

複雑な問題教材に対応した XML 記述方式 oqXML と応用システム

小島 一秀[†] 堀 一成[‡] 高橋 明[‡]

[†] 大阪外国語大学 情報処理センター

[‡] 大阪外国語大学 外国語学部

{kkojima, hori, takahasi}@osaka-gaidai.ac.jp

1 はじめに

現在の e-learning システムは選択問題やリスニング問題などある程度多様な問題形式を扱うことができる[1][2]。しかしながら、通常の紙媒体の試験問題のような対象知識の論理構造に沿った形式には対応せず、5 択問題や順序作成問題のような単純な形式に問題を押し込めている場合が多い。また、このような e-learning と紙媒体の間の問題形式の表現力の差が、e-learning と紙媒体の間での問題教材の共有を阻んでいる。

そこで、本稿では通常の紙媒体の問題集や試験で出題されるような複雑な問題形式に対応した問題記述方式として oqXML[3]を提案する。oqXML は、XML に準拠し、問題文や正解などの通常の情報に加えて、自動採点のための情報を記述することができる。oqXML における採点のための情報は、解答欄のグループ、完答要求、解答順序によって表現され、複雑な問題まで統一的に記述することができる。この oqXML とソフトウェア群により、紙媒体と e-learning の問題教材が一元化され、両者の区別のない問題教材の活用環境を実現している。また、本稿ではそれらを活用した大阪外国語大学の e-learning システムの開発事例についても述べる。

2 oq シリーズの体系

oq シリーズは次の 4 要素からなる (図 1)。まず、oq-producer によって問題データを作成する。

作成される問題データは oqXML で記述されている。作成された問題データを oq-player で動作させると e-learning の問題教材となり、oq-printer で処理すると紙媒体の問題集や試験用紙を作成できる。oqXML とソフトウェア群により紙媒体と e-learning の問題教材を区別することのない環境が実現されている[3]。



図 1 oq シリーズの体系

2.1 oq-producer

oq-producer は oqXML 形式で問題データを作成するためのソフトウェアで、Java で開発されている。また、e-learning 教材の生成機能も持ち、問題データに oq-player 本体や必要ファイルを付加して e-learning 教材としてすぐに実行できる状態にする。また、Java で開発されているため多くの外国語に対応している。

2.2 oq-player

oq-player は oqXML 形式の問題データを e-learning として動作させるためのソフトウェアであり、現時点では Java アプレットの形態をしている。oq-player は問題データに従って、問題文を表示し、解答欄を生成し、解答欄に入力された内容を採点する。oq-player は 3.1 で述べる oqXML

の持つ特徴の全てを活用して動作する。

2.3 oq-printer

oq-printerはoqXML形式の問題データから紙媒体の試験用紙や問題集を作成するためのソフトウェアである。生成される用紙は、問題文のみや解答欄のみなどから構成を選び、採点で用いるための正解や配点を付けるかも選ぶことができ、非常に多様である。

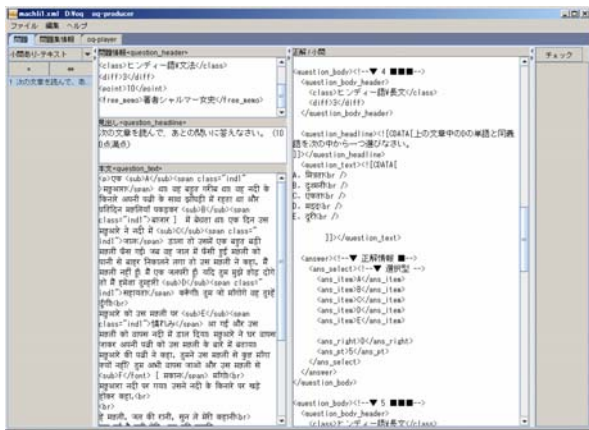


図2 oq-producer

3 oqXMLによる問題記述

3.1 oqXMLの特徴

oqXMLには次のような特徴がある。

- 小問に対応
- 問題文にはほぼ完全なHTMLが使用可能
- 画像、動画、音声に対応
- 日本語や英語以外の多くの外国語に対応
- 選択問題と単純な記述問題に対応
- 極めて複雑な正誤判定や配点に対応

小問は、語学では必須である長文問題で必要である。問題文はHTMLで記述するため、画像や動画、音声で使用可能で非常に表現力が高く、図が必要な問題やリスニング問題など様々な問題を実現可能である。多くの外国語に対してはXMLが通常使う文字コードであるUnicodeによ

り対応可能である。また、oq-playerも多言語に対応しているJavaで開発されているため、多くの外国語に対応可能である。複雑な問題に対しては、次節で述べる問題記述モデルにより、実際の問題程度に複雑な配点や正誤判定には十分に対応できる。oqXMLは通常の問題集や試験問題の中心である選択問題と単純な記述問題、それらを組み合わせた問題に対応している。通常の問題集や試験問題において中心になることは少なく、計算機の処理が苦手としている論述式の問題や絵や図形を描く問題には対応していない。

3.2 問題記述モデル

oqXMLでは一般的によく使われる問題形式に対応するために、正解や配点以外に、解答欄のグループ、完答要求、解答順序の情報を記述する。図3の問題の解答方法と採点用情報は、oqXMLでは図4のような正解ツリーで表現される。

1 次の天体に関する問いに答えよ。

1.1 次の空欄を漢字で埋めよ。
地球が地軸を中心に回転していることを[a]と呼び、太陽の周りを回転していることを[b]と呼ぶ。

a[] b[]

1.2 正しい惑星と衛星の組み合わせを2つ作れ。
A. 地球 B. フォボス C. タイタン D. 月
E. 土星 F. 木星

組み合わせ1[][] 組み合わせ2[][]

図3 問題例

解答欄が複数あるときには、グループが階層構造を持っている場合がある。図3の1.2では4つの解答欄は、組み合わせ1と組み合わせ2の2つのグループに分かれる。また、図3の解答欄の全体では、1.1と1.2に分かれる。解答欄のグループの木構造において、ノードは解答欄のグループを、リーフは解答欄を示す。図4では、解答欄が、1.1と1.2のグループに分かれ、1.2ではさらに組み合わせ1と組み合わせ2のグループに分かれる構造が表現されている。

複数の解答欄があるときには、採点時に特定の

範囲の全ての解答欄が正解であるかどうか、すなわち完答要求があるかどうかの問題になる場合がある。図3の1.1の2つの解答欄は別々に正誤判定ができるが、1.2の各組み合わせが持つ2つの解答欄は両方同時に正解と一致しなければならない。図4では、1.2の組み合わせ1と組み合わせ2のグループで、完答要求が有効になっている。



図4 正解ツリー

複数の解答欄があるときには、解答欄に記述された内容が順不同である場合とそうでない場合がある。図3の1.1では2つの解答欄の内容を入れ替えられないが、1.2では2種類の入れ替わりが可能である。1つ目は、各組み合わせの中での入れ替わりである。1.2の正解はADとCEであるが、各組み合わせの中で、DAとECのように入れ替えても正解である。2つ目は、2つの組み合わせの入れ替わりである。例えば、ADとCE

をCEとADとしても正解である。図4では、1.2の各組み合わせ内で順不同であり、2つの組み合わせの間でも順不同であることが記述されている。1.2の採点用情報の部分をoqXMLで表現すると図5のようになる。answerはノードに、ans_selectは選択型のリーフに対応するが、図4と図5の構造が対応していることがわかる。

```

<answer> ←1.2の採点用情報
  <ans_header> ←複数の解答欄のための情報
    <ans_order>random</ans_order> ←順不同
    <ans_cond>independent</ans_cond> ←完答不要
  </ans_header>

  <answer> ←組み合わせ1
    <ans_header> ←複数の解答欄のための情報
      <ans_order>random</ans_order> ←順不同
      <ans_cond>perfect</ans_cond> ←要完答
    </ans_header>

    <ans_select> ←選択
      <ans_lab>組み合わせ1</ans_lab> ←解答欄ラベル
      <ans_item>A</ans_item> ... <ans_item>F</ans_item>
      <ans_right>A</ans_right> ←正解 ↑選択肢
      <ans_pt>1</ans_pt> ←得点
    </ans_select>

    <ans_select> ←選択 ↓選択肢
      <ans_item>A</ans_item> ... <ans_item>F</ans_item>
      <ans_right>D</ans_right> ←正解
      <ans_pt>1</ans_pt> ←得点
    </ans_select>
  </answer>

  ...組み合わせ2の記述(組み合わせ1と同じ構造)...
</answer>

```

図5 1.2の採点用情報のoqXMLによる記述

4 oqXMLにおける採点処理

oqXMLで記述された問題において、入力に対する採点処理は正解との単純な比較だけでは実現できない。基本的には、完答要求や解答順序の情報に従いながら、正解ツリーと入力ツリーの比較処理を行う必要がある。入力ツリーは図4の正解の部分を入力に置き換え、判定以外の情報をなくしたものである。採点処理では、最初に正解ツリーのルートと入力ツリーのルートが採点関数に入力され、その後は再帰的に処理される。図3の採点では次のようになる。まず、正解ツリー

と入力ツリーのルートを探点関数に与える。その後、正解ツリーと入力ツリーのノードの要素を順に探点関数に与えて、再帰的に処理を行う。探点関数は再帰呼び出しを行うだけでなく、与えられたリーフの入力と正解を比較して正誤判定を行う。図4の1.1の部分では、解答順序は記述通り、完答要求は無効なので、単純に正誤判定をすればよい。

しかし、1.2では、2階層のグループ構造を持つ解答欄、完答要求の存在、順不同の解答順序により採点は複雑である。正解ツリーのあるノードを R とする。 R の完答要求が存在するとき、 R 直下のリーフが全て正解と判定されれば R は正解と判定され、そうでなければ不正解と判定される。場合によっては、 R の下にリーフではなくノードが来る場合もあるが、リーフの場合と違いはなく、直下のノードが全て正解と判定されたとき、 R は正解と判定される。

R の解答順序が順不同となっている場合は、 R 直下の要素 r_i と、同時に採点関数に入力された入力ツリーのノード N の直下の要素 n_j の全ての組み合わせを作って、採点関数を呼び出す。呼び出された採点関数では正誤を判定するが、 r_i と n_j のどちらか一方が既に正解と判定されている場合は判定を行わない。1度使った両ツリーの正解を2度使わないようにするためである。これにより、例えば1.2の2つの組み合わせの両方がADであるとき一方のみを正解と判定することができる。正解ツリーと入力ツリーの両方に判定結果を記録するのはこの処理を実現するためである。

5 大阪外国語大学における事例

大阪外国語大学では、ヒンディ語やスペイン語などにおいて、長文読解、ヒアリングなどのe-learningが実現している[4]。図6はヒンディ語の長文問題がe-learning教材として動作している画面であるが、oqXML, oq-playerの両方が外国

語に対応していることがわかる。

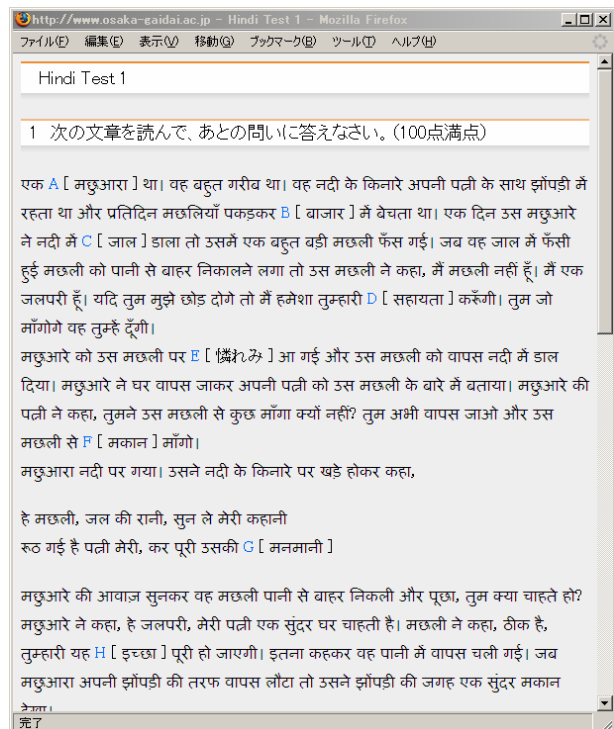


図6 ヒンディ語のe-learning

6 おわりに

本稿では多様な問題文と複雑な採点用情報を記述するためのoqXMLを提案した。また、ソフトウェア群によって紙媒体とe-learningを分けない問題教材の活用環境を実現している。今後の課題としては、解説への対応、SCORM[5]等の標準への対応、学習管理システムと連携などが挙げられる。

本研究は文部科学省の現代的教育ニーズ取組支援プログラム“異文化障壁を乗り越える対話と交渉能力の育成”と、大阪外国語大学の特別研究費II“多言語資源の拡充とe-learningシステムの高度化”の支援を受けて行われた。

参考文献

- [1] <http://www.webclass.jp/>
- [2] <http://hotpot.uvic.ca/>
- [3] <http://www.osaka-gaidai.ac.jp/~sas/Hindi/oqp/>
- [4] <http://www.osaka-gaidai.ac.jp/~sas/Hindi/>
- [5] <http://www.adlnet.org/scorm/index.cfm>