

問題の作成から試験の実施・採点までを支援する簡易 CBT システムの提案

松浦敏雄*1・安留誠吾*2・西田知博*3・宮本友介*4・吉田智子*5・中西通雄*6

*1:大阪市立大学, *2:大阪工業大学, *3:大阪学院大学,
*4:大阪大学, *5:京都ノートルダム女子大学, *6:追手門学院大学

◎Key Words: CBT, 問題自動作成, プログラミング環境, XML, Web

1. はじめに

2025 年度からの大学入試共通テストにおいて「情報」が出題される^[1]。これに合わせて入試を CBT (Computer Based Testing) によって実施することを発表した大学もあり^[2]、今後、CBT の普及が予想されている。入試における CBT は、IT パスポート試験や TOEIC などの資格試験で行われている多肢選択問題のみからなる試験とは異なり、従来の紙ベースの試験と同様の出題形式への対応や、さらには、プログラミングをその場で実行できるような環境を備えたものが望まれている。高等学校の教育現場では、入試対策として CBT への対応が求められるが、このような出題形式に対応し、かつ、手軽に利用できる CBT は見当たらない。

我々は、このような多様な出題形式に対応した CBT システムの実現方法を明らかにし高校現場で簡便に利用可能な CBT システムの実現を目指している。この CBT システムでは、出題者の問題作成を支援し、作成された一つの問題から多数の派生問題を半自動で作成できる^[3]。また、受験者の解答に対して、解が一意に定まるものについては自動採点を行い、そうでない問題については、採点支援のための GUI を用意している。

本稿では、問題の作成方法を示し、作成された問題をどのように記述しているか、また、画面上で問題を表示し試験を実施するために、問題記述をどのように変換しているか、さらには、自動採点、および、採点支援をどのように行なっているかについて明らかにする。

2. CBT システムの概要

本稿で提案する CBT システムでは、問題の作成支援、CBT 実施環境の提供、および、採点まで、CBT の実施を統合的に支援する。

本 CBT で扱う問題は、○×問題、多肢選択問題、穴埋め問題、記述式問題などの紙ベースの試験でよく用いられる基本的な問題(小問と呼ぶ)と、複数の小問から構成される中間、および、少数の中間、もしくは、多数の小問から構成される大問からなる。通常の定期テストや入学試験のような一まとまりのテストは、3 から 5 問程度の複数の大問から、もしくは、多数の小問のみから構成される。

以下、問題作成の手順、試験実施、および、採点の手順について概説する。

(1) 問題の作成

作題者は大問毎に MDL (Mondai Description Language) と呼ぶ形式によって記述する。MDL は、通常の問題文そのものに加えて、以下の付加情報を含む：

- (i) 大問、中間、小問の識別のための記号、および、
- (ii) 各小問の形式、識別番号、正解、惑わしの選択肢、配点などを示す記号

直接テキストで問題を記述しても良いが、問題作成 GUI (図 1) を用いることで簡便に作成できる。ここで作成された問題は、サーバ上の問題プールに蓄積される。

派生問題作成プログラムにより、一つの大問(種問題と呼ぶ)から、類似した多数の派生問題を半自動で生成できる。派生問題の生成方法は文献[15]で述べたのでここでは省略する。MDL の記述方法については 3 章で述べる。

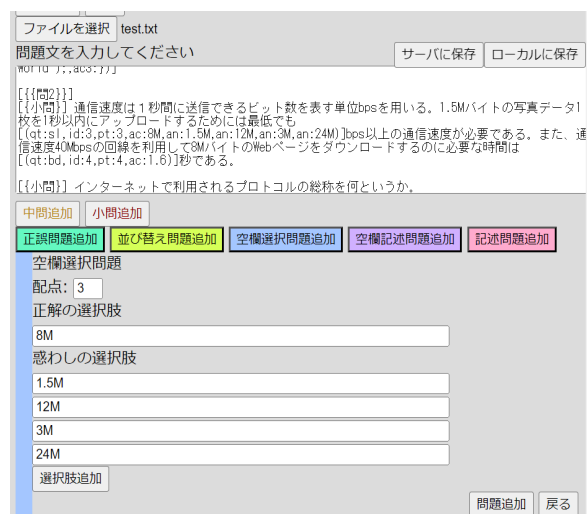


図 1 問題作成 GUI

(2) MDL から XML への変換

MDL は、出題者が少しの記法を学ぶだけで問題の内容を把握できるが、計算機が処理するには必ずしも適切な形式とは言えない。そこで、問題形式変換プログラムを用意し、MDL 形式のテストを入力として、計算機が処理しやすい XML 形式に変換する。XML での記述については、4 章で詳述する。

(3) 正解表の作成

正解表作成プログラムは、MDL による問題記述から問題番号と正解を抽出し、正解表を出力する。このプログラムが MDL を入力としているのは、試験実施の際に Web ブラウザが XML での記述を動的に読み込むの

で、この中に正解を含めることができないためである。

(4) CBT 実施環境

CBT 実施環境は、XML による問題記述を入力し、Web ブラウザ上にテスト問題を提示し試験を実施する。試験問題の画面表示例を図 2 に示す。テストの終了時には、答案を回収しサーバ上に保管する。

第1問
問1 以下の文において、内容が正しければT、誤っていればFをつけなさい。
(1) T F 2023年は令和5年である
(2) Java は JavaScript の略称である T F

第2問
問1 大阪より東にある都市を全て選べ。
 福岡 東京 京都
問2 大阪に最も近い都市を選べ。
 福岡 東京 京都
問3 大阪に最も近い都市は である。

第3問
問1 キーボードから1つの整数を入力し、それを3倍して1を加えた値を表示するプログラムを作成せよ。

編集欄 実行画面を開く 最初に戻す 全削除 選択数 横幅: 縦幅:
変数
 ←
 ← input()
 を表示する
 を改行なしで表示する

図 2 テスト実施時の画面表示例

(5) 半自動採点と採点支援

多肢選択問題や短い語句を問う記述問題などの自動採点可能な問題については、自動採点プログラムが、正解表と各受験者の答案を比較し、採点結果として部分採点済答案を出力する。記述問題など自動採点が困難な問題に対しては、採点支援プログラム(GUI)を利用して、採点者が部分採点済答案を見ながら、Web ブラウザ上で採点する。採点済答案はサーバ上に保存される。

3. MDL による問題の記述

MDL による問題の記述例を以下に示す。

[[[第 2 問]]]

[[問 1]] 大阪より東にある都市を全て選べ。

[[qt: sl, id: 4 pt: 3, an: 福岡, ac: 東京, ac: 京都]]

上記のように、大問、中間、小問のタイトルは、

[[{大問}]], [[{中間}]], [[{小問}]]

のように[[{ ... }]]で囲んで記述する。各小問の問題形式、配点、正解など、画面表示、および、採点に必要な情報を [[(...)]] で囲んで表記している。

“qt”は問題の種類を示すタグであり、その値“sl”は(複数選択可能な)選択問題を示している。“id”は

問題の識別番号、“pt”は問題の配点を表す。“an”は惑わしの選択肢を表し、“ac”は正解を示す。問題の種類の一部を表 1 に示す。

表 1 問題の種類(一部)

記号	問題の説明
sl	select: 選択問題(複数選択可)
rb	radio button: 単一選択問題
tf	true/false: ○×問題
so	sorting: 並べ替え問題
ds	description: 記述問題

4. XML による問題の記述

図2のテスト問題のうち、第1問のみについてのXMLによる問題記述例を次に示す。

```
<?xml version="1.0" encoding="utf-8"?>
<assessment-item>
  <title>期末試験(2024 年度)</title>
  <body>
    <section label="第 1 問">
      <h1>第 1 問</h1>
      <subsection label="問 1">
        <span class="level2">問 1</span>
        以下の文において、内容が正しければ T、
        誤っていれば F をつけなさい。
        <subsubsection label="(1)">
          <question id="1" type="tf">
            <point>10</point>
          </question>
          2023 年は令和 5 年である
        </subsubsection>
        <subsubsection label="(2)">
          Java は JavaScript の略称である
          <question id="2" type="tf">
            <point>10</point>
          </question>
        </subsubsection>
      </subsection>
    </section>
  </body>
</assessment-item>
```

テスト問題は、assessment-item 要素内に記述する。assessment-item 要素の子要素として title 要素と body 要素があり、title 要素はテストのタイトル、body 要素はテスト本文を記述する。テスト本文は、HTML をベースとして記述する。MDL での各小問の記述は、XML では question 要素に変換される。

5 答案と採点の記述形式(JSON)

5.1 答案の記述形式

試験を実施した後の答案は、JSON 形式で集めている。答案の記述例を以下に示す。行番号は説明の都合で付与している。

```

1: {
2:   "examid": "00000012",           - 試験問題 ID
3:   "exameeid": "12345",          - 受験者 ID
4:   "date": {
5:     "start": "20240621T15:10",   - 開始時刻
6:     "submitted": "20240621T16:30" - 提出時刻
7:   },
8:   "answers": {                  - ここから 問題番号: 解答
9:     "1": "3",
10:    "2": "RAID1",
11:    "3": "わかりません",
12:   }
13: }

```

8 行目から 12 行目までの answer 要素で、全ての問題について、問題 id とその解答の組みが記されている。問題 id1 の問題に"3"、id2 に"RAID1"と解答しており、id3 のは記述問題であり、受験者は「わかりません」と解答している。

5.2 部分採点済答案の記述形式

5.1 節の解答の記述例に対する部分採点済答案の記述は、5.1 節の解答例の 12 行目と 13 行目の間に、以下の score 要素が追加されたものとなる。

```

13:   "score": {
14:     "1": { "pt": "2" },
15:     "2": { "pt": "0" }
16:   }
17: }

```

問題 id1 は正解で 2 点が付与され、問題 id2 は不正解で 0 点となっている。ここで、問題 id3 に対する記述がないが、これは未採点であることを示している。

5.3 採点済答案の記述形式

5.2 の部分採点済み答案に対して、採点支援 GUI を用いて、未採点の全ての問題の答案を参照しながら、採点者が採点する。その結果として、部分採点済答案に採点結果を付加した採点済答案が完成する。

付加する部分は、score 要素中の解答が存在していない問題について、問題番号と採点結果を追記する(16 行目)。なお、受験者の解答に対して、採点者がコメント (com 要素)を返すこともできる。

```

13:   "score": {
14:     "1": { "pt": "2" },
15:     "2": { "pt": "0" }
16:     "3": { "pt": "0", "com": "がんばろう" }
17:   }
18: }

```

6 考察

6.1 問題および答案の記述について

問題の記述のために MDL を定義した。作題の際に、できるだけ問題文そのものに集中できるように、問題

文以外でコンピュータ処理に必要な記述を[...]内に限定し、最小限で済ませるようにした。この部分の記述方法は単純ではあるが、必ずしも分かりやすいわけではないので、GUI によって問題作成を支援している。

MDL は問題を画面表示するために必要な情報を含んでいない。そこで MDL による記述を入力し、XML に変換するプログラムを用意した。XML では、Web 画面上に問題を表示して解答を入力するための記述が可能である。

試験問題を XML で記述する標準的な方法が、QTI⁴⁾として提案されている。我々の試験環境は従来の問題集合に加えて、プログラムの実行環境を備えているので、既存の QTI だけでは対応できない。ここでは、QTI の記述方法を離れて、独自の XML を用いているが、将来的に QTI に準拠することを考えている。

受験者の答案は、JSON で記述している。問題と答案の記述を JSON に統一した方がプログラムが簡単になったかもしれないが、QTI の普及を睨んで問題記述は XML ベースを採用した。

6.2 試験実施のために必要な設定

本稿で提案した CBT を動作させるには、Web サーバを用意する必要がある。一部のソフトウェアは Web サーバ上で動作し、また、試験問題、答案、採点済答案などもサーバに置くことになる。

限定された作題者のみに試験問題の登録権限を与えること。試験問題は、試験の実施期間のみ登録された正規の受験者に対して提示し、実施できること。採点結果も受験者本人のみにアクセスを許可するなど、厳密なアクセスコントロールが必要であるが、この部分は未実装である。

試験を実施するためには、サーバやネットワークの適切な設定が必要であり、そのため、本 CBT システムを利用して試験をするためには、これらの知識を有する担当者が必要となる。サーバやネットワークに関する必要な知識をなるべく低減させたかったが、現状では十分とはいえない。

Web サーバとして、Amazon AWS のようなクラウドサービスを利用することもでき、多少の作業量の軽減が図れるかもしれないが、本質的に必要となる知識は変わらないと思われる。

6.3 試験の形式

本稿で提案している CBT は、従来の紙ベースで出題されているような問題を含み、試験中にプログラムを書いて実行できるような CBT ならではの問題をも対象としている。

一方、IT パスポート試験⁷⁾や医学部の臨床実習開始前の「共用試験」⁸⁾など現在広く利用されている多くの CBT は、IRT 方式⁹⁾を採用している。IRT 方式については多くの研究成果が発表されており、その有効性は確認されている。しかし、現状の大学入試に求められている厳密性との親和性が必ずしも良いとはいえないので、世の中の理解が得られるのに多くの時間が必要と思われる。それまでの間は、従来の紙ベースを置き換えられることが CBT に求められると予想している。

7. あとがき

本稿では、Web 上で簡単に利用可能な CBT の一つの実現方法を示した。試験問題を記述するための MDL と、試験実施の際に利用しやすい XML による問題記述方法を提案した。MDL から XML への変換プログラムを用意し、XML をベースに Web 上で試験を実施する仕組みを試作した。試験実施時には、受験者がプログラムを記述し実行する環境も備えている。

派生問題の自動作成プログラム、採点支援 GUI の実装は今後の課題である。また、試験を実施するための Web サーバ上のアクセス制御の仕組みの実現も今後の課題として残されている。

参考文献

- (1) 文部科学省: “「令和7年度大学入学者選抜に係る大学入学共通テスト実施大綱の予告」及び「令和7年度大学入学者選抜実施要項の見直しに係る予告」について”
https://www.mext.go.jp/content/20210729-mxt_daigakuc02-100001207_1.pdf (March 2020).
- (2) 大学入試センター: “CBT での「情報 I」の出題に関する調査研究について(報告)”,
https://www.dnc.ac.jp/research/cbt/cbt_houkoku.html (2022/06).
- (3) 電気通信大学: “2025 (令和7) 年度入学者選抜における CBT システムの活用について (第3報)”, 電気通信大学入試情報,
https://www.uec.ac.jp/news/admission/2024/20240617_6300.html (2024/06).
- (4) 1EdTech Consortium: “Question & Test Interoperability (QTI) 3.0 Overview”, 1EdTech Final Release Spec Version 3.0,
<https://www.imsglobal.org/spec/qti/v3p0/overview> (1 May 2022).
- (5) W3C: “Extensible Markup Language (XML) 1.0 (Fifth Edition)”,
<https://www.w3.org/TR/2008/REC-xml-20081126/> (2024/1/31).
- (6) Ecma Int'l: “ECMA-404”,
<https://ecma-international.org/publications-and-standards/standards/ecma-404/> (2024/1/31).
- (7) 情報処理推進機構: “IT パスポート試験”,
<https://www3.jitec.ipa.go.jp/JitesCbt/index.html> (2024/06/22 確認)
- (8) 医療系大学間共用試験実施評価機構: “臨床実習開始前の「共用試験」”,
<http://www.cato.umin.jp/e-book/17/> (2024/06/22 確認).
- (9) 宇佐美 慧, 荘島 宏二郎 他: “項目反応理論 IRT の考え方と実践 — 測定の質の高いテストや尺度を作成するための技術 —”, Annual Report of Educational Psychology in Japan, vol. 58, pp. 321-329 (2019).
- (10) Nishida, T., Harada, A., Yoshida, T., Nakamura, R., Nakanishi, M., Toyoda, H., Abe, K., Ishibashi, H. & Matsuura, T.: “PEN: A Programming Environment for Novices — Overview and Practical Lessons —”, In J. Luca & E. Weippl (Eds.), *Proceedings of ED-MEDIA, 2008*, pp. 4755- 4760 (Jun 30 2008).
- (11) Michio Nakanishi, Toshio Matsuura, Tomohiro Nishida, Daichi Omiya: “Programming Environment for Novices with Learning and Examination Functions”, EdMedia: World Conference on Educational Media & Technology 2019 (Jun 2019).
- (12) 松浦敏雄, 大宮大地, 安留誠吾, 吉田智子, 西田知博, 中西通雄: “プログラミング学習環境 wPEN を用いた入門用教材”, CIEC 2019 PC Conference (2019/08).
- (13) 松浦敏雄, 清水素彦, 香西省治, 萩原兼一: “教科「情報」における CBT のための問題作成支援機能の実現”, 2017 情報処理学会 関西支部大会, E05 (2017/09)
- (14) Y. Miyamoto, M. Nakanishi, S. Yasutome, T. Nishida, T. Matsuura: “wPEN – Programming Environment for Novices: its Extensions and Classroom Practice”, World Conference on Educational Media & Technology 2023, Viena (2023/07).
- (15) 松浦敏雄, 安留誠吾, 西田知博, 宮本友介, 吉田智子, 中西通雄: “CBT におけるプログラミング問題の派生問題の半自動作成方法の提案”, CIEC 2023 PC Conference (2023/08).