

情報分野におけるコンピュータ利用特性を活かした ブラウザ型 CBT システムの開発

香西省治¹ 角谷良彦^{2,6} 西田知博³ 植原啓介⁴ 萩谷昌己² 萩原兼一⁵

概要：文部科学省の大学入学者選抜改革推進委託事業（情報分野）で研究開発した CBT (Computer Based Testing) システムを実装面から概説する。本事業の目的は、思考力・判断力・表現力を評価する「情報科」の試験問題を研究することである。この評価のためには、紙による出題よりプログラミング能力などを効果的に評価しやすい CBT による出題の方が試験問題の幅を広げることができると考えた。この CBT システムは、「CBT ならでは」の試験問題を受験者に出題することを重視して、実際に作成された「CBT ならでは」の模擬試験問題を一般化・仕様化し、その後発生する仕様変更にも柔軟に対応できるブラウザ型の専用システムとして開発した。この CBT システムは、出題する試験問題を確認する作問機能部、受験者が受験するための機能を持つ試験機能部、試験結果を採点・集計する採点・集計機能部の 3 論理サブシステムから構成される。CBT システムによる試験実施運用シーケンスに基づく各論理サブシステムの動作と、プログラミング問題、ゲームブック型問題等の特徴的な 5 形態の試験問題の処理を示すことを通して、開発した CBT システムがこの事業で研究した「CBT ならでは」の試験問題を出題できることを示す。尚、2017 年と 2018 年に実施した実証実験では、主にサーバ負荷の面で問題は発生していない。

キーワード：試験監督機能、試験問題定義書、思考力評価、プログラム実行機能、ゲームブック型、短冊型

Development of Web-Based CBT System

Utilizing Computer to Solve Examination Questions in “Information”

Shoji Kosai¹ Yoshihiko Kakutani^{2,6} Tomohiro Nishida³
Keisuke Uehara⁴ Masami Hagiya² Kenichi Hagiwara⁵

1. はじめに

試験実施の過程全てに関してコンピュータを用いて行なう CBT (Computer Based Testing) は、従来のマークシート選択式や筆記試験と比べて、さまざまな機能性を有している [1, 2, 3, 4]。

大阪大学は、文部科学省から大学入学者選抜改革推進委託事業（情報分野）（以下、本事業）を受託し、平成 28 年度から平成 30 年度まで情報処理学会および東京大学と連携して本事業を推進した [5, 6, 7, 8]。本事業の目的は、思考力・判断力・表現力を評価する「情報科」の試験問題を研究することである。この評価のためには、紙による出題 (Paper Based Testing, PBT) よりプログラミング能力などを効果的に評価しやすい CBT による出題の方が試験問題の幅を広げることができると考え、CBT システム（以下、本システム）を開発した [9, 10]。具体的には、平成 28 年度は従来 PBT の形態で実施してきた試験 [11, 12] を CBT の形態で実施することを目的に開発し、平成 29 年度はさらにこの CBT システムに CBT の特長を生かす形態の試験を実施するための機能拡充を行った。

本稿では、本システムについて、その開発内容を中心に、以下を報告する。

(1) 「CBT ならでは」の試験問題を受験者に出題することを重視して、実際に作成された「CBT ならでは」の模

擬試験問題をできるだけ試験問題形式として一般化・仕様化する一方、その後発生する仕様変更にも柔軟に対応するためにブラウザ型専用 CBT システムの形態で本システムを開発した。

- (2) 本システムは、出題する試験問題を確認する作問機能部、受験者が受験するための機能を持つ試験機能部、試験結果を採点・集計する採点・集計機能部の 3 論理サブシステムから構成される。本システムによる試験実施運用シーケンスに基づく各論理サブシステムの動作と、プログラミング問題、ゲームブック型問題等の特徴的な 5 形態の試験問題の処理を示すことを通して、開発した CBT システムはこの事業で研究した「CBT ならでは」の試験問題の出題を実現した。
- (3) 2017 年度と 2018 年度に実施した本 CBT システムを使用した実証実験では、サーバ負荷の面で本システム運用上の問題は発生していない。

2. CBT システムモデル

2.1 取り扱う試験問題の構成とその記述

試験問題は、一つ以上の大問 (図 2-1 に「第 1 問」「第 2 問」と表示) から構成される。大問は他の大問と設問の内容が独立で、関連しない。大問は、図 2-1 において「問 1」「問 2」と表示される一つ以上の中間で構成され、さらに中間は具体的な設問である一つ以上の小問で構成される。

1 データアクセス株式会社/DataAccess Inc.

2 東京大学/The University of Tokyo

3 大阪学院大学/Osaka Gakuin University

4 慶応義塾大学/Keio University

5 大阪大学/Osaka University

6 株式会社マインド・アーキテクト/Mind Architect Inc.

本システムでは、大問の単位でその内容を XML 形式のテキストファイルである大問定義書に記述する。

本システムでは、実施する試験問題は、出題する大問の大問定義書を内部に含んだ XML 形式のテキストファイルである試験問題定義書に記述する。図 2-2 に試験問題定義書の構成と、表 2-1 に試験問題定義書ヘッダ部の構成を示す。

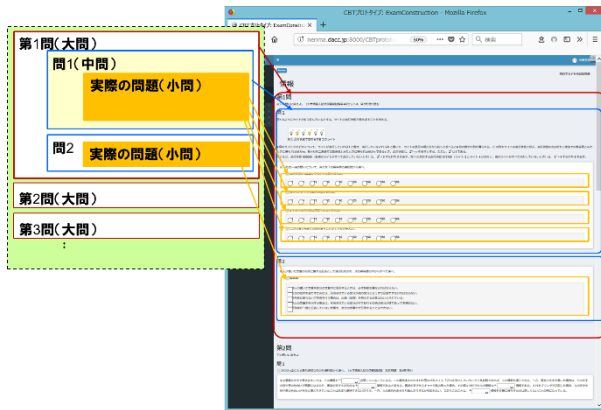


図 2-1 試験問題の構成

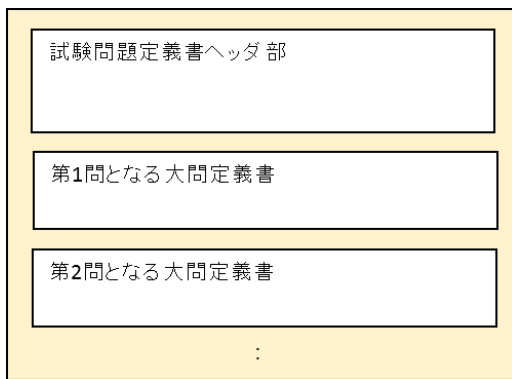


図 2-2 試験問題定義書の構成

表 2-1 試験問題定義書ヘッダ部の構成

記述項目	説明	備考
試験名	実施する試験の名称.	受験者 登録と 関連する.
科目名	実施する試験の科目名.	
補足情報 1	実施する試験の補足情報 1 (省略可).	関連する.
補足情報 2	実施する試験の補足情報 2 (省略可).	
試験タイプ	受験者がログイン後、試験中のみ閲覧・解答できる試験か、常に閲覧、解答可能なアンケートかを指定する.	

2.2 CBT システムのユースケース図、運用のモデル

本システムのユースケース図を図 2-3～図 2-5 に示す。ユースケース図に基づき、本システムにより CBT での試験を実施する際の全体の作業の流れを以下の(1)～(7)に示す。

(1) 大問の作成

作問者は、大問定義書を作成することにより大問を作成する。作問者は、作成した大問定義書を作問機能部にアップロードしてその内容を確認する。

(2) 試験問題の作成

試験問題コーディネータは、作問者が作成済みの大問から出題する大問を選択して試験問題を構成する。これは、出題対象の全ての大問定義書を編集して試験問題定義書を作成することにより行う。査読者は、試験問題コーディネータが作成した試験問題の内容にコメントする。試験問題コーディネータは適切にそのコメントに対応して、試験問題定義書を完成させる。

(3) 試験スケジュールの作成・周知

会場管理者は、実施する試験の試験問題定義書を決定し、試験名と試験科目名、試験会場、試験日時、受験者数等の試験スケジュールを作成し、関係者に周知する。

(4) 試験問題の登録

試験問題登録者は、試験スケジュールに従って試験問題定義書のヘッダ部を編集し、試験機能部に登録する。

(5) 受験者の登録

受験者登録者は、試験スケジュールに従って受験者登録ファイルを作成し、試験機能部に登録する。作成・登録した受験者の ID/パスワードを関係者に通知する。

(6) 試験当日の試験監督

試験当日、試験監督者は、試験機能部により受験者に試験の開始と終了を指示し、受験させる。

(7) 採点・集計

試験終了後、採点者は、採点・集計機能部により受験者解答をダウンロードし、採点して、必要な集計を行う。

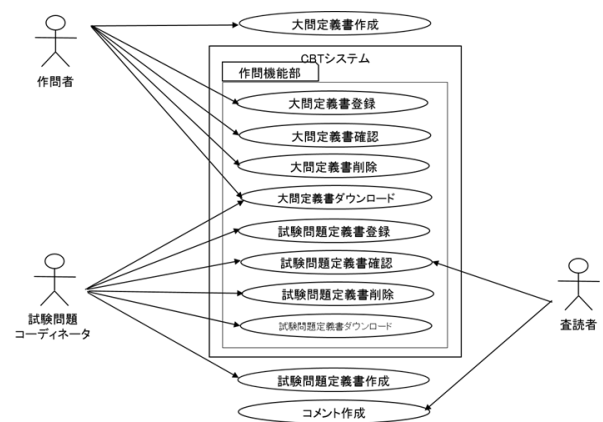


図 2-3 CBT システムのユースケース図 (1)

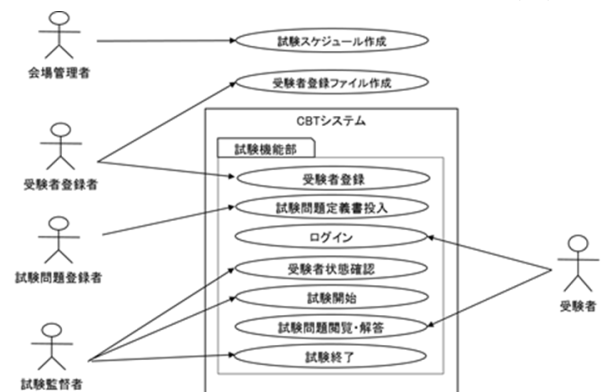


図 2-4 CBT システムのユースケース図 (2)

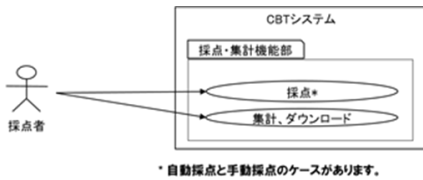


図 2-5 CBT システムのユースケース図 (3)

2.3 その他

受験者が使用する端末は PC で、キーボードとマウスを操作デバイスとする。

3. CBT システムの開発方針

本システムの開発に当たっては、これまでの CBT システムに囚われない発想に基づく「CBT ならではの」様々な形態の試験問題を受験者に出题することを重視し、開発期間を含む限られた開発リソースの下でこれらを有効利用して開発することとした。

具体的には、以下の方針で本システムを開発した。

(1) 大問定義書と試験問題定義書の各仕様化

実際に作成された「CBT ならではの」模擬試験問題をベースに、その形式をできるだけ一般化して大問定義書と試験問題定義書の仕様を策定する。その仕様に基づいて開発することにより、本システムで扱う「CBT ならではの」試験問題の汎用性を確保した。

(2) 既存システムベース開発か、本事業専用開発か

既存システムをベースにした開発は、そのシステムが具備する機能を利用する範囲では開発期間と開発稼働量の両方の削減効果が期待できるが、その一方、新しい発想に基づく「CBT ならではの」試験問題出題の実現に当たって、その既存システムが持つ様々な制約等の影響を受ける虞がある。そのため本システムは、本事業専用の CBT システムとして開発して「CBT ならではの」試験問題出題実現の自由度を確保するとともに、公開されている OSS ライブラリ等を活用して、開発期間と開発稼働量の削減に努めることとした。

(3) 受験者の CBT での受験形態

受験者が CBT で受験する形態として、一般に以下の 2 形態が考えられる。

(形態 1) 専用アプリの形態とし、受験者の PC にダウンロードした専用アプリを動作させ受験する。

(形態 2) ブラウザ型の形態とし、受験者の PC ブラウザ上で動作させ受験する。

「CBT ならではの」試験問題の仕様変更を CBT システムに反映するには、CBT システムのプログラム変更が必要となる。プログラム変更の都度、形態 1 では PC 上のアプリ変更が必要となる一方、形態 2 では CBT システムのサーバプログラム修正のみで良い。本事業では CBT システムのプログラム変更は頻繁に発生することが予想さ

れたため、形態 2 で本システムを開発し、「CBT ならではの」試験問題の研究に伴う仕様変更柔軟に対応可能とした。

(4) 特に重視しない事項とその対応・対策

一般に、システム開発で重要となる表 3-1 に示す項目について、本システムでは特にプログラムでの対応を行わず、原則として運用で対応することとした。

表 3-1 特に重視しない事項とその対応・対策

項目	対応・対策
受験時の CBT システムの堅牢性/耐故障性	受験者 PC 故障時は PC を交換して受験を継続するか、その受験者の受験を中止する。サーバまたはネットワークの故障発生時は本システムを用いた受験全体を中止する。これらの判断は試験会場監督者等の試験実施運用者の判断に委ねる。
CBT システム全般のセキュリティ	通信路及び本システムに保持するデータの暗号化を行わないため、受験者の個人情報に関わる情報を CBT システムに保持しないことを前提に本システムを運用する。
受験者のカンニング等の不正行為	受験者の不正行為の検出等については、試験監督者等の管理・監督に委ねる。
システムの拡張性	サーバ過負荷が発生して本システムの運用に支障が出るのが予想される場合、スケールアップにより解決を図る。

4. CBT システムの具体的な開発

4.1 CBT システム動作環境・条件

本システムの構成を図 4-1 に、実際に使用した各サーバのハードウェア条件を表 4-1 にそれぞれ示す。また、本システムのソフトウェア条件を表 4-2 に示す。

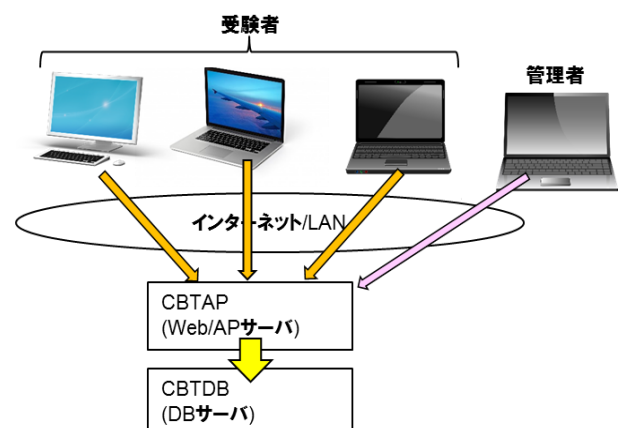


図 4-1 システム構成

表 4-1 ハードウェア条件

サーバ	機能分類	諸元
CBTAP	Web/AP サーバ	インテル XeonE5-2680v4 2.4GHz, 35M
CBTDB	DB サーバ	キャッシュ, 9.60GT/s QPI, ターボ, HT, 14C/28×2, MEM64GB

表 4-2 ソフトウェア条件 (サーバ, PC)

項目	内容	備考	
サーバ	OS	CentOS7.4	
	プログラミング言語	PHP 5.4.16, CakePHP2.9.4	CBTAP サーバ
	HTTP サーバ	Apache/2.4.6	CBTAP サーバ
	RDBMS	5.5.56-MariaDB	CBTDB サーバ
PC	ブラウザ	Firefox (最新版)	標準として規定

尚, 図 4-1 のシステム上で, 図 4-2 と表 4-3 に示す本システムのソフトウェアの機能部とサブ機能部が動作する。

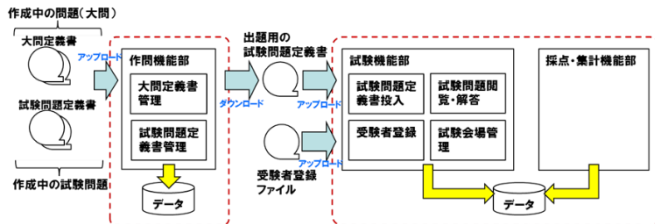


図 4-2 ソフトウェア構成

表 4-3 CBT システムを構成する機能部とサブ機能部

機能部	サブ機能部	説明・提供機能
作問機能部	大問定義書管理	作成中の大問定義書のアップロード, 動作確認, ダウンロードの各機能を提供する。
	試験問題定義書管理	作成中の試験問題定義書のアップロード, 動作確認, ダウンロードの各機能を提供する。
試験機能部	試験問題定義書投入	実施試験の試験問題定義書のアップロード機能を提供する。
	受験者登録	実施試験の受験者の登録機能を提供する。
	試験会場管理	実施試験の試験会場毎に試験開始・終了指示の機能を提供する。
	試験問題閲覧・解答	受験者ログイン後の試験問題閲覧, 解答の各機能を提供する。
採点・集計機能部		受験者解答ファイルのダウンロード, 採点結果ファイルのアップロード, 集計の各機能を提供する。

4.2 CBT システム実行権限

本システムでは, 表 4-4 に示す実行権限を定義し, アカウント/ID に付与する。

4.3 試験問題の作成, 登録

本システムで出題できる問題の形態を表 4-5 に示す。

図 4-2 に示す通り, 大問定義書を作問機能部の大問定義書管理機能により, 試験問題定義書を作問機能部の試験問題定義書管理機能により, それぞれアップロードして本システムに蓄積し, 動作確認とともに, その大問定義書と試験問題定義書をダウンロードできる。この操作は作問者/試験問題コーディネータ権限を持ったアカウントで実行可能である。

さらに, 図 4-2 に示す通り, 作問機能部で動作確認した

試験問題定義書を編集し, 実施する試験問題として, 試験機能部の試験問題定義書投入機能によりアップロードする。この操作は試験問題登録者権限を持ったアカウントで実行可能である。

作問機能部の大問定義書管理と試験問題定義書管理の機能, 試験機能部の試験問題定義書投入機能では, アップロードされた大問定義書/試験問題定義書を処理し, HTML ファイルの基ファイル (テンプレートファイル) に変換して, 本システムに蓄積する。アップロードした大問定義書/試験問題定義書の動作確認 (プレビュー) は, 本システムがテンプレートファイルから HTML ファイルを生成してプレビュー要求した PC に転送し, PC ブラウザ上で実行することにより行う。

表 4-4 CBT システムでの実行権限

実行権限	実行権限の内容	備考
作問者/試験問題コーディネータ	作成した大問定義書と試験問題定義書を作問機能部にアップロード/ダウンロードし, 内容を確認する。	図 4-1 における管理者の位置付け
試験問題登録者	本システムで実施する試験問題定義書を試験機能部に登録する。	
受験者登録者	本システムで実施する試験の受験者を試験機能部に登録する。	
試験監督者	本システムで実施する試験を監督する。試験機能部で試験実施・終了指示, 試験中の受験者状態監視を行う。	
採点者	本システムで実施した試験結果を採点・集計機能部でダウンロード/アップロードし, 集計結果を閲覧する。	
受験者	本システムで CBT 試験を受験する。	

表 4-5 出題できる問題の形態一覧

出題単位	問題形態	説明
小問	記述型	解答欄に単語, 式, 数値, 文を記述して解答する問題。
	ラジオボタン選択肢型	縦並び/横並びの選択肢から一つ選択して解答する問題。
	チェックボックス選択肢型	縦並び/横並びの選択肢から複数選択して解答する問題。
	プルダウンメニュー選択肢型	プルダウンメニュー形式の選択肢から一つ選択して解答する問題。
	短冊型	選択肢欄から選択肢を解答欄にドラッグ・アンド・ドロップ (DnD) し, 解答欄の選択肢を並べ替え, 削除する問題。
中間	プログラム実行型	ブロック形式プログラミングを行い, 実行させることを可能とする問題。
	穴埋め型	中間の問題文中の解答欄に解答する問題。
大問	ゲームブック型	中間を構成する小問の解答結果に依存して次に出題する小問の表示を変更できる問題。
	ランダム出題型	受験者毎に大問を構成する中間の出題内容を可変にする問題。

4.4 受験者登録

図 4-2 に示す通り、受験者登録ファイルを作成して試験機能部の受験者登録機能でアップロードすることにより、受験者を本システムに登録する。この操作は受験者登録者権限を持ったアカウントで実行可能である。

受験者登録ファイルには、表 4-6 に示す情報を CSV 形式のテキストファイルに記述する。

表 4-6 受験者登録ファイルの記述情報

記述項目	説明	備考
試験名	受験する試験名。	試験名は試験問題定義書ヘッダ情報の試験名と関連する。試験会場管理機能で使用する。
試験日、試験開始・終了日時	受験する日付と開始、終了時刻。	
受験者毎の情報	受験者毎の以下の情報 ・受験者 ID, パスワード ・科目名 ・補足情報 1 (省略可) ・補足情報 2 (省略可) ・試験会場名	科目名、補足情報 1, 2 は試験問題定義書ヘッダ情報の科目名、補足情報 1, 2 と関連する。試験会場名は試験会場管理機能で使用する。

4.5 試験会場管理

受験者登録ファイルに記述された以下の内容で試験会場を一意に特定し、その会場の受験者の試験開始、終了を試験機能部の試験会場管理機能により指示する。この操作は試験監督者権限を持ったアカウントで実行可能である。

- 試験名、試験日、試験開始・終了日時
- 試験会場名

4.6 受験（試験問題閲覧・解答）

4.6.1 受験できる試験問題

本システムは、受験者権限を持ったアカウント（受験者 ID）で試験機能部にログインした受験者毎に、表 4-7 に示す条件でその受験者が受験できる試験問題を決定し、その一覧を表示する。

試験監督者が試験開始操作を実施後、その試験問題を開覧し、解答できる。

表 4-7 受験者が受験できる試験問題定義書の条件

受験者登録ファイルの記述項目		試験問題定義書ヘッダ部の記述項目との関係
試験名		試験名と一致すること。
科目名		科目名と一致すること。
補足情報 1	設定時	補足情報 1 と一致すること。
	省略時	補足情報 1 によるチェックを行わない。
補足情報 2	設定時	補足情報 2 と一致すること。
	省略時	補足情報 2 によるチェックを行わない。

4.6.2 受験者 PC ブラウザと CBT システムサーバ間の通信

受験者の PC ブラウザと本システムのサーバ間の通信プロトコルは全て HTTP である。PC ブラウザの以下のタイミングで通信が発生する（図 4-3）。それぞれのタイミングでの PC ブラウザとサーバでの処理を以下に示す。

- ログイン/ログアウト認証 …受験者の ID とパスワードによる PC ブラウザとサーバでの認証処理。
- 試験問題閲覧開始（試験開始、試験問題 URL アクセス） …サーバが当該試験問題の HTML コンテンツを生成して PC ブラウザへ送信する。PC ブラウザは、受け取った HTML コンテンツを画面に表示する。
- 試験問題の解答箇所解答 …JavaScript ライブラリ機能により、フォーカスしている（マウスを移動して解答可能になっている）解答欄のそのフォーカスが外れた（マウスをその解答欄から移動した）タイミングで、PC ブラウザからサーバに解答欄入力内容を送信する。サーバは PC ブラウザに解答を受け取った時刻を返却する。この時刻は PC ブラウザ上にその受験者の最新解答時刻として表示する。

尚、本システムでは、当該 PC ブラウザで受験した受験者 ID と試験問題の識別情報、解答内容をその PC ブラウザのローカルストレージに保存する。PC ブラウザは、試験問題閲覧開始時にサーバから受け取った HTML コンテンツを画面に表示する時、ローカルストレージに保存した受験者 ID と試験問題識別情報が一致する場合は、ローカルストレージに保存した解答内容を各解答箇所に表示する。これにより、同一受験者が同一の PC でログアウト後に再ログインして同一の試験問題を再開する操作がスムーズに行える。ログインした受験者 ID がローカルストレージに保存した受験者 ID と異なる場合、PC ブラウザは、ローカルストレージに保存したその試験問題の解答内容をクリアして、未解答の状態で大解答箇所を表示する。

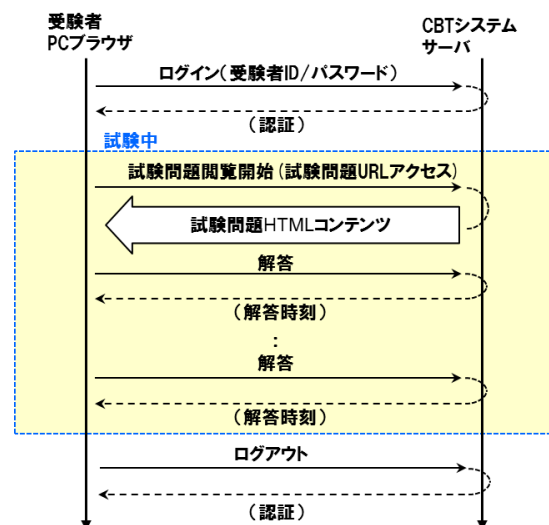


図 4-3 受験者 PC ブラウザと CBT システムサーバ間の通信

4.6.3 受験者操作分析用データ収集

本システムは、受験者毎の解答操作の分析を目的に、試験実施中に表 4-8 に示す情報を RDBMS の具備するトリガーの機能により本システムのサーバに収集し、蓄積する。図 4-4 に本データの CSV 形式のテキストファイルへの出力例を示す。

表 4-8 受験者操作分析用収集データ

項目	説明	備考
試験名	試験問題定義書に記述した試験名。	
科目名	試験問題定義書に記述した科目名。	
受験者 ID	受験者 ID。	
操作識別情報	ログイン/ログアウト、初回解答、解答修正の各操作。	試験開始、終了の操作は未収集。
問題識別情報	解答した試験問題箇所を一意に特定する情報。	
解答内容	本システムのサーバが受け取った解答内容。	
操作時刻	本システムのサーバが操作内容を受け取った時刻。	

試験名	科目名	操作識別情報	解答内容	
2018年度模試	情報12a	XX1800	INSERT,"1_63_1",["オ"]	2018-08-01 10:16:00
2018年度模試	情報12a	XX1800	UPDATE,"1_63_1",["オ"]	2018-08-01 10:16:02
2018年度模試	情報12a	XX1800	UPDATE,"1_63_1",["オ"]	2018-08-01 10:16:07
2018年度模試	情報12a	XX1800	INSERT,"1_48_1",["ア"]	2018-08-01 10:19:22
2018年度模試	情報12a	XX1800	INSERT,"1_65_1",["ウ"]	2018-08-01 10:19:34
2018年度模試	情報12a	XX1800	UPDATE,"1_65_1",["ウ"]	2018-08-01 10:19:41
2018年度模試	情報12a	XX1800	UPDATE,"1_65_1",["ウ"]	2018-08-01 10:19:53
2018年度模試	情報12a	XX1800	UPDATE,"1_65_2",["ウ"]	2018-08-01 10:20:00
2018年度模試	情報12a	XX1800	INSERT,"1_65_3",["ウ"]	2018-08-01 10:20:40
2018年度模試	情報12a	XX1800	INSERT,"1_60_1",["ABAACA"]	2018-08-01 10:21:26
2018年度模試	情報12a	XX1800	INSERT,"1_68_2",["ウ"]	2018-08-01 10:21:52
2018年度模試	情報12a	XX1800	INSERT,"1_68_1",["ウ"]	2018-08-01 10:22:02
2018年度模試	情報12a	XX1800	UPDATE,"1_68_2",["ア"]	2018-08-01 10:22:17
2018年度模試	情報12a	XX1800	UPDATE,"1_68_1",["ウ"]	2018-08-01 10:22:43

図 4-4 受験者操作分析用収集データ CSV ファイル出力例

4.6.4 特徴的な問題形態の開発・実装

本システムで扱う特徴的な問題形態として、短冊型、プログラム実行型、穴埋め型、ゲームブック型、ランダム出題型の各問題形態の開発・実装内容を順に示す。

(1) 短冊型問題

短冊型問題とは、図 4-5 に例示するように、選択肢欄の選択肢を小問の解答欄にドラッグ・アンド・ドロップ(DnD)し、解答欄のその選択肢を並べ替え、削除することを基本操作とする出題形式の問題である。選択肢欄と解答欄に、選択肢を横並びと縦並びのいずれかで表示することができる(図 4-5 は縦並び)。図 4-5 に対応した大問定義書の記述(一部抜粋)を図 4-6 に示す。

本システムでは、解答欄に並べた選択肢に設定された値(value タグ設定値)を PC ブラウザからサーバに JSON 配列の形式で送信する(図 4-5 で解答欄のそれぞれの選択肢にア, ウ, エ, オの値が設定されている場合, ["ア", "ウ", "エ", "オ"]を送信する)。



図 4-5 短冊型問題例

```

<s_questions>
<text>
「正の整数 n を入力したとき、n を超えない奇数の合計を出力する。」プログラム。 &lt;br&gt;
(たとえば 7 を入力したら、1 + 3 + 5 + 7 = 16 なので、16 と出力する。)
</text>
<type>sort_vertical</type>
<choices><value>ア</value><caption>ア.n に整数値を入力</caption></choices>
<choices><value>ウ</value><caption>ウ.i ← 0</caption></choices>
<choices><value>エ</value><caption>エ.j ← 1</caption></choices>
<choices><value>オ</value><caption>オ.k ← 1</caption></choices>
</s_questions>

```

図 4-6 短冊型問題の大問定義書記述例

この形態の問題の発展形として、以下を大問定義書に記述することができる。

- ▶ 選択肢欄の選択肢を複数の解答欄(小問)で共通化
 - ▶ 選択肢を複数のグループの選択肢欄に分けて表示
 - ▶ 解答欄を動的に複数行に増減可能にして表示し、それぞれの行に選択肢を DnD して横並びで表示し、並べ替え等の操作を行う。これは、解答欄の 1 行を SQL 文と見立て、SQL 文要素を選択肢として各行に配置して、SQL 文列の作成を出題する利用ケースを想定している。
- また、各選択肢をプログラム文と見なし、解答欄の選択肢の並びをプログラム構造に基づき自動インデントして表示すること、さらに各選択肢に JavaScript プログラム文を当てはめてその解答欄の選択肢の並びのプログラムを実行させること、等の拡張が可能である。

(2) プログラム実行型問題

プログラム実行型問題は、受験者がプログラムを書く形態の問題であり、以下のような特徴を備えている。

- ▶ 与えられた制限の範囲内で受験者は自由にプログラムを書くことができる。
- ▶ プログラミング言語はブロック形式であり、利用可能なブロックは出題者が設定する。
- ▶ 試験中、受験者が自身の書いたプログラムを PC ブラウザ上で実行し、その結果を確認することができる。ブロック形式のプログラミング言語において、プログラムは連結された一連のブロックとして表現される。本シ

テムでは、マウスを使ってプログラミング領域内でブロックをDnDすることで、プログラミングを行うことができる。新しくブロックを追加するには、ブロックペインからブロックをDnDする。図 4-7 では、左側がブロックペイン、右側がプログラミング領域となっている。試験開始時にどのブロックをどこに配置するか、ブロックペインにどのブロックを準備するかは問題ごとにカスタマイズが可能な仕様となっている。

短冊型問題はブロックを一列に並べる形式の問題と捉えることもできるが、本プログラム実行型問題はより複雑な入れ子構造にも対応しており、短冊型問題の拡張としてプログラミング以外に利用することも不可能ではない。

プログラム実行型問題の際立った特徴は、受験者が試験中にプログラムを実行できるという点にある。自分が書いたプログラムが正しいかどうかを確認しながらプログラムを書くことができるため、受験者はデバッグ作業を行うことが可能となる。多くの場合、デバッグ作業はプログラミングに不可欠であり、より実践的なプログラミング能力を問うのに適していると言える。

本システムでは、図 4-7 にあるように、プログラムへの入力としてテーブル状のデータを想定しており、プログラムの実行結果もテーブルに表示される。現在の実装は二次元データのみを対象としているが、どのように表示するかという点を除けば、多次元への拡張は容易である。

今回のシステムでは、プログラム実行型問題は小問として扱われる。プログラム実行型問題を問題類型として、単独のプログラミング問題を作問することは当然想定されているが、加えて、問題の一部に図として組み込むことも可能となっている。

実装には、Google が開発している Blockly[13]を使用している。Blockly はブロック形式のプログラミング言語であり、PC ブラウザ上で利用するためのライブラリがフリーソフトウェアとして公開されている。本システムは Blockly のカスタマイズ機能を利用しており、問題に合わせてブロックを定義することができる。新しく定義されたブロックをシステムで有効にするには、システム管理者の操作が必要である。その他の設定は大問定義書で行うが、作問者の便宜に配慮してブロックの配置は別に XML ファイルとして準備することになっている。図 4-7 に対応した大問定義書の記述（一部抜粋）を図 4-8 に示す。

プログラム実行時、ブロックとして書かれたプログラムは一旦 JavaScript コードにコンパイルされる。プログラムのコンパイル及び実行は PC ブラウザがクライアントサイドで行うので、実行自体はオフラインでも可能である。他の型の問題と同じくプログラミング領域の情報は常に PC ブラウザにキャッシュされているので、試験を中断したとしても、途中の状態から再開することが可能である。

本システムにおいて、サーバに送られる答えはコンパイ

ル後の JavaScript コードである。実装としては、ブロックの配置情報を XML などの形式で送るという選択肢もあるが、プログラミングの採点を自動で行う場合には、実際に実行可能なコードの方が便利なのが多いという設計思想に基づいて、現在の仕様となっている。

受験者が書いたプログラムを PC ブラウザ上で実行する機能により、プログラム実行型問題には、信頼できない不適切なコードを実行してしまうというセキュリティ上の懸念がある。しかしながら、利用可能なブロックを作問者が指定できるため、設問を適切に行うことでセキュリティ上問題のあるコードの実行を抑制することが可能である。また、無限ループや異常な回数のループを用いる高負荷なプログラムの実行に対しては、コンパイル時にループ回数の上限を設定することで対処が可能であり、実際に本システムではそのような対処を行っている。



図 4-7 プログラム実行型問題例

```
<m_questions>
<block>
<text>
以下のとおりに、0から数を最後まで順に進むプログラムを作れ。
なお、無駄に多くのブロックを使っている場合は、減点、あるいは0点となる。
</text>
</block>
<block_panel>
<s_questions>
<type>tableworld</type> ←ブロック形式のプログラム実行型問題とする記述
<text>(1)</text>
<tw_table id="tw_table_2_1" left="目標" right="実行結果" show_index="false">
<tw_sample row="6" col="6">


|   |   |   |   |   |    |
|---|---|---|---|---|----|
|   |   |   |   | 9 | 10 |
|   |   |   | 7 | 8 |    |
|   |   | 5 | 6 |   |    |
|   | 3 | 4 |   |   |    |
| 1 | 2 |   |   |   |    |
| 0 |   |   |   |   |    |


</tw_sample>
</tw_table>
<tw_exec code="tw_editor_2_1" table="tw_table_2_1">プログラムを実行</tw_exec>
<tw_editor id="tw_editor_2_1" width="600" height="400">
<type>algorlogic</type>
</tw_editor>
</s_questions>
</block_panel>
</m_questions>
```

【ブロック形式プログラミング実行型問題記述部】

図 4-8 プログラム実行型問題の大問定義書記述例

(3) 穴埋め型問題

穴埋め型問題とは、図 4-9 に例示するように、以下の特徴を持つ中間である。

- ▶ 中間の単位で穴埋め型問題を構成し、その問題文中にラベル付けされた解答欄を表示する。
- ▶ ラベルが異なる解答欄毎に一つの小問が対応する。
- ▶ 解答欄に該当する小問には、記述型かプルダウンメニュー選択肢型の何れかの形態を設定可能（図 4-9 はプルダウンメニュー選択肢型の例）である。
- ▶ 問題文中の同一ラベル名の解答欄は、一方に設定した解答内容を他方の解答欄に自動的に反映して表示する。

図 4-9 に対応した大問定義書の記述（一部抜粋）を図 4-10 に示す。

本システムでは、記述型の解答欄はその入力した値を、プルダウンメニュー選択肢型の解答欄はその選択肢に設定された値（value タグ設定値）を、それぞれ PC ブラウザからサーバに送信する。

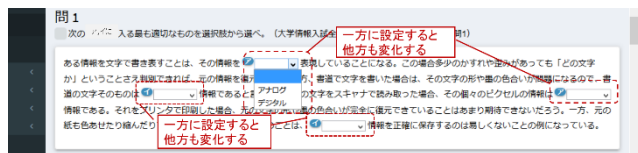


図 4-9 穴埋め型問題例

```
<m_questions>
<block>
<text>
次のア、イに入る最も適切なものを選択肢から選べ。
(大学情報入試全国模擬試験 試作問題 第1問問1)
</text>
</block>
<block_panel>
<text>
ある情報を文字で書き表すことは、その情報を%===[1]===%表現していることになる。この場合多少のかすれや歪みがあっても「どの文字か」ということさえ判別できれば、元の情報を復元できる。一方、書道で文字を書いた場合は、その文字の形や墨の色合いが問題になるので、書道の文字そのものは%===[2]===%情報であると言える。書道の文字をスキャナで読み取った場合、その個々のピクセルの情報は%===[1]===%情報である。それをプリンタで印刷した場合、元の文字の形や墨の色合いが完全に復元できていることはあまり期待できないだろう。一方、元の紙も色あせたり縮んだりするかも知れない。つまりこのことは、%===[2]===%情報を正確に保存するのは易しくないことの例になっている。
</text>
</block_panel>
【問1の穴埋め問題文の「ア」の解答箇所の解答形式の記述部】
<s_questions>
<text></text>
<type>selectbox</type> <プルダウンメニューによる解答をとる記述
<choices><value>ア</value><caption>アナログ</caption></choices>
<choices><value>イ</value><caption>デジタル</caption></choices>
</s_questions>
【問1の問題文中の「イ」の解答箇所の解答形式の記述部】
<s_questions>
<text></text>
<type>selectbox</type> <プルダウンメニューによる解答をとる記述
<choices><value>ア</value><caption>アナログ</caption></choices>
<choices><value>イ</value><caption>デジタル</caption></choices>
</s_questions>
</m_questions>
```

図 4-10 穴埋め型問題の大問定義書記述例

(4) ゲームブック型問題

ゲームブック型問題とは、受験者の解答結果に応じて次

に表示する小問を可変にできる、以下の特徴を持つ中間である。

- ▶ 中間の単位で一つのゲームブック型問題を構成する（大問定義書の中間に、ゲームブック型の問題であることを示すタグを記述する）。
- ▶ ラジオボタン選択肢型、チェックボックス選択肢型、プルダウンメニュー選択肢型、記述型、短冊型の形態の小問を出題可能である。
- ▶ 小問単位で PC ブラウザ上に問題を表示する。
- ▶ 小問毎に以下に示す「次小問表示ルール」を設定する。
 - ラジオボタン選択肢型、チェックボックス選択肢型、プルダウンメニュー選択肢型の小問では、解答結果毎に次に表示する小問名を記述する。
 - 記述型、短冊型の小問では、次に表示する小問名のみを記述する。即ち、解答結果による次の問題表示内容の変更を行わない。
 - 次に表示する小問名を記述しないルールの設定が可能で、そのルールを実行すると、そのゲームブック型問題を終了する。
- ▶ 終了したゲームブック型問題は、再表示できない。即ち、解答内容を修正できない。

本システムでは、ゲームブック型問題の各小問の問題形態毎に、記述型は解答欄に入力した値を、プルダウンメニュー選択肢型とラジオボタン選択肢型は解答欄の選択肢に設定された値（value タグ設定値）を、チェックボックス選択肢型と短冊型は解答欄の選択肢に設定された値（value タグ設定値）を JSON 形式の配列にして、それぞれ PC ブラウザからサーバに送信する。

尚、大問定義書のゲームブック型問題の小問の問題文中に文字列を値とする変数を記述することができ、問題文にその変数の設定値を表示できる。その変数の値の設定は、上記「次小問表示ルール中」に記述する。これにより、問題文の一部のみ異なる多数の小問で構成されるゲームブック型問題では、変更分を変数化することにより、大問定義書の記述量を削減することができる。

(5) ランダム出題型問題

ランダム出題型問題とは、受験者毎に異なる中間を出題する形態の大問で、以下の特徴を持つ。

- ▶ 大問の単位でランダム出題型の問題とし、予め作成した中間の問題群（問題プール）から受験者毎に指定数の中間を選択して出題する。
- ▶ 受験者毎に出題する中間の問題プールからの選択は、以下を行う。
 - 問題プールの各中間を大分類（例えば、プログラミングと非プログラミングの2分類）と小分類（類似の問題内容等で分類）の2階層に分類してグループ化し、ランダム出題型問題の大問定義書に記述する。
 - 受験者に出題する中間数分の小分類名を、出題順に

出題リストとしてランダム出題型問題の大問定義書に記述する。

- 出題リストに記述された大分類名の順に、その大分類に含まれる小分類をランダムに選択し、さらに選択した小分類に含まれる中間をランダムに選択して出題する。ただし、同じ大分類名が出題リストに出現した場合、その大分類に含まれ既に出題した小分類と重複しないように小分類を選択する。即ち、大分類に含まれる小分類数は、出題リスト中のその大分類の出題数以上であることを条件とする。

- 各中間は、ラジオボタン選択肢型、チェックボックス選択肢型、プルダウンメニュー選択肢型、記述型、短冊型の小問で構成することが可能で、さらに穴埋め型、ゲームブック型の中間とすることが可能である。

本システムでは、上記の制御を以下のように行う。

- 問題プールの中問を、中間数が1の大問定義書として作成する。
- 試験問題定義書のランダム出題型の大問に、ランダム出題型を示すタグを記述し、問題プール部と出題リスト部を設けて以下のように記述する(図4-11)。
 - 問題プール部に出題対象の中問の大問定義書毎に大分類名とその大分類に属する小分類名、その中間識別用のID(問題ID)を付与して、問題プールに含まれる問題数分の全ての大問定義書を記述する。
 - 出題リスト部に、大分類名を出題順に、出題する中間数だけ記述する。
- 受験者IDをハッシュキーにして、出題リスト部に記述した順に、問題プール部の大問定義書を選択して、受験者に出題する中間を決定する。

本システムでは、ランダム出題型の大問を構成する中間・小問の問題の形態に従って、PCブラウザからサーバに解答を送信する。

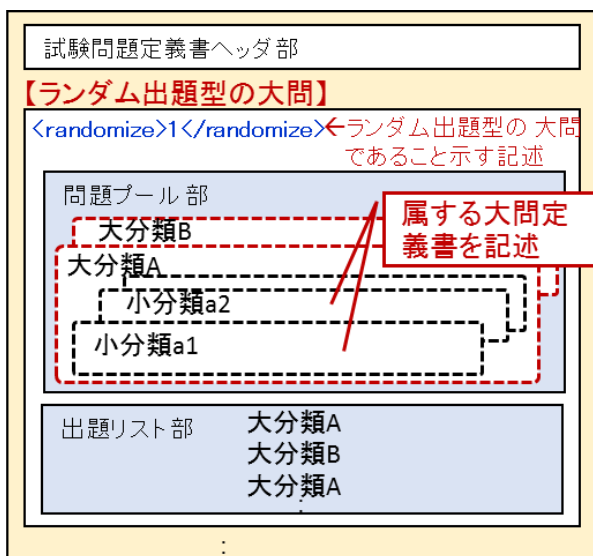


図 4-11 ランダム出題型大問を含む試験問題定義書記述例

4.7 採点・集計

本システムは、実施した試験毎に、全受験者の解答結果を表4-9に示すCSV形式のテキストファイルに出力する。この操作は採点者権限を持ったアカウントで実行可能である。

表 4-9 全受験者の解答結果出力内容

項目	説明
試験 ID	本システムが付与した試験を特定する情報。
受験者 ID	受験者 ID。
受験者の解答内容	受験者の解答箇所毎に以下の3情報を連続して出力する。 ・問題を特定するコード(問題コード) ・解答内容 ・採点結果(採点前なので常に「0」)

表4-9における「受験者の解答内容」には、受験者のPCブラウザから本システムのサーバが受け取った内容をそのまま出力する。ただし、ゲームブック型問題では、どのような順でその問題に解答したかが重要と考え、ゲームブック型問題の中間毎に本システムのサーバで解答履歴を保存し、問題コードと解答内容を要素とするJSON形式の配列で以下のように出力する。

(出力例)

```
[["最初に解答した問題コード":解答内容],[2番目に解答した問題コード":解答内容],...,[最後に解答した問題コード":解答内容]]
```

5. 開発した CBT システムの動作評価

本システムを用いて、表5-1と表5-2に示す条件で2018年8月に東京大学で実施した実証実験において、本システムのAPサーバとDBサーバの負荷状況を図5-1に示す。図5-1では、横軸に模試実施時刻の経過を、縦軸に各サーバのロードアベレージを、それぞれ示している。

実証実験は、2017年度に2大学と11高校、2018年度に2大学と8高校で実施したが、本システムの運用上でのトラブルは報告されていない。

表 5-1 実証実験での試験問題の内容

科目名等	出題の内容
必須:情報セット1	ランダム出題型問題で、全72問の問題プールから15問出題する試験問題。
選択:情報セット2a	システムのデバッグ/動作検証を行うゲームブック型問題と、システム設計を行う試験問題。
選択:情報セット2b	ブロック形式プログラム実行型の問題と、短冊型問題による複数行のSQL文を作成する試験問題。

表 5-2 実証実験の実施方法

実施試験	受験者数	備考
情報セット1と情報セット2a	24人	約90分の試験時間中、一斉に情報セット1を実施後、一斉に情報セット2aまたは情報セット2bを実施する。
情報セット1と情報セット2b	23人	

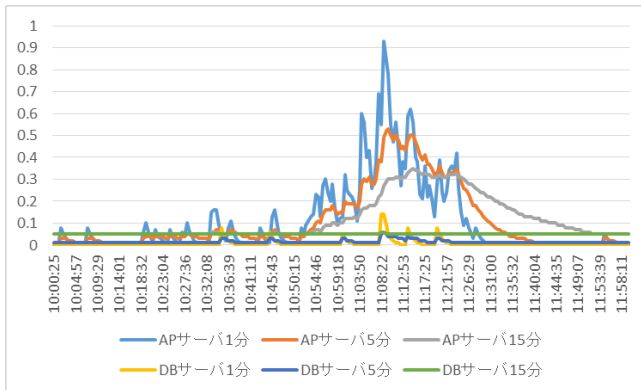


図 5-1 サーバの負荷状況

6. おわりに

本稿では、文部科学省から受託した大学入学者選抜改革推進委託事業（情報分野）において開発した CBT システムについて、その開発の方針を示し、それに基づく全体の具体的な開発・実装の概要、開発した CBT システムを用いて実施した実証実験でのサーバ負荷状況等を報告した。

本システムで実際に開発したプログラムの規模は約 12KL である。

開発した CBT システムでは、試験問題を、仕様化された大問定義書や試験問題定義書に XML 形式のテキストファイルとして記述する。この方式は本事業で研究した「CBT ならでは」の形態の模擬試験問題を、CBT で様々なバリエーションで出題するには有効な手段である。しかし、本稿で紹介した大問定義書や試験問題定義書の仕様はまだ初期段階のものであり、例えば、出題する全ての大問（大問定義書）の内容を試験問題定義書中に展開して記述する必要がある。そのため、大問定義書に修正が発生した場合、その都度、その大問を含む試験問題定義書の全てに修正した大問の内容を反映する必要がある、運用上問題がある。特に、現在のこの仕様のままで、CBT の今後の有効な適用形態の一つで、大量の大問レベルの問題を使用する項目応答理論 (IRT) への本格的な適用には現実的でない。これらは今後の検討課題とする。

謝辞 本稿は、平成 28 年度から平成 30 年度に実施した文部科学省大学入学者選抜改革推進委託事業（情報分野）の研究成果の一部です。本プロジェクトに関わったすべての方々に感謝いたします。

参考文献

- [1] Open Assessment Technologies: TA0. <<https://www.taotesting.com>> (参照 2020 年 1 月 10 日).
- [2] University of Cambridge The Psychometrics Centre: Concerto. <<https://concertoplatform.com>> (参照 2020 年 1 月 10 日).
- [3] 吉岡俊正: 医学系共用試験 CBT の医学部における取り組み, 日本テスト学会 第 1 回研究会 (2004).

- <http://www.jartest.jp/pdf/jirei1_1.pdf> (参照 2020 年 1 月 10 日).
- [4] 村木英治: コンピュータ版テスト (CBT) の実施と理論研究, 計測と制御, 第 40 巻, 第 8 号, pp. 549-554 (2001).
 - [5] 久野靖: 思考力・判断力・表現力を測るには?, 情報処理, Vol. 58, No. 8, pp. 733-736 (2017).
 - [6] 西田知博 他: 「情報科」大学入学者選抜における CBT システムの研究開発, SSS2017 情報処理学会情報教育シンポジウム論文集, pp. 182-187 (2017).
 - [7] 萩原兼一: 2025 年度高校教科「情報」の入試における思考力・判断力・表現力の評価の考え方と方法, 実教情報教育資料 No. 46, pp. 1-8, (Apr. 2018).
 - [8] 文部科学省 大学入学者選抜改革推進委託事業 情報分野: CBT-V2 試験問題, <http://www.mext.go.jp/component/a_menu/education/detail/_icsFiles/afieldfile/2019/03/13/1413650_010_1.pdf> (参照 2020 年 1 月 10 日).
 - [9] 西田知博 他: 「情報科」の大学入試実施のための CBT システムの開発とその検討, 大学入試研究ジャーナル 第 29 号, pp. 117-123 (2019).
 - [10] 西田知博 他: 「情報科」大学入試実施のための CBT システム V2 と試行試験, SSS2019 情報処理学会情報教育シンポジウム論文集, pp. 226-233 (2019).
 - [11] 中野由章 他: 「大学情報入試全国模擬試験」の実施と評価, SSS2014 情報処理学会情報教育シンポジウム論文集, pp. 11-17 (2014).
 - [12] 谷聖一 他: 「第 3 回・第 4 回大学情報入試全国模擬試験」の実施と評価, SSS2016 情報処理学会情報教育シンポジウム論文集, pp. 7-14 (2016).
 - [13] Google: Blockly. <<https://developers.google.com/blockly>> (参照 2020 年 1 月 13 日).