighted sum of the scores assigned to the individual child objects that have been presented. An object is included within the scoring algorithm if it has a variable that has been declared as 'SCORE' and with type integer.

Default Variables:

Name	*Type*	*Usage*
SCORE	Integer	The weighted sum of the individual 'SCORE; values assigned within each of the child objects.
SCORE.min	Integer	The weighted sum of the individual 'minvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.max	Integer	The weighted sum of the individual 'maxvalue' entries assigned to the 'SCORE' variable for each child object.

	SCORE.normalized	Real	The normalized value (range 0-1) for the total score.

Constraints:

The 'SCORE' integer variable must be used otherwise the <map_input> instruction must identify the new variable name.
The 'minvalue' attribute for the variable must have been defined. If undefined then the SCORE.min value is undefined.
The 'maxvalue' attribute for the variable must have been defined. If undefined then the SCORE.max value is undefined.
The weighting that is to be applied is defined using the values assigned in the outcomes processing element statement <objects_parameter pname="qmd_weighting">. In the case where no weighting has been defined then it is assumed to be one (1).

Algorithm:

```
parentobject.SCORE = 0
parentobject.SCORE.min = 0
parentobject.SCORE.max = 0
parentobject.SCORE.normalized = 0.0
for ALL_SELECTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(SCORE)) = True
      then
         begin
            parentobject.SCORE.min = parentobject.SCORE.min
                                   + (childobject.SCORE.minvalue
                                     * childobject.weighting)
            parentobject.SCORE.max = parentobject.SCORE.max
                                   + (childobject.SCORE.maxvalue
                                     * childobject.weighting)
            parentobject.SCORE = parentobject.SCORE
                                 + (childobject.SCORE
                                   * childobject.weighting)
         endif
   enddo
parentobject.SCORE.normalized  =
   parentobject.SCORE-parentobject.SCORE.min/
   parentobject.SCORE.max-parentobject.SCORE.max
```

Sequencing:

Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.SCORE' and 'parentobject.SCORE.max'.

Derive Parameters:

Percentage Score = parentobject.SCORE.normalized * 100

Notes:

If this algorithm is used concurrently with another that operates on the 'SCORE' variables then the <map_output> element should be used to ensure that 'SCORE' variable is not overloaded and therefore provides unreliable results.

B12 – 'Parameter Weighted Sum of Scores (Attempted)' Scoring Algorithm

Name:

ParameterWeightedSumofScoresAttempted

Description:

This is the weighted sum of the scores assigned to the individual child objects that have been presented and attempted. An object is included within the scoring algorithm if it has a variable that has been declared as 'SCORE' and with type integer.

Default Variables:

Name	*Type*	*Usage*
SCORE	Integer	The weighted sum of the individual 'SCORE; values assigned within each of the child objects.
SCORE.min	Integer	The weighted sum of the individual 'minvalue' entries assigned to the 'SCORE' variable for each child object.

SCORE.max	Integer	The weighted sum of the individual 'maxvalue' entries assigned to the 'SCORE' variable for each child object.
SCORE.normalized	Real	The normalized value (range 0-1) for the total score.

Constraints: The 'SCORE' integer variable must be used otherwise the <map_input> instruction must identify the new variable name.
The 'minvalue' attribute for the variable must have been defined. If undefined then the SCORE.min value is undefined.
The 'maxvalue' attribute for the variable must have been defined. If undefined then the SCORE.max value is undefined.
The weighting that is to be applied is defined using the values assigned in the outcomes processing element statement <objects_parameter pname="qmd_weighting">. In the case where no weighting has been defined then it is assumed to be one (1).

Algorithm:

```
parentobject.SCORE = 0
parentobject.SCORE.min = 0
parentobject.SCORE.max = 0
parentobject.SCORE.normalized = 0.0
for ALL_ATTEMPTED(childobjects) do
   begin
      if child.object(HAS_VARIABLE(SCORE)) = True
      then
         begin
            parentobject.SCORE.min = parentobject.SCORE.min
                                   + (childobject.SCORE.minvalue
                                     * childobject.weighting)
            parentobject.SCORE.max = parentobject.SCORE.max
                                   + (childobject.SCORE.maxvalue
                                     * childobject.weighting)
            parentobject.SCORE = parentobject.SCORE
                                 + (childobject.SCORE
                                   * childobject.weighting)
         endif
   enddo
parentobject.SCORE.normalized  =
   parentobject.SCORE-parentobject.SCORE.min/
   parentobject.SCORE.max-parentobject.SCORE.max
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of 'parentobject.SCORE' and 'parentobject.SCORE.max'.

Derive Parameters: Percentage Score = parentobject.SCORE.normalized * 100

Notes: If this algorithm is used concurrently with another that operates on the 'SCORE' variables then the <map_output> element should be used to ensure that 'SCORE' variable is not overloaded and therefore provides unreliable results.

B13 – 'Best K from N' Scoring Algorithm

Name: BestKfromN

Description: This is the sum of the best 'K' scores assigned to the individual child objects that have been attempted. An object is included within the scoring algorithm if it has a variable that has been declared as 'SCORE' and with type integer. This algorithm is normalized using all objects that have been selected and presented.

Default Variables:	*Name*	*Type*	*Usage*

SCORE	Integer	The sum of the individual best ‘K’ ‘SCORE’ values assigned across the set of attempted child objects.
SCORE.min	Integer	The sum of the individual worst ‘K’ ‘minvalue’ entries assigned to the ‘SCORE’ variable for each child object.
SCORE.max	Integer	The sum of the individual best ‘K’ ‘maxvalue’ entries assigned to the ‘SCORE’ variable for each child object.
SCORE.normalized	Real	The normalized value (range 0-1) for the total score.

Constraints: The ‘SCORE’ integer variable must be used otherwise the <map_input> instruction must identify the new variable name.
The ‘minvalue’ attribute for the variable must have been defined. If undefined then the SCORE.min value is undefined. All of the objects must have the same value for ‘minvalue’.
The ‘maxvalue’ attribute for the variable must have been defined. If undefined then the SCORE.max value is undefined. All of the objects must have the same value for ‘maxvalue’.
The <processing_parameter> instruction must be used to assign an integer value to ‘K’. In the absence of this instruction then ‘K’ is set to the number of objects attempted.

Algorithm:

```
parentobject.SCORE = 0
parentobject.SCORE.min = 0
parentobject.SCORE.max = 0
parentobject.SCORE.normalized=0.0
parentobject.SCORE.max = value_of_K  * childobject.SCORE.maxvalue
parentobject.SCORE.min = value_of_K  * childobject.SCORE.minvalue
SORT (ALLPRESENTED_OBJECTS and child.object(HAS_VARIABLE(SCORE)
                        and ORDERED_BY_ DECREASING_SCORE)
for TOP(value_of_K) do
   begin
      parentobject.SCORE =
         parentobject.SCORE  + childobject.SCORE
   enddo
parentobject.SCORE.normalized =
   (parentobject.SCORE- parentobject.SCORE.min) /
   (parentobject.SCORE.max- parentobject.SCORE.min)
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant. This affects the values of ‘parentobject.SCORE’ and ‘parentobject.SCORE.max’.

Derive Parameters: Percentage Score = parentobject.SCORE.normalized * 100

Notes: If this algorithm is used concurrently with another that operates on the ‘SCORE’ variables then the <map_output> element should be used to ensure that ‘SCORE’ variable is not overloaded and therefore provides unreliable results.

B14 – ‘Guessing Penalty’ Scoring Algorithm

Name: GuessingPenalty

Description: This is the count of objects that have been answered correctly, incorrectly and which have not been attempted. An object is included within this scoring algorithm if it has a variable that has been declared as ‘CORRECT’ and with type Boolean. The object is defined as correct when CORRECT=True.

Default Variables:

Name	*Type*	*Usage*

	COUNT	Integer	The score calculated taking into account the relevant penalty.
	COUNT.correct	Integer	The number of objects answered correctly i.e. with CORRECT=True.
	COUNT.incorrect	Integer	The number of objects answered incorrectly i.e. with CORRECT=False.
	COUNT.unattempted	Integer	The number of objects not attempted

Constraints: The variable 'CORRECT must have been declared as a Boolean type otherwise the <map_input> instruction must identify the new variable name.
The child objects must have meta-data entries that contain the meta-data field 'qmd_penaltyvalue'. Where no penalty value has been defined then it is set to one (1).

Algorithm:

```
parentobject.COUNT = 0
parentobject.COUNT.correct = 0
parentobject.COUNT.incorrect = 0
parentobject.COUNT.unattempted = 0
for ALL_PRESENTED(childobjects)and (HAS_VARIABLE(CORRECT)  do
   begin
      if child.object(ATTEMPTED) = True
      then
         begin
            if childobject.CORRECT = True
            then
               begin
                  parentobject.COUNT.correct =
                     parentobject.COUNT .correct + 1
                  parentobject.COUNT = parentobject.COUNT + 1
               end
            else
               begin
                  parentobject.COUNT.incorrect =
                     parentobject.COUNT .incorrect + 1
                  parentobject.COUNT = parentobject.COUNT –
                        childobject.penaltyvalue
               end
         endif
      else parentobject.COUNT.unattempted =
         parentobject.COUNT.unattempted + 1
   enddo
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant.

Derive Parameters: None.

Notes: If this algorithm is used concurrently with another that operates on the 'COUNT' variables then the <map_output> element should be used to ensure that 'COUNT' variable is not overloaded and therefore provides unreliable results.

B15 – 'Weighted Guessing Penalty' Scoring Algorithm

Name: WeightedGuessingPenalty

Description: This is the weighted count score for objects that have been answered correctly, incorrectly and which have not been attempted. An object is included within this scoring algorithm if it has a variable that has been declared as 'CORRECT' and with type Boolean. The object is defined as correct when CORRECT=True.

Default Variables:	*Name*	*Type*	*Usage*

	COUNT	Integer	The score calculated taking into account the relevant penalty.
	COUNT.correct	Integer	The number of objects answered correctly i.e. with CORRECT=True.
	COUNT.incorrect	Integer	The number of objects answered incorrectly i.e. with CORRECT=False.
	COUNT.unattempted	Integer	The number of objects not attempted

Constraints: The variable ‘CORRECT must have been declared as a Boolean type otherwise the <map_input> instruction must identify the new variable name.
The child objects must have meta-data entries that contain the meta-data field ‘qmd_penaltyvalue’. Where no penalty value has been defined then it is set to one (1).
The child objects must have meta-data entries that contain the meta-data field ‘qmd_weighting’. Where no penalty value has been defined then it is set to one (1).

Algorithm:

```
parentobject.COUNT = 0
parentobject.COUNT.correct = 0
parentobject.COUNT.incorrect = 0
parentobject.COUNT.unattempted = 0
for ALL_PRESENTED(childobjects)and (HAS_VARIABLE(CORRECT)  do
   begin
      if child.object(ATTEMPTED) = True
      then
         begin
            if childobject.CORRECT = True
            then
               begin
                  parentobject.COUNT.correct =
                     parentobject.COUNT .correct + 1
                  parentobject.COUNT = parentobject.COUNT +
                     parentobject.weighting
               end
            else
               begin
                  parentobject.COUNT.incorrect =
                     parentobject.COUNT .incorrect + 1
                  parentobject.COUNT =
                     parentobject.COUNT *childobject.weighting–
                     childobject.penaltyvalue
               end
         endif
      else parentobject.COUNT.unattempted =
         parentobject.COUNT.unattempted + 1
   enddo
```

Sequencing: Selection and ordering will determine which objects are presented and thus may be attempted by the participant.

Derive Parameters: None.

Notes: If this algorithm is used concurrently with another that operates on the ‘COUNT’ variables then the <map_output> element should be used to ensure that ‘COUNT’ variable is not overloaded and therefore provides unreliable results.

Appendix C – Logic Rules

The elements *or_objects*, *and_objects* and *not_objects* are used within the *objects_condition* element to select the Items and/or Sections whose scoring variables are to be aggregated. An aggregation rule takes the form such as:

<outcomes_metadata mdname="qmd_topic" mdoperator="EQ">trigonometry</outcomes_metadata>

i.e. select all of the objects that have a meta-data entry of “qmd_topic” that is equal to ‘trigonometry’.

The objects_condition element will contain a set of rules, i.e. $R_1, R_2, R_3 \ldots R_k$. The logic operators between these rules are defined as:

- $\neg R_1$ means ‘NOT R_1’ which is represented in QTIASI-XML as <not_objects>R_1</not_objects>
- R_1 and R_2 means ‘R_1 AND R_2’ which is represented in QTIASI-XML as <and_objects>R_1, R_2</and_objects>
- R_1 or R_2 means ‘R_1 OR R_2’ which is represented in QTIASI-XML as <or_objects>R_1, R_2</or_objects>

Using this initial set of constructions gives rise to the following representations:

- $\neg$ (R_1 OR R_2) which is represented in QTIASI-XML as

```
<not_objects>
   <or_objects>
         R1
         R2
   </or_objects>
   </not_objects>
```

- $\neg$ (R_1 AND R_2) which is represented in QTIASI-XML as

```
<not_objects>
   <and_objects>
         R1
         R2
   </and_objects>
</not_objects>
```

- ($\neg R_1$ AND $\neg R_2$) which is represented in QTIASI-XML as

```
<and_objects>
   <not_objects>R1</not_objects>
   <not_objects>R2</not_objects>
</and_objects>
```

- ($\neg R_1$ OR $\neg R_2$) which is represented in QTIASI-XML as

```
<or_objects>
   <not_objects>R1</not_objects>
   <not_objects>R2</not_objects>
</or_objects>
```

- ((R_1 OR R_2) AND (R_3 OR R_4)) which is represented in QTIASI-XML as

```
<and_objects>
   <or_objects>
         R1
         R2
   </or_objects>
   <or_objects>
         R3
         R4
   </or_objects>
</and_objects>
```

- $((R_1 \text{ OR } R_2 \text{ OR } R_3) \text{ AND } (R_4 \text{ OR } R_5 \text{ OR } (R_6 \text{ AND } R_7 \text{ AND } R_8)) \text{ AND } \neg(R_9 \text{ OR } R_{10} \text{ OR } R_{11}))$ which is represented in QTIASI-XML as

```
<and_objects>
   <or_objects>
         R1
         R2
         R3
   </or_objects>
   <or_objects>
         R4
         R5
      <and_objects>
            R6
            R7
            R8
      </and_objects>
   </or_objects>
   <or_objects>
      <not_objects>
         <or_objects>
               R9
               R10
               R11
         </or_objects>
      </not_objects>
   </or_objects>
</and_objects>
```

This set of rules gives rise to the following guidelines:

- Only a single <and>, <or> or <not> structure is required at the top-most level of the logic statement;
- Multiple occurrences of the <and>, <or> and <not> elements is permitted within <and> and <or> elements to allow the construction of complex logic statements;
- The <not> element can only contain a single element the result of which is to invert the logic of the test.

About This Document

Title	IMS Question & Test Interoperability ASI Outcomes Processing
Editors	Colin Smythe, Lane Brewer, and Steve Lay
Version	1.2
Version Date	11 February 2002
Status	**Final Specification**
Summary	This document presents the IMS QTI ASI Outcomes Processing Specification. This specification is one of the set of the IMS Question & Test Interoperability specifications. This Outcomes Processing specification is an extension to the IMS QTI ASI V1.2 specification.
Revision Information	22 January 2002
Purpose	Defines the 'Assessment and Section Outcome Processing' features that have been added to IMS QTI V1.1 as part of the development of the IMS QTI V1.2 specification.
Document Location	http://www.imsglobal.org/question/v1p2/imsqti_asi_outv1p2.html

List of Contributors

The following individuals contributed to the development of this document:

Russell Almond	ETS, USA
Lane Brewer	Galton Technologies Inc.
Todd Brewer	Galton Technologies Inc.
Russell Grocott	Can Studios Ltd.
Andy Heath	CETIS/JISC, UK
Paul Hilton	Can Studios Ltd.
Richard Johnson	Goal Design Inc.
John Kleeman	Question Mark Computing Ltd.
Steven Lay	University of Cambridge Local Examinations Syndicate, UK
Jez Lord	Can Studios Ltd.
Paul Roberts	Question Mark Computing Ltd.
Nial Sclater	CETIS/University of Strathclyde, UK
Eric Shepherd	Question Mark Corporation
Colin Smythe	Dunelm Services Ltd.

Revision History

Version No.	Release Date	Comments
Final Version 1.2	11 February 2002	The first release of the QTI ASI Outcomes Processing Final specification.

Index

E

Elements

H

I

M

O

P

Q

R

S

T

U

V

X