

## CORE EXTERNAL REQUEST



COR-10047483

QUSQ

CSIRO -- World Wide Web -- 2004- Current, delayed 180 days

International journal of distance education technologies

Ariel email: ariel@usq.edu.au Ariel IP: 139.86.208.56 ILL email: libdds@usq.edu.au  
Library

Australia

ATTN:	SUBMITTED:	2010-03-15	10:19:41
PHONE: 07 4631 2462	PRINTED:	2010-03-15	11:59:23
FAX: 07 4631 2920	REQUEST NO.:	COR-10047483	
E-MAIL:	SENT VIA:	ISO	
	EXPIRY DATE:	2010-03-24	
	EXTERNAL NO.:	279123	

---

COR	Core	Copy	Journal
-----	------	------	---------

---

TITLE:	INTERNATIONAL JOURNAL OF DISTANCE EDUCATION TECHNOLOGIES
VOLUME/ISSUE/PAGES:	5 (3) 8-23
DATE:	2007
AUTHOR OF ARTICLE:	Clark, Damien and Baillie-de Byl, Penny
TITLE OF ARTICLE:	ENHANCING THE IMS QTI TO BETTER SUPPORT COMPUTER ASSISTED MARKING
ISSN:	1539-3100
MAX COST:	\$20.00
COPYRIGHT COMP.:	Fair Dealing - S49
SHELF MARK:	Online access. Available for document delivery. Restrictions apply.
DELIVERY:	E-mail attachment: libdds@usq.edu.au
REPLY:	Mail:

This document contains 16 pages. You will be invoiced for \$13.20. This is NOT an invoice.



# Enhancing the IMS QTI to Better Support Computer Assisted Marking

Damien Clark, Central Queensland University, Australia

Penny Baillie-de Byl, University of Southern Queensland, Australia

---

## ABSTRACT

*Computer aided assessment is a common approach used by educational institutions. The benefits range into the design of teaching, learning, and instructional materials. While some such systems implement fully automated marking for multiple choice questions and fill-in-the-blanks, they are insufficient when human critiquing is required. Current systems developed in isolation have little regard to scalability and interoperability between courses, computer platforms, and learning management systems. The IMS Global Learning Consortium's open specifications for interoperable learning technology lack functionality to make it useful for computer assisted marking. This article presents an enhanced set of these standards to address the issue.*

*Keywords: assessment; computer aided assessment; computer assisted marking; distance education; educational technology; internet-based technology; interoperable learning technology; rubrics; technological innovations; xml*

---

## INTRODUCTION

Computer aided assessment (CAA), one of the recent trends in education technology, has become common-place in educational institutions as part of delivering course materials, particularly for large classes. This has been driven by many factors, such as:

- The need to reduce educational staff workloads (Dalziel, 2000; Jacobsen & Kremer, 2000; Jefferies, Constable et al., 2000;
- Pain & Heron, 2003; Peat, Franklin et al., 2001);
- A push for more timely feedback to students (Dalziel, 2001; Jefferies, Constable et al., 2000; Merat & Chung, 1997; Sheard & Carbone, 2000; Voit & Mason, 2000);
- Reduction in educational material development and delivery costs (Jefferies, Constable et al., 2000; Muldner & Currie, 1999); and,

- The proliferation of online education (White, 2000).

Internet-based technologies in CAA can be broadly categorised into the following system types: online quiz systems, fully automated marking, and semiautomated/computer assisted marking systems. The most common form of CAA, online quizzes, typically consist of multiple choice questions (MCQ) (IMS, 2000), as they can be automatically marked. Yet, there is much conjecture on the effectiveness of MCQs, particularly in the assessment of Bloom's higher learning outcomes (1956) such as analysis, synthesis, and evaluation (Davies, 2001). This limits the scope by which a student's abilities can be assessed. Short response and essay type questions are commonly used to assess the higher order skills of Bloom's taxonomy. Still, these types of assessments are time consuming to mark manually (Davies, 2001; White, 2000).

A more ambitious approach to CAA involves the use of fully-automated marking systems. These can be defined as systems that can mark electronically submitted assignments such as essays (Palmer, Williams et al., 2002) via online assignment submission management (OASM) (Benford, Burke et al., 1994; Darbyshire, 2000; Gayo, Gil et al., 2003; Huizinga, 2001; Jones & Behrens, 2003; Jones & Jamieson, 1997; Mason & Woit, 1999; Roantree & Keyes, 1998; Thomas, 2000; Trivedi, Kar et al., 2003), and automatically generate a final grade for the assignment with little to no interaction with a human marker. The obvious benefit to this approach is the ability to assess some higher order thinking as per Bloom's Taxonomy (1956) in a completely automated manner, thus improving marking turn-around times for large classes. Fully automated systems include MEAGER, which is designed to automatically mark Microsoft Excel spreadsheets (Hill, 2003), automatic essay marking systems, such as those evaluated by Palmer, Williams et al. (2002), and English and Siviter's system (2000) designed to assess student hypertext mark-up language (HTML) Web pages, to name a few. Unfortunately, this approach is not suitable for all as-

essment types and can often require significant time to develop the model solution. In addition, most of the automated functionality examines students' solutions against model solutions. This may lead to issues relating to marking quality when it is impossible for the assessment creator to identify all possible solutions.

The last approach is the use of semiautomated or computer assisted marking (CAM). This is a compromise between online quiz and fully automated systems. CAM assists with the reduction of poor marker consistency and the quantity and quality of feedback in marking team situations. By using CAM, many of the laborious and repetitive tasks associated with marking can be automated (Baillie-de Byl, 2004), resulting in more timely returns to students. CAM describes systems that have some components of the marking process automated, but still require at least some human interpretation and analysis to assign grades. For example, CAM systems have been developed to support the routine tasks associated with marking programming assignments, like compilation and testing of student submitted programs (Jackson, 2000; Joy & Luck, 1998). Although allocation of a final grade is the sole responsibility of the marker, this determination can be achieved faster, with greater accuracy and consistency, by relying on the results of automated tests (Joy & Luck, 1998). In cases where human interpretation and analysis occurs, this is referred to as manual marking.

One example of CAM is implemented in the Classmate system. It is designed to assist in automating many of the typical laborious tasks associated with marking, such as retrieval and presentation of submissions, feedback and grade storage, application of late penalties, and student returns (Baillie-de Byl, 2004). Other contributions in this area include an MS-Word Integrated CAM Template (Price & Petre, 1997), development of a CAM prototype based on research into how markers rate programming assignments (Preston & Shackleford, 1999), and Markin, a commercial CAM product by Creative Technology (Creative-Technology, 2005).

One of the major problems with current CAM systems is that much of the work is being undertaken by independent or small groups of researchers who are developing systems to service the particular needs of their courses and institutions, without regard for interoperability. The IMS global learning consortium (IMS, 2005) are addressing this problem through the production of open specifications for interoperable learning technology, and have developed a well adopted specification (IMS, 2004). The IMS question & test interoperability (QTI) specification provides an interoperable standard for describing questions and tests using extensible mark-up language<sup>1</sup> (XML) (IMS, 2000). The QTI specification is broken down into multiple subspecifications. Two of significance to the research herein are the *assessment, sections and items* (ASI) and the *results reporting* (RR) bindings. The ASI binding is used to describe the materials presented to the student, such as which questions, called *items*, form part of an assessment, how they are marked, how scores are aggregated, and so forth. The RR binding is responsible for describing students' results following completion of the marking process.

A major focus of the design for the QTI to date has been to support the interoperability of online quiz systems. These systems are typically fully automated and require little human intervention. Thus, the QTI lacks specific functionality for online systems providing student assessment that relies heavily on human intervention and critiquing. By enhancing the IMS QTI specification to better support CAM, tools can become interoperable, such that assessment materials can be exchanged between CAM systems in the same way as quiz question banks can between online quiz systems. The research presented in this paper introduces the QTICAM specification addressing the shortcomings of the IMS QTI in support of CAM.

### QTI COMPUTER ASSISTED MARKING SPECIFICATION

The QTI Computer Assisted Marking (QTICAM) specification has been designed as an extension to the IMS QTI to address the lack of

support for human intervention and critiquing. Its architecture ensures it remains backward compatible with the existing QTI specification. This ensures existing QTI XML documents can be validated against QTICAM. Furthermore, the QTICAM specification allows a mixture of automatic and manually marked items within the same assessment. The QTICAM provides improvements to both the ASI binding and RR binding as outlined in the following sections. A more complete description for the IMS QTI ASI (IMS, 2002a), and the IMS QTI RR (IMS, 2002b) can be accessed from the IMS Web site (<http://www.imsglobal.org>).

#### Mark Increments

The QTI provides scoring variables to track the marks associated with an assessment question. These scoring variables can be aggregated in various ways to derive a total score for the students' work. For example, the XML:

```
<decvar varname="SCORE"
      vartype="Integer"
      minvalue="0"
      maxvalue="10">
```

declares a variable with `<decvar>`<sup>2</sup> called SCORE to store a result. In this case, the result is restricted to a whole number (decimal) between and inclusive of the values 0 and 10.

This current format, while dictating some boundaries for a marker, does not restrict the marker from using their own part-marking scheme between the minimum and maximum values. The QTICAM provides the increment attribute to address this issue. For example, if the previous result should only be marked in increments of 2, the XML would be:

```
<decvar varname="SCORE"
      vartype="Integer"
      minvalue="0"
      maxvalue="10"
      increment="2">
```

This enhancement provides two advantages. Firstly, it improves the consistency in marks within a marking team, ensuring the markers

adhere to the scoring criteria, and secondly, it provides clearer instructions to an electronic marking tool as to what values it can allow as legal scores for a particular question.

### Manual Marker Rubrics

In addition to expressing the response processing of an item in machine terms, the QTICAM also supports response processing for human interpretation via a marking rubric. The `<interpretvar>`<sup>3</sup> element structure from the QTI ASI has been reused to describe such marking rubrics within the QTICAM ASI. For each `<interpretvar>` element, there is a matching scoring variable. The scoring variable is used to track the performance of the student against its rubric within the `<interpretvar>` element. There are no facilities for recording rubrics within the QTI RR for the marker. Therefore, an `<interpretscore>` element has been included in the QTICAM RR binding. This is demonstrated in Listing 1, along with its scoring variable SCORE.

The contents of the `<interpretscore>` element structure are derived from the `<interpretvar>` element of the ASI binding. The `varname` attribute defines the scoring variable SCORE with which the `<interpretscore>` rubric is associated. This is illustrated at the bottom of Listing 1 using the `<score>`<sup>4</sup> element, highlighted in bold. The example is a marking rubric for an IT-related short response question. Students are asked to briefly compare flat and hierarchical directory structures provided by network operating systems.

### Recording the Marker

Typically, the QTI is used to describe objective tests that will be marked by computer. With manual marking, it is necessary to record the identity of the marker for quality control. The allocation of student assessments among a group of markers can vary. For example, assessments can be allocated by student or by individual questions. The QTICAM therefore requires the ability to record the marker of each individual item. Thus, using QTICAM RR XML achieves this:

```
<manualscorer>
  <name>Damien Clark</name>
  <generic_identifier>
    <identifier_string>clarkd</identifier_string>
  </generic_identifier>
</manualscorer>
```

The `<manualscorer>` element content reuses the existing `<name>`, `<generic_identifier>`, and `<identifier_string>` elements of the QTI RR specification, which are currently used to describe the student. If an item has not yet been marked, there will be no `<manualscorer>` element structure, or its contents will be empty.

Currently, the QTICAM does not support the recording of multiple markers. Such an instance might occur in a peer revision process where several markers are assigned the task of providing a score for the same item. The authors recognise the need for this feature and expect to implement it in future revisions.

### Recording Marker Feedback and Marks

The QTI RR binding provides support for the `<feedback_displayed>` element structure which identifies feedback already displayed to the student, as a result of automated marking. This feedback is fixed and prescribed in the ASI XML when the item is conceived. This further illustrates the focus of the QTI on automated marking systems. It is not possible for the item author to foresee all potential errors made by students, and therefore it is necessary to provide support for feedback not prescribed within the item definition (QTI ASI). To support this function, QTICAM includes the `<manualfeedback>` container element. All feedback and marks are stored within this structure, as demonstrated in Listing 2.

Within `<manualfeedback>` are `<scorefeedback>` elements. Each `<scorefeedback>` can contain a feedback comment (`<comment>`), a mark (`<score_value>`) or both. Each `<scorefeedback>` is associated one-to-one with a scoring variable through the `varname` attribute. This provides an import linkage. It allows a comment or mark to be associated with a specific rubric (`<interpretscore>`). Furthermore, each `<scorefeedback>`

*Listing 1. Manual-marker Rubric (QTICAM RR)*

```

<interpretscore varname="SCORE">
  <material label="solution">
    <matemtext>
      A hierarchical directory structure is considered superior for enterprise
      networking.
    </matemtext>
    <matbreak/>
    <matemtext>
      A flat directory structure is slower and less efficient than a hierarchical
      directory structure.
    </matemtext>
    <matbreak/>
    <matemtext>
      It is much harder to find things in a flat directory structure than in a
      hierarchical directory structure.
    </matemtext>
  </material>
  <material>
    <mattext>
      One mark is allocated for each point above that the student has in their answer.
    </mattext>
  </material>
</interpretscore>
...
<outcomes>
  <score varname="SCORE">
    <score_value>0</score_value>
    <score_increment>1</score_increment>
    <score_min>0</score_min>
    <score_max>3</score_max>
  </score>
</outcomes>

```

is also uniquely identified within the scope of the item through the ident attribute. The ability to uniquely identify each comment or mark is described in the following section.

### Linking Feedback and Marks to the Student Response

Feedback on student assessment is an important element of the learning process (Dalziel, 2001). A novel approach to improving feedback presentation in CAM systems was investigated by Mason, Voit et al. (1999) where feedback is provided in-context of the students' submission, rather than summarised at the end. This is equivalent to the way a marker would assess a

paper-based submission, providing comments and marks in proximity of the passages being addressed. This is achieved in the QTICAM, as illustrated in Listing 3.

The solution provided by the student already stored within the QTI RR <response\_value> element is copied verbatim into the <taggedresponse> element. Next, passages of the student's response are tagged with the <tagresponse> element. Recall from Listing 2 each <scorefeedback> element had an ident attribute. Listing 3 shows the linkage of this ident attribute with the <tagresponse> element's ident attribute. This linkage is how a comment or

mark is associated in-context with the student's response. Therefore, the comment:

One output line transmits the data and the other transmits the complement of the signal

from Listing 2 is associated with the student passage

while RS-422a has two data output lines

from Listing 3.

This `<taggedresponse>` feedback can be presented to the student in various ways. For example, if presented in a Web-browser, the material within a `<tagresponse>` element could be a hyperlink to a popup window which displays the comment or mark. Alternately, a mouseover javascript event could present the comment or mark when the student places their mouse over the `<tagresponse>` area. If the feedback is to be printed, the comments or marks could be placed at the start or end of the underlined `<tagresponse>` material. How the material is presented is up to the implementer. The QTICAM ensures comments or marks are provided in-context of the student's solution.

### Recording Question Content Presented to the Student

The QTI RR binding does not include support for recording the question material that was presented to the student in completion of an item. To support the manual marking process, it is advantageous for the marker to see exactly what was presented to the student. This provides complete context for the student's solution. Furthermore, it is also necessary where parameterised questions are implemented (Clark, 2004). The QTICAM RR binding provides the `<material_presented>` element. This element should contain all the material that was presented to the student when they attempted the question, in HTML format. An example of the `<material_presented>` element looks like:

```
<material_presented>
<![CDATA[
<p>In your own words briefly compare
flat and hierarchical directory structures
provided by NOS.</p>
]]>
</material_presented>
```

Use of a CDATA<sup>5</sup> node is recommended to quote all HTML elements within the `<material_presented>` element as illustrated. This material can be presented to the marker when marking the students' solutions.

### Recording a Model Solution for an Item

The QTI RR binding provides support for recording the solution to an item through the `<correct_response>` element. This element is designed to identify a selectable choice or a model answer. Unfortunately, this element provides for only a textual value with no formatting. To improve readability for the manual marker, the `<solutionmaterial>` element is provided in the QTICAM RR binding. The `<solutionmaterial>` element is illustrated in Listing 4.

The `<solutionmaterial>` element incorporates the `<material>`<sup>6</sup> element used throughout the QTI specification to provide basic formatting of material for presentation. This allows the question author to provide a model solution to an item with basic formatting. The solution shown in Listing 4 is for a C programming item.

### QTICAM Implementation

The design of the QTICAM is implementation independent, meaning it does not constrain or dictate how a CAM tool should be implemented. It provides the supporting data model of how material from a testing system should be exchanged for marking. Therefore, an implementation of QTICAM could be written in various languages such as Java, Perl, or C++. Furthermore, a CAM tool could be implemented as an online or off-line application. For example, an online marking tool would maintain a connection with a network server and exchange QTICAMXML as required during marking. In an off-line environment, the marking tool would download large batches of

Listing 2. Recording marker feedback and marks (QTICAM RR)

```

<manualfeedback>
  <scorefeedback varname="SCORE" ident="1">
    <score_value>0.5</score_value>
  </scorefeedback>
  <scorefeedback varname="SCORE" ident="2">
    <score_value>0.5</score_value>
  </scorefeedback>
  <scorefeedback varname="SCORE" ident="3">
    <score_value>0.5</score_value>
  </scorefeedback>
  <scorefeedback varname="SCORE" ident="4">
    <score_value>1</score_value>
  </scorefeedback>
  <scorefeedback varname="SCORE" ident="5">
    <score_value>0.5</score_value>
  </scorefeedback>
  <scorefeedback varname="SCORE" ident="6">
    <comment>
      One output line transmits the data and the other transmits the complement of
      the
      signal.
    </comment>
    <score_value>0.5</score_value>
  </scorefeedback>
  <scorefeedback ident="7">
    <comment>
      Refer to the model solution for other factors you have not considered.
    </comment>
  </scorefeedback>
</manualfeedback>

```

Listing 3. In-context feedback of a student's response (QTICAM RR)

```

<taggedresponses>
  <!--The taggedresponse is the same as response_value (below) except tagresponse
  elements tag parts of it. These will be highlighted in some way when presented back to
  the candidate, and the feedback assigned will be shown (perhaps through mouseover or
  in another window)-->
  <taggedresponse ident_ref="CommQ2"><tagresponse ident="1">RS-232 has a slow
  data rate of 19.6 kbps.</tagresponse>
  <tagresponse ident="2">It is also only capable of distances up to 15 metres.</
  tagresponse>
  <tagresponse ident="3">RS-422a is capable of much faster transfers.</tagresponse>
  <tagresponse ident="4">RS-232 is unbalanced, while RS-422a is balanced.</
  tagresponse>
  <tagresponse ident="5">RS-232 has one signal wire</tagresponse>, <tagresponse
  ident="6">while RS-422a has two data output lines.</tagresponse>
  </taggedresponse>
</taggedresponses>

```

Listing 4. Record of the model solution for an item (QTICAM RR binding)

```

<solutionmaterial>
<material label="solution">
<mattext texttype="text/html" xml:space="preserve"><![CDATA[
<pre>
void replaceAll(char *aString, char *c1, char c2)
{
char *ptr;
ptr = aString;

while(*ptr != '\0')
{
if (*ptr = c1)
*ptr = *c2;
}
}
}
</pre>
]]></mattext>
</material>
</solutionmaterial>

```

QTICAM XML assessments. This could then be taken off-line during the marking process. Off-line implementation is of particular benefit to those with poor bandwidth such as analogue modem users, or for those with a roaming laptop. Alternately, a hybrid approach could be implemented where the marking tool supports both online and off-line operation.

The following section introduces the computer assisted marking prototype (CAMP), which demonstrates the use of the QTICAM specification.

### CAMP: PROTOTYPE MARKING TOOL

To demonstrate the QTICAM specification at work, the CAMP system has been developed. CAMP is a CAM tool implemented in Java. It is currently a prototype and not yet optimised for complete usability. However, it demonstrates the features of the QTICAM specification. CAMP makes use of the XML document object model (DOM) application programming interface (API)<sup>7</sup> to manipulate the QTICAM RR XML

containing the material that is to be marked. It can load multiple RR XML files, which it stores in memory. As an item is marked, the changes are kept in memory. Once the marker clicks the *save* button, moves onto another item, or otherwise closes the application down, the changes in memory are written to their respective XML file.

The CAMP tool supports the following functions:

- The ability to open multiple QTICAM RR XML documents and display a hierarchical tree structure, which summarises all items broken down into sections and student assessments.
- For each item loaded, it displays:
  - the material presented to the student;
  - the student's submission/s;
  - an optional model solution;
  - all the marking rubrics;
  - the student score for the item;
  - the student score for the assessment; and

- the student and marker's names.
- The ability for the marker to tag passages of the student's solution and attach feedback with a comment or mark.
- The modification of the comments and marks by clicking on an existing tagged passage.
- The deletion of existing comments and marks by clicking on an existing tagged passage.
- The saving of changes back to the XML file during the marking process.
- The flagging of an item as marked when marking is complete.

Automatic aggregation of marks is supported, totaling scoring variables for rubrics and item, section and assessment scores. Figure 1 illustrates the process of assigning feedback to a student's solution using CAMP.

This figure highlights the functionality provided by the QTICAM: (a) the assessment question; (b) the marking rubric; (c) the student's assessable answer where the marker has highlighted the passage more manageable for feedback, before clicking the Add Feedback button to present the feedback dialog (d). The dialog allows the marker to assign only a legitimate mark (0.5) within the bounds for the item and a comment: *Each part is more manageable than the whole*. Placing the mouse over the tagged passage more manageable in (c) will display (e), a popup window showing the recorded feedback for that passage; and (f) The total score of the item and Fred Smith's assessment score before the 0.5 mark was assigned.

To elaborate further, Figure 1 shows that the marker has highlighted the passage more manageable from the student's solution. To open the dialog box shown in Figure 1(d), the marker clicks the *Add Feedback* button. This dialog allows the marker to select the rubric to which their comment or mark is associated. On selecting the required rubric, the marker can only enter a mark that meets the constraints of the rubric. For example, the marker cannot

assign a mark that would push the total for the rubric beyond its upper or lower limits defined in the QTICAM. In this case, the rubric score has been specified with:

```
<decvar varname="SCORE"
  vartype="Decimal"
  minvalue="0"
  maxvalue="3"
  increment="0.5">
```

It restricts the assigned mark to values between 0 and 3 with increments of 0.5. This improves consistency in the marking and makes it quicker for the marker to select a mark. The dialog also contains a list of comments (*Feedback History*) made previously by this marker for the same item answered by other students. This helps with consistency in feedback and efficiency in allowing the marker to reuse comments. On selecting a comment from the dropdown list, it is placed in the *Feedback* text area at the bottom of the dialog. The marker can choose to customise the comment if they wish. Alternately, the marker can create a new comment by typing directly into this empty text area.

On feedback completion, the associated passage from the student's solution (originally highlighted by the marker) appears underlined to indicate it has feedback associated with it, and the QTICAM RR XML for this item has changed, as illustrated in Listing 5.

The code presented in bold illustrates the changes made to the XML file once a marker has provided feedback using CAMP.

When item marking is complete, the *Completed* tick box at the bottom of Figure 1 is selected. By forcing the marker to make the conscious decision to flag an item as complete, this ensures items are not overlooked, when for example, a marker moves from one item to another comparing different students' solutions. When an item is flagged as unmarked, it is represented in QTICAM RR XML as:

Figure 1. CAMP: Selecting passage for feedback

Fred Smith (q91234567): CommQ1 (Marker: Damien Clark)

**Question:**

Why was the OSI Reference Model developed? Why are the layers of the OSI so important?

(a)

---

The OSI Model was developed to provide open interconnection between heterogenous systems. It divides the task of network communication into separate components. This makes the communication process more manageable. It also allows different functions to be implemented by separate entities and yet still remain interoperable.

(c)

(Feedback)

The OSI Reference Model was developed to divide the task of designing a network into smaller parts. Each part is independent of each other. Each part is more manageable than the whole. Each layer is a part of the whole. Layers are important because they allow the network to be divided into distinct functions. Different groups can develop each of the functions independently. Each layer becomes more manageable.

(b)

A half mark is allocated for each highlighted point above that the student has in their answer.

**Student Response**  
more manageable

**Marker Feedback**  
Each part is more manageable than the whole.

(e)

**Score**  
Score: 0.5

**Provide Feedback**

Rubric: SCORE

Score: -1.0 ▾

Feedback History: -0.5

Feedback: 0.0

(d)

---

Each part is more manageable than the whole.

CommQ1

Model Solution SCORE [1.0/3.0]

Completed

Item Score: 1.0/3.0    Student Score: 4.5/103.0

(f)

Listing 5: QTICAM RR XML: Changes to XML after adding feedback

```

<taggedresponses>
  <taggedresponse ident_ref="CommQ1">The OSI Model was developed to provide open inter-
  connection between heterogeneous systems. <tagresponse ident="1">It divides the task of
  network communication into separate components.</tagresponse> This makes the commu-
  nication process <tagresponse ident="2">more manageable</tagresponse>. <tagresponse
  ident="3">It also allows different functions to be implemented by separate entities and yet still
  remain interoperable.</tagresponse>

  <tagresponse ident="4"/>
</taggedresponse>
</taggedresponses>
<manualfeedback>
  <scorefeedback ident="1" varname="SCORE">
    <score_value>0.5</score_value>
  </scorefeedback>
  <scorefeedback ident="3" varname="SCORE">
    <score_value>0.5</score_value>
  </scorefeedback>
  <scorefeedback ident="4" varname="SCORE">
    <comment>Other points to consider include that each layer is independent and that each
    part is more manageable than the whole. The layers are also distinct functions. Good effort.</
    comment>
  </scorefeedback>
  <scorefeedback ident="2" varname="SCORE">
    <comment>Each part is more manageable than the whole.</comment>
    <score_value>0.5</score_value>
  </scorefeedback>
</manualfeedback>

```

```

<manualscoring>
  <status>
    <status_value>Unmarked</status_value>
  </status>
</manualscoring>

```

When a tick is placed in the *Completed* tick box, the XML is changed to:

```

<manualscoring>
  <status>
    <status_value>Marked</status_value>
  </status>
</manualscoring>

```

The marker navigation window, as illustrated in Figure 2(a), shows that question CommQ1 of Section Part A has now been marked.

This window gives a hierarchical view of all student assessments that have been loaded into memory. Once an entire branch of the hi-

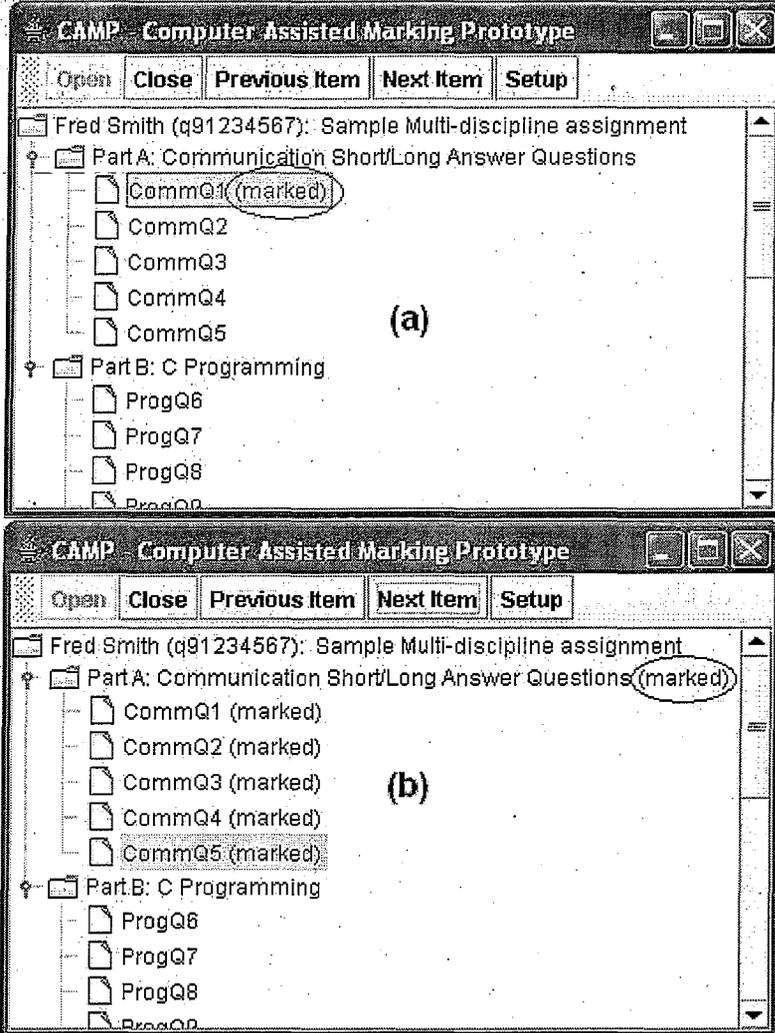
erarchy has been completely marked, its parent branch will also be flagged as marked. This is demonstrated in Figure 2(b).

When section *Part B* is marked, this will flag the entire assessment *Sample Multi-discipline assignment* for *Fred Smith* as marked, in the same manner. This allows the marker to see at a glance what remains to be marked from their allocation of student assessment.

## CONCLUSION

QTICAM is an enhancement of the IMS QTI specification and provides support for interoperable computer assisted marking. Its functionality has been illustrated via the demonstration of CAMP. Features of the QTICAM include: support for limiting mark increments, inclusion of human readable marking rubrics, ability to record the marker for each marked item,

Figure 2. CAMP: Navigation window flagging marked items



recording manual marker feedback including comments and marks, linking marker feedback to passages of the students' solutions, recording the material presented to the student in the results report, and the ability to record formatted model solutions for items.

One of the main benefits for markers in the use of CAM software is increased productivity through automation of repetitive mechanical tasks (Joy & Luck, 1998). Such benefits include: automatic collation of marks at the item, section, and assessment levels, and the ability to easily reuse feedback comments by selecting from a list. Another major benefit to CAM software is improved quality. For example, typically a marker will, after completion of marking, add the marks assigned and record the total on a marking sheet. This manual process introduces a high risk of error during the addition and transcription of the marks. Through CAM, marks can be collated and recorded automatically, eliminating this quality issue. Other benefits to CAM include:

- Improved marking consistency: providing constraints on scoring variables ensures the markers assign marks consistently within the scope of the marking rubric
- Manual handling of results is eliminated: results from student assessments can be automatically uploaded into a LMS reducing staff workload and errors
- Improved marking feedback: permitting the marker to associate feedback with passages of the student's solution allows the student to interpret the feedback in the context of their own work (Mason, Woit et al., 1999)
- Potential to automate correction of marking errors across large assessment collections

The QTICAM specification currently adds essential support to the QTI for computer assisted marking. Future development will see the inclusion of advanced features that will:

- Automate late submission penalty application
- Share feedback between multiple markers
- Classify markers' comments for later analysis
- Automate marking moderation

With the adoption of an interoperable CAM specification such as QTICAM, interoperable CAM applications can be a reality.

## REFERENCES

- Baillie-de Byl, P. (2004). An online assistant for remote, distributed critiquing of electronically submitted assessment. *Educational Technology and Society*, 7(1), 29-41.
- Benford, S.D., Burke, E.K., Foxley, E., Higgins, C.A. (1994). A courseware system for the assessment and administration of computer programming courses in higher education. *Complex Learning in Computer Environments (CLCE'94)*
- Bloom, B.S. (1956). *Taxonomy of educational objectives handbook 1: Cognitive domain*. New York: Longman, Green, & Co.
- Clark, D. (2004). Enhancing the IMS Q&TI specification by adding support for dynamically generated parameterised quizzes (p. 230). Toowoomba, University of Southern Queensland: Department of Mathematics and Computing.
- Creative-Technology (2005, January 16). Program features. Retrieved March 8, 2007, from <http://www.cict.co.uk/software/markin/index.htm>.
- Dalziel, J. (2000). Integrating CAA with textbooks and question banks: Options for enhancing learning. *Computer Aided Assessment (CAA2000)*, Leicestershire, UK.
- Dalziel, J. (2001). Enhancing Web-based learning with CAA: Pedagogical and technical considerations. *Computer Aided Assessment (CAA2001)*, Leicestershire, UK.
- Darbyshire, P. (2000). Distributed web-based assignment management. In A. Aggarwal (Ed.), *Web based learning and teaching technologies: Opportunities and challenges* (pp. 198-215). Idea Group.

- Davies, P. (2001). Computer aided assessment must be more than multiple-choice tests for it to be academically credible? *Computer Aided Assessment (CAA2001)*, Leicestershire, UK.
- English, J., & Siviter, P. (2000). Experience with an automatically assessed course. In *Proceedings of the Conference on Integrating Technology into Computer Science Education (ITiCSE)* (pp. 168-171). Helsinki, Finland.
- Gayo, J.E.L., Morales, J.M.G., Fernandex, A.M.A., Sagastegui, H.C. (2003). A generic e-learning multiparadigm programming language system: IDEFIX project. *Technical Symposium on Computer Science Education* (pp. 391-395). Reno, NV: ACM Press.
- Hill, T.G. (2003). MEAGER: Microsoft Excel automated grader. *The Journal of Computing in Small Colleges*, 18(6), 151-164.
- Huizinga, D. (2001). Identifying topics for instructional improvement through on-line tracking of programming assessment. In *Proceedings of the Conference on Integrating Technology into Computer Science Education (ITiCSE)* (pp. 129-132). Canterbury, UK.
- IMS. (2000). IMS question & test interoperability specification: A review. IMS Global Learning Consortium. Retrieved March 8, 2007, from <http://www.imsproject.org/question/whitepaper.pdf>
- IMS. (2002a). IMS question & test interoperability: ASI XML binding specification. IMS Global Learning Consortium. Retrieved March 8, 2007, from <http://www.imsproject.org>
- IMS. (2002b). IMS question & test interoperability: Results reporting XML binding specification. IMS Global Learning Consortium. Retrieved March 8, 2007, from <http://www.imsproject.org>
- IMS. (2004). Directory of products and organisations supporting IMS specifications. IMS Global. Retrieved March 8, 2007, from <http://www.imsproject.org/direct/directory.cfm>
- IMS. (2005). IMS Global Learning Consortium. Retrieved March 8, 2007, from <http://www.imsproject.org>
- Jackson, D. (2000). A semi-automated approach to online assessment. In *Proceedings of the Conference on Integrating Technology into Computer Science Education (ITiCSE)* (pp. 164-167). Helsinki, Finland.
- Jacobsen, M., & Kremer, R. (2000). Online testing and grading using WebCT in computer science. In *Proceedings of the World Conference on the WWW and Internet* (pp. 263-268).
- Jefferies, P., Constable, I., et al. (2000). Computer aided assessment using WebCT. *Computer Aided Assessment (CAA2000)*, Leicestershire, UK.
- Jones, D., & Behrens, S. (2003). Online assignment management: An evolutionary tale. In *Proceedings of the Hawaii International Conference on System Sciences*, Waikoloa Village.
- Jones, D., & Jamieson, B. (1997). Three generations of online assignment management. In *Proceedings of the Australian Society for Computers in Learning in Tertiary Education Conference* (pp. 317-323). Perth, Australia.
- Joy, M., & Luck, M. (1998). Effective electronic marking for on-line assessment. In *Proceedings of the Conference on Integrating Technology into Computer Science Education (ITiCSE)* (pp. 134-138). Dublin, Ireland.
- Mason, D.V., & Voit, D.M. (1999). Providing mark-up and feedback to students with online marking. *SIGCSE Technical Symposium on Computer Science Education* (pp. 3-6). New Orleans, LA.
- Mason, D.V., Voit, D., Abdullah, A., Barakat, Ho., Pires, C., D'Souza, M. (1999). Web-based evaluation for the convenience of students, markers, and faculty. In *Proceedings of the North American Web Conference*, Fredericton, Canada.
- Merat, F.L., & Chung, D. (1997). World Wide Web approach to teaching microprocessors. In *Proceedings of the Frontiers in Education Conference* (pp. 838-841). Stipes Publishing.
- Muldner, M., & Currie, D. (1999). Techniques to implement high speed scalable dynamic on-line systems. In *Proceedings of the World Conference on the WWW and Internet* (pp. 782-787).
- Pain, D., & Heron, J.L. (2003). WebCT and online assessment: The best thing since SOAP? *Journal*

- of *International Forum of Educational Technology & Society*, 6(2), 62-71.
- Palmer, J., Williams, R., Dreher, H. (2002). Automated essay grading system applied to a first year university subject. *Informing Science*, 1222-1229.
- Peat, M., Franklin, S., & Lewis, A. (2001). A review of the use of online self-assessment modules to enhance student learning outcomes: Are they worth the effort of production. In *Proceedings of the ASCILITE2001* (pp. 137-140). Melbourne, Australia.
- Preston, J., & Shackelford, R. (1999). Improving online assessment: An investigation of existing marking methodologies. In *Proceedings of the Conference on Integrating Technology into Computer Science Education (ITiCSE)* (pp. 29-32). Crocow, Poland.
- Price, B., & Petre, M. (1997). Teaching programming through paperless assignments: An empirical evaluation of instructor feedback. In *Proceedings of the Conference on Integrating Technology into Computer Science Education (ITiCSE)* (pp. 94-99). Uppsala, Sweden.
- Roantree, M., & Keyes, T.E. (1998). Automated collection of coursework using the Web. In *Proceedings of the Conference on Integrating Technology into Computer Science Education (ITiCSE)* (pp. 206-208). Dublin, Ireland.
- Sheard, J., & Carbone, A. (2000). Providing support for self-managed learning? In *Proceedings of the World Conference on the WWW and Internet 2000* (pp. 482-488).
- Thomas, P. (2000). Reducing the distance in distance education. *Computer Aided Assessment (CAA2000)*, Leicestershire, UK.
- Trivedi, A., Kar, D.C., Patterson-McNeill, H. (2003). Automatic assignment management and peer evaluation. *The Journal of Computing in Small Colleges*, 18(4), 30-37.
- White, J. (2000). Online testing: The dog sat on my keyboard. In *Proceedings of the International Conference on Technology in Collegiate Mathematics*, Atlanta, GA.
- Woit, D., & Mason, D. (2000). Enhancing student learning through online quizzes. *SIGCSE Technical Symposium on Computer Science Education* (pp. 367-371). Austin, TX.

## ENDNOTES

- 1 Readers not familiar with XML are directed to read the following online resources: <http://www.xml.com>, <http://xml.coverpages.org/xml.html>, <http://www.w3.org/XML/>, <http://www.xml.org>.
- 2 The <devar> element is used within the QTI ASI specification for declaring a scoring variable. It allows the question author to define attributes for a scoring variable such as minimum, maximum, and default values.
- 3 The <interpretvar> element describes how to interpret the meaning of scores assigned to scoring variables.
- 4 <score> is used within the QTI RR binding to record the score achieved by a student as defined by the <devar> element of the QTI ASI.
- 5 A CDATA node is a quoting mechanism within XML syntax to allow the special meaning of other XML characters to be escaped as part of an XML document.
- 6 The <material> element provides a container object for any content to be displayed. It allows various data types such as plain or emphasised text, images, audio, videos, or applets.
- 7 The XML DOM API is a standard platform independent programming interface for manipulating the content of XML documents in computer memory.

*Damien Clark commenced his academic research career in 2003 developing a parameterisation enhancement to the IMS QTI, resulting in an honours equivalent thesis. Clark's paper published in this edition of JDET is his first for an international journal. He completed a bachelor's degree in computer science from Central Queensland University, Australia in 1995. He also holds a master's degree in computer science from the University of Southern Queensland, Australia. During his career, he worked as a UNIX systems administrator before taking a position as lecturer at Central Queensland University in 2002. He teaches system administration, computer networking, and information security.*

*Penny Baillie-de Byl has been researching in the area of online assessment management systems, artificial intelligence and computer games programming since 1995. She has written a number of international conference papers, journal papers, book chapters and two books in these areas. During her career, Dr. Baillie-de Byl has consulted as a computer programmer, computer games designer, website engineer and artificial intelligence designer. Dr. Baillie-de Byl currently works as a senior lecturer in computer graphics and computer games programming and manages a games research and development laboratory at the University of Southern Queensland, Australia.*