Accessibility Issues in Two Specifications for e-Learning Tests: IMS QTI 1.2 and IMS QTI 2.0

Christophe Strobbe

Katholieke Universiteit Leuven – Dept. of Electrical Engineering, Kasteelpark Arenberg 10, B-3001 Heverlee – Leuven (Belgium)
christophe.strobbe@esat.kuleuven.be

Abstract. The IMS Global Learning Consortium developed the QTI (Question and Test Interoperability) specification to allow the exchange of question and test data, and their corresponding result reports, between learning systems. QTI 1.2 had some accessibility issues, as the VISUAL project discovered when transforming QTI tests into accessible HTML and voice user interfaces. Some problems were due to an insufficient mechanism to specify alternative text, other problems were due to the ambiguity of the intent of certain interaction types. QTI 2.0 solved the issue of alternative text, but the ambiguity with regard to the intent of interaction types was not sufficiently addressed.

Introduction

In Web accessibility, much attention is paid to accessibility guidelines for content, user agents and authoring tools, to software that checks content against accessibility guidelines, to the accessibility of user agents to assistive technologies, and to the development of assistive technologies and adapted devices. However, it is also important to review technical specifications of content types so as to check for accessibility features. If specifications are not reviewed for inherent accessibility issues, guidelines and tools can only work at the fringes of accessibility. The IMS Global Learning Consortium develops specifications for "distributed learning", most of which are based on XML to facilitate the exchange of content, metadata etcetera between learning systems. One of these is IMS Question and Test Interoperability (QTI), a specification for tests and assessments, from single test questions ("items") to sets of questions ("sections" and "assessments"), and interactive content. In addition to defining test questions and interaction, QTI files also contain "response processing" and grading information. QTI is an XML-based format that can be used to exchange tests and assessments between systems that use a proprietary format internally, but it can also used as the native format for tests in an e-learning system. At the time of writing, the current version is 2.0, which presents a thorough overhaul when compared to its predecessor, QTI 1.2. Version 2.1 is currently undergoing review.

QTI 1.2

QTI 1.2 Response Types

The QTI 1.2 specification consist of three documents, one if which is the *ASI Best Practice and Implementation Guide* [11]. This document lists 20 example basic item types. For each type, the guide provides a visual rendering and the corresponding XML code. For most of these types, it is possible to create an equivalent for use in a traditional class room, even though some of them cannot be realized with only pen and paper, for example standard multiple choice with audio (§ 4.1.4). The two types with sliders (multiple choice with slider rendering, § 4.1.8, and numerical entry with slider, § 4.4.3) are more typical of human-computer interfaces. Some item types, namely standard short answer (§ 4.3.3) and open ended questions with fill-in-blank (§ 4.3.1, e.g. "Name a renaissance playwright"), cannot be scored automatically with current technology.

The *ASI Best Practice & Implementation Guide* only provides a visual rendering of each of the items; the specification does not consider other renderings, in spite of the emergence of voice user interfaces, research on multimodal interfaces and the growing awareness of accessibility and inclusive design. From a pedagogic point of view, a number of examples would benefit from more extended feedback. The CETIS Assessment SIG worked on the examples from the specification and modified the XML where this was necessary or desirable[1].

The QTI 1.2 specification is agnostic with regard to the technology that is used to present the questions. Some online demonstrators use Flash for both the question text and the presentation of choices/input, others use a combination of HTML and Flash or Java applets. Flash and Java have accessibility issues, so the presentation implemented in the VISUAL project relied on HTML where possible and only resorted to an embedded interface where necessary. Nine out of the twenty examples in the specification can be handled with "pure" HTML, whereas the other eleven types require the use of embedded interfaces (at least, the examples in the *ASI Best Practice & Implementation Guide* suggest embedded interfaces).

The following types can be handled with pure HTML (paragraph numbers identify the sections in the *ASI Best Practice & Implementation Guide*):

- standard true/false (text: § 4.1.1);
- standard multiple choice (with text: § 4.1.2; with images: § 4.1.3);
- standard multiple response (text: § 4.1.5);
- standard fill in blank (text: § 4.3.1; decimal: § 4.4.1; integer: § 4.4.2);
- standard multiple fill-in-blank (text: § 4.3.2);
- standard short answer (text: § 4.3.3).

The following types require the use of embedded interfaces or DHTML:

- standard multiple choice (audio: § 4.1.4; with image hot spot rendering: § 4.1.6; with slider rendering: § 4.1.8);
- multiple response with image hot spot rendering (§ 4.1.7);
- standard order objects (text: § 4.1.9; image: § 4.1.10);

[1] These examples are no longer available on the Web.

- connect-the-points (logical identifier: § 4.1.11; XY response type: § 4.2.2);
- standard image hot spot (§ 4.2.1);
- numerical entry with slider (§ 4.4.3);
- drag-and-drop (images: § 4.5.1).

The above distinction is relevant to accessibility, because embedded interfaces have their own set of accessibility issues. For the first nine types, accessibility can be achieved by making sure that the HTML output conforms to the Web Content Accessibility Guidelines. However, this is not always possible with the QTI 1.2, especially with the examples provided in the specification. The following sections discuss these basic accessibility issues. These issues are excerpted from a report that was sent to the IMS Global Learning Consortium in June 2004.

QTI 1.2 Accessibility Issues

Place-Holder Text for Edit Boxes

Several response types require edit boxes in HTML: fill-in-blank is rendered with `<input type="text" …>`, and standard short answer is rendered with `<textarea rows="20" cols="80"></textarea>`. Checkpoint 10.4 of the Web Content Accessibility Guidelines recommends: "Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas." User agents or assistive technologies that don't recognize empty form controls have become very rare (for example, JAWS 3.5 with Netscape 4.7.x), so it is probably no longer necessary to provide place-holder text. However, this was not the case when work on QTI began, and the QTI specification provided no information about this. All example basic item types use empty `response_label` elements, instead of using child elements in `response_label` for place-holding text.

Alternative text

The basic item type "standard multiple choice (images)" (§ 4.1.3) uses images; when rendering this item type in HTML, it is necessary to provide alternative text for accessibility.

The `altmaterial` element is an optional child element of the `material` element and is primarily intended to provide alternative language content in a single item. When used, there is one `altmaterial` element for each different language in the item; the `xml:lang` attribute indicates the language. The *ASI Information Model Specification* [12] explains that it is an element for "alternative content to be displayed in case the linked material cannot be rendered". A note explains that "this alternative material should not be of the same type as the original otherwise it too will not be rendered. Different versions should be used to support other languages or Accessibility options" (p. 42).

The following code sample adds alternative text to the first response label in one of the example from the *Best Practice and Implementation Guide* (mchc_ir_002.xml).

```
<response_label ident="A">
  <material>
    <matimage imagtype="image/gif"
```

```
        uri="mchc_ir_002_image1.gif"/>
      <altmaterial>
        <mattext>
          Round sign with white background, a red border…
        </mattext>
      </altmaterial>
    </material>
  </response_label>
```

The code sample shows that the relationship between the image and its alternative text is implicit. When the `material` element contains just one `matimage` followed by `altmaterial`, it is probably safe to assume that the `altmaterial` element contains alternative text for the image. However, there can be any number of `matimage` and `altmaterial` elements and there is no way to specify which `altmaterial`(s) go(es) with which `matimage`. This is different from the explicit links between media and alternative text (or between form fields and labels) that are used in HTML. For example,

- the alternative text for an image is provided as an attribute,
- the alternative text for an embedded object is provided in child elements within the `object` element,
- labels are explicitly linked with form controls by means of the `id` and `for` attributes.

The *Best Practice and Implementation Guide* should have stated more explicitly the importance of alternative text for accessibility. The DTD for QTI 1.2 should have made `altmaterial` and `mattext` required elements and should have specified a mechanism to "link" `altmaterial` with an image (or other media). In order to render an HTML `img` element with an empty `alt` attribute, implementers could leave the `mattext` element empty.

The same comments also apply to the use of other media (audio, video, …). The basic item type "standard multiple choice (audio)" (§ 4.1.4) uses sound files; when rendering this item type in HTML, it is necessary to provide alternative text for accessibility.

Accessibility and Test Validity

Alternative text in QTI would be necessary to allow "equivalent access". The *IMS Guidelines for Developing Accessible Learning Applications* state that *equivalent access* "provides the disabled user with content identical to that used by the non-disabled user", whereas *alternative access* "provides the disabled user with a learning activity that differs from the activity used by the non-disabled user" [2]. In the context of QTI, this distinction translates into "equivalent assessment" versus "alternative assessment". The challenge in alternative assessments is ensuring that one is assessing the same learning outcomes as with the original method; some assessments fail to do this [10]. Equivalent assessment should be provided whenever possible, but there may be tests where accessibility features conflict with validity constraints. Validity here refers to "[t]he degree to which accumulated evidence and theory support specific test scores entailed by proposed uses of a test" [1]. In some tests, accessibility features are essential for overcoming threats to validity, while in others, accessibility features can

actually pose threats to validity [2]. Suppose, for example, that an art history test is intended to assess the student's ability to distinguish between specific chamber music genres (string trio, quartet, and quintet) and asks: "Which of the following extracts is played by a string quartet?" Providing text alternatives (for the hard of hearing, as required by WCAG [3] and the XML Accessibility Guidelines [4]) would threaten the validity of the test. In this case, one would provide alternative assessment to learners with hearing impairments; IMS has developed *IMS AccessForAll Metadata* [6] and *IMS Learner Information Package Accessibility for LIP* [9] to enable the retrieval of alternative content.

Some cases in the *Best Practice and Implementation Guide*, however, are ambiguous with regard to the aim of the question, and this makes it impossible to decide whether equivalent or an alternative assessment should be chosen. The example of drag-and-drop interaction contains a drawing of the solar system with an empty text box below each planet, and asks the subject to place text markers with the names of the planets inside the relevant boxes. However, it is not clear what the outcome is intended to test: whether the learner knows the order of the planets, whether the learner can recognize the planets by their relative size and colour, or something else. The intent of the test would determine if and what kind of text description of the planets would be appropriate. However, the relationship of this ambiguity with accessibility is not discussed in the QTI specification.

QTI and Voice Interaction

In the European project VISUAL, QTI items were not only transformed into accessible HTML (with XSLT): there were also efforts to adapt several interaction types to a voice user interface. Interaction with a voice user interface is different from the interaction of sighted users with a Web interface.

1. HTML specifies a two-dimensional layout, whereas a voice user interface works purely in the time dimension.
2. HTML is displayed in whole-page units, whereas VoiceXML and similar languages describe dialogues, which are in turn presented in smaller units (steps, forms, prompts, …).
3. An HTML page can present the user with dozens of options; voice applications must limit the number of options at any step in the dialogue to ease the burden on the user's memory and to improve the performance of speech recognition.

Generating voice interaction from QTI items is more problematic than generating a visual rendering, for several reasons. Since the QTI 1.2 specification only considers visual renderings, any rendering that is not purely visual is of necessity a proprietary extension of the specification (i.e. it overrides the presentation that is defined in the QTI specification). Moreover, the rendering format is often related to the didactic purpose of the exercise, so it may be inappropriate or even impossible (e.g. image hot spot/image map) to 'override' the rendering format with a different user interface. This problem can be solved by providing alternative Items with rendering formats that are adapted to the needs of the user. Voice interaction adds complexity to the process of answering a question. If the answer to a question goes beyond a single choice or input, the user should at any time be able to review the answer he has built up before

completing it. Users should also be able to correct a partial answer before moving on: this adds another level of complexity. Voice interaction is not only more complex than accessible HTML: there is also a greater difference between the interaction for interactive content or for assessments. (The dialogues that were modelled in VISUAL only considered interactive content. Assessments require that the user be able to review an answer and, if necessary, to correct it. With certain types of questions, corrections are very hard to handle in a voice user interface; it may even be easier to input the answer again from scratch.) Also, certain types of feedback which users of visual interfaces get 'for free' (e.g. maximum number of characters for text input) are not relevant to or can hardly be implemented in a voice user interface.

The complexity of voice user interfaces calls for some clarifications in the QTI specification. For example, QTI 1.2 has metadata to define

- whether or not feedback is to be made available (`qmd_feedback_permitted`), and
- whether or not hints are to be made available (`qmd_hints_permitted`),

but no metadata to specify if a question may be repeated or not when the learner does not immediately understand the question. This type of repetition is a feature of human-to-human communication ("I beg your pardon?", "Could you repeat that, please?") and needs to be handled in voice user interfaces. This is probably too fundamental for voice interaction to provide metadata or other means to disable this.

QTI 2.0

IMS QTI 2.0 (see [8] for an overview) is a complete overhaul of the language and allows authors to define the same types of questions (and others) with much leaner code. It specifies many types of interactions that were previously defined as extensions of the language (for example drag-and-drop to order items) or that were previously not defined (for example file upload and drawing). It uses many elements from HTML/XHTML, which makes transformation to HTML more straightforward. Reusing features from an existing language is usually good practice, especially from a language with documented and widely supported accessibility features (see checkpoint 2.9 in [4]). QTI 2.0 also borrows HTML's `object` element to allow authors to define alternative media and alternative text that are explicitly associated with each other (a suggestion from the feedback mentioned above).

Reusing HTML elements solved an important accessibility issue, but the new specification does not address the question of rendering in other modalities than visual interaction. The examples in the QTI 2.0 Implementation Guide still only consider visual renderings and the naming of most question types suggests a rendering instead of a learning outcome. The introduction to the implementation guide states: "[The screen shots] are designed to illustrate how a system might implement the specification and are not designed to be prescriptive. *Other types of rendering are equally valid*." However, the document does not discuss non-visual renderings, and the names of many interaction types suggest visual renderings (for example, `hotspotInteraction` and `drawingInteraction`). If the intent of the test is related to a specific visual rendering, it is often impossible to define an

equivalent test without making the outcome invalid. However, for tests where the rendering is only a function of the learning system, QTI 2.0 does not define a mechanism that allows authors to specify that the rendering is not important. Moreover, the specification never discusses alternative renderings for question types that can be rendered in non-visual modalities.

QTI takes the opposite approach of W3C's XForms [5], the successor of HTML forms that will be integrated into XHTML 2.0. XForms are designed for accessibility because—among other reasons—they do not make assumptions about how forms will be rendered (for example visually or through voice interaction). When trying to convert the examples from the QTI 2.0 Implementation Guide into XForms, one would find that some of the interactions illustrated in the guide are ambiguous. The example for `associateInteraction` asks the learner to identify three pairs of rivals in a list of six Shakepearean characters. The illustration is a screen shot of a drag-and-drop rendering, and it is not clear if a rendering with edit boxes or selection lists would also be appropriate. In other words, it is not clear if the rendering is essential or whether one is allowed to render the question in other modalities.

Conclusions

The IMS specifications for Question and Test Interoperability, versions 1.2 and 2.0, have left certain aspects about the allowed renderings and the intent of questions undefined, and this leads to ambiguity with regard to how certain question types may be adapted for people with disabilities. If more guidance were provided on the relation between interaction types and the intent of the tests, it would be possible to create more equivalent assessments instead of always relying on alternative assessments. This would benefit people with disabilities directly, but it would also eliminate some of the overhead caused by alternative assessments and their associated metadata.

Acknowledgements

Part of this work (accessibility issues of QTI 1.2) was undertaken in the framework of the project VISUAL — IST-2001-32495 (October 2001-June 2004) — funded by the IST Programme of the European Commission.

References

1. AERA, APA & NCME: Standards for Educational and Psychological Testing. Washington DC: American Educational Research Association, American Psychological Associations, National Council on Measurement in Education (1999).

2. Barstow, C., McKell, M., Rothberg, M., Schmidt, C. (eds.): IMS Guidelines for Developing Accessible Learning Applications. White Paper. 27 June 2002. Available at: http://www.imsglobal.org/accessibility/accv1p0/imsacc_guidev1p0.html.
3. Chisholm, W., Vanderheiden, G., Jacobs, I. (eds.): Web Content Accessibility Guidelines 1.0 – W3C Recommendation 5-May-1999. Available at: http://www.w3.org/TR/WAI-WEBCONTENT
4. Dardailler, D., Palmer S. B., McCathieNevile C. (eds.): XML Accessibility Guidelines – W3C Working Draft 3 October 2002. Available at: http://www.w3.org/TR/2002/WD-xag-20021003
5. Dubinko, M., Klotz, L. L. Jr., Merrick, R., Raman T. V. (eds.): XForms 1.0 – W3C Recommendation 14 October 2003. Available at: http://www.w3.org/TR/2003/REC-xforms-20031014/
6. Jackl, A. (ed.): IMS AccessForAll Meta-data Best Practice and Implementation Guide. Version 1.0. Final Specification. 12 July 2004. Available at: http://www.imsglobal.org/acclip/acclipv1p0/imsacclip_bestv1p0.html
7. Lay, S. (ed.): IMS Question and Test Interoperability Implementation Guide. Version 2.0. Final specification. 24 January 2005. Available at: http://www.imsglobal.org/question/qti_v2p0/imsqti_implv2p0.html
8. Lay, S. (ed.): IMS Question and Test Interoperability Overview. Version 2.0. Final specification. 24 January 2005. Available at: http://www.imsglobal.org/question/qti_v2p0/imsqti_oviewv2p0.html
9. Norton, M., Treviranus, J. (eds.): IMS Learner Information Package Accessibility for LIP Best Practice and Implementation Guide. Version 1.0. Final Specification. 18 June 2003. Available at: http://www.imsglobal.org/acclip/acclipv1p0/imsacclip_bestv1p0.html
10. Sharpe, K., Earle, S.: Assessment, Disability and the Problem of Compensation. Assessment and Evaluation in Higher Education, 25. 2 (2000) 191-199.
11. Smythe, C., Shepherd, E., Brewer, L., Lay, S. (eds.): IMS Question and Test Interoperability: ASI Best Practice & Implementation Guide. Version 1.2. Final Specification. 11 February 2002. Available at: http://www.imsglobal.org/question/v1p2/imsqti_asi_bestv1p2.html.
12. Smythe, C., Shepherd, E., Brewer, L., Lay, S. (eds.): IMS Question and Test Interoperability: ASI Information Model Specification. Version 1.2. Final Specification. 11 February 2002. Available at: http://www.imsglobal.org/question/v1p2/imsqti_asi_infov1p2.html