
Web システムにおける視覚効果の向上

富士ソフトABC 株式会社

■ 執筆者Profile ■



高野 啓

1995年 富士ソフトABC株式会社 入社

2002年10月 現在

IT事業本部第3システム部

第6技術グループ 課長



入江内 弥

1997年 富士ソフトABC株式会社 入社

2002年10月 現在

IT事業本部第3システム部

第6技術グループ リーダー



深津 賢治

1996年 富士ソフトABC株式会社 入社

2002年10月 現在

IT事業本部第3システム部

第6技術グループ

■ 論文要旨 ■

一般に、Web アプリケーションは Windows アプリケーションに比べ、操作性や見栄えの点で劣っている。また、ユーザの操作を制御することが難しく、予期せぬ操作によって想定外の結果を招くこともある。その一方で、インターネットの普及に伴い、不特定多数のユーザに低コストで公開できるという Web アプリケーションの利点は、ますます大きくなっている。

こうした状況の中、児童向け学力試験をインターネットで行うシステムの案件を受注した。

そのシステムでは、画像、アニメーション、音声を使った、視覚効果及び表現力が要求されたが、Web アプリケーションにおける問題点は、Web ブラウザのプラグインや JavaScript を利用することによって、解決することができた。

今後の課題としては、プラットフォームによる違い、セキュリティ、サーバ側実装の検討、などが挙げられる。

■ 論文目次 ■

1. はじめに	《 4》
2. システム概要	《 4》
2. 1 方式検討	
2. 2 ソフトウェア構成	
3. 実装方式の検討	《 6》
3. 1 Flash ファイルの構成	
3. 2 ActionScriptの利用	
3. 3 画像ファイルの読み込み	
3. 4 音声ファイルの読み込み	
3. 5 音声形式	
3. 6 音声のストリーミング再生	
3. 7 正解の保持方法	
3. 8 コンテンツのダウンロード方法	
4. Webブラウザのユーザ操作制限	《 10》
4. 1 別ウインドウの起動	
4. 2 履歴を残さない画面遷移	
5. Macromedia Flashプレイヤーの特徴	《 11》
5. 1 ユーザビリティ	
5. 2 プラットフォーム非依存	
5. 3 セキュリティ	
6. むすび	《 12》

■ 図表一覧 ■

図1 ソフトウェア構成図	《 6》
図2 Flashファイル構成図	《 6》
図3 画像ファイルの取り込み	《 8》
図4 音声ファイルの取り込み	《 8》
表1 音声ファイルの圧縮	《 8》

1. はじめに

当社は 1970 年の創立以来、独立系ソフトハウスとして、特定分野に偏らないソフトウェアの受託開発を行ってきた。近年のインターネットの普及に伴い、Web アプリケーションの開発も行うようになったが、業務アプリケーション的性質のものが多く、見栄えについては比較的重要視されなかった。

そのような中で、児童向け学力試験をインターネットで行うシステムの案件を受注した。挙げられたシステム要件のうち、Web アプリケーションの長所と短所に関わる点として、次のようなものがあった。

- (1) インターネット経由 Web ブラウザ上で動作
インターネット上に公開することで、不特定多数のユーザが利用可能とする。
- (2) 解答制限時間
問題には制限時間を設け、所定の時間が経過すると、自動的に次の問題へ進むようにする。学力試験であるため、解答時間の厳密な制御が必要とされる。
- (3) 音声、画像による出題
英語のヒアリング問題に、音声を使用する。また、対象ユーザが児童であるため、見た目に楽しめる画面にし、解答するとアニメーションするなど、動きを付ける。
- (4) ユーザ操作の制御
途中で Web ブラウザを閉じたり、回線切断などによって試験が中断したりしても、再開時に前回中断したところから開始できる。また、Web ブラウザの「戻る」ボタンなどによって、前問へ戻ることができないようにする。

本論文では、これらの要件を実現するための環境及び実装技法について、検討した内容を説明する。

2. システム概要

2. 1 方式検討

Web ブラウザ上で動作するアプリケーションを作成する手段として、HTML、アプレット、プラグインを使用する方法が考えられる。初めに、本システムの要件「インターネット経由」「解答制限時間」「音声、画像」を実現可能か、それぞれの方法について検討を行った。「ユーザ操作の制御」については、4 章で検討を行う。

- (1) HTML 方式
Web サーバ上のプログラム (CGI) を起動して HTML を生成し、クライアント側の動作は HTML 中のスクリプトで記述する。
一般的な Web アプリケーションであり、HTML とスクリプトだけを使用するので、一般

に公開するという点では他の方式よりも有利といえる。

しかし、画面の書き換え時にサーバとの通信が発生する点、及び書き換えの間ユーザが操作できなくなる点から、解答時間を厳密に扱うことができない。更に、表現力に欠けるなど、機能要件を満たすことが困難である。

(2) アプレット方式

Web サーバから Java アプレットをダウンロードして、クライアント側で実行する。

アプレットの実行には Java 仮想マシンが必要となるが、多くの PC にインストール済みであるため、一般公開という点で許容可能といえる。また、画面を更新する際にサーバとの通信や画面全体の再描画は不要なので、HTML 方式のようにユーザの操作を遮ることがなく、解答時間の制御の点でも問題ない。

音声・画像については扱うことは可能だが、視覚効果が特に用意されていないので、回転・拡大縮小などを実現するには、開発工数がかかる上に高度なスキルが要求される。また、一般に Java 仮想マシンは動作が遅いので、実用的な速度を得られない恐れがある。

(3) プラグイン方式

Web ブラウザのプラグインを使用し、データをダウンロードしてクライアント側で実行する。

クライアント側のソフトウェアインストールが必要な点はアプレット方式と同じだが、ものによっては Java 仮想マシン以上に普及している。そのようなプラグインを採用すれば、アプレット方式同様、許容可能といえる。また、画面描画はアプレット同様プラグイン内部で完結するので、解答時間制御についても問題ない。

画像や音声については、Web ブラウザ上で再生することに特化したプラグインが存在するので、それらから選ぶことで画像の回転、拡大縮小、半透明など、豊富な視覚効果を簡単に実現可能になる。音声も、圧縮形式のサポートや、ストリーミング再生など、高度な機能が用意されている。

以上から、広く普及し、かつ画像と音声の再生能力が高いという条件付きであるが、本システムの要件を満たすには、プラグインが最も適しているといえる。この条件に合致するものとして選択した結果、Macromedia Flash プレイヤーを採用することにした。Macromedia Flash プレイヤーは、Windows 98, Windows XP, Apple Macintosh OSをはじめ、多くのコンピュータや Web ブラウザにプレインストールされているため、ほとんどの環境で利用できる。また、画像の回転、拡大縮小、半透明など、豊富な視覚効果を簡単に実現できる。音声も、圧縮形式のサポートや、ストリーミング再生など、高度な機能が用意されている。

2. 2 ソフトウェア構成

サーバ側のソフトウェアは、以下の構成で検証した。

OS	: TurboLinux 7
Web サーバ	: Apache 1.3.20, Tomcat 3.2.4

DB サーバ : Sybase Adaptive Server Enterprise 11.0.3 for Linux
開発言語 : J2EE, Java サブレット, Flash ActionScript

クライアント側のソフトウェアは、以下の構成で検証した。

OS : Windows98/2000
Web ブラウザ : Netscape Navigator 4.7, Microsoft Internet Explorer 5.5
プラグイン : Macromedia Flash プレイヤー 5.0.r42

サーバ・クライアント間のソフトウェア構成は、**図 1**のとおりである。

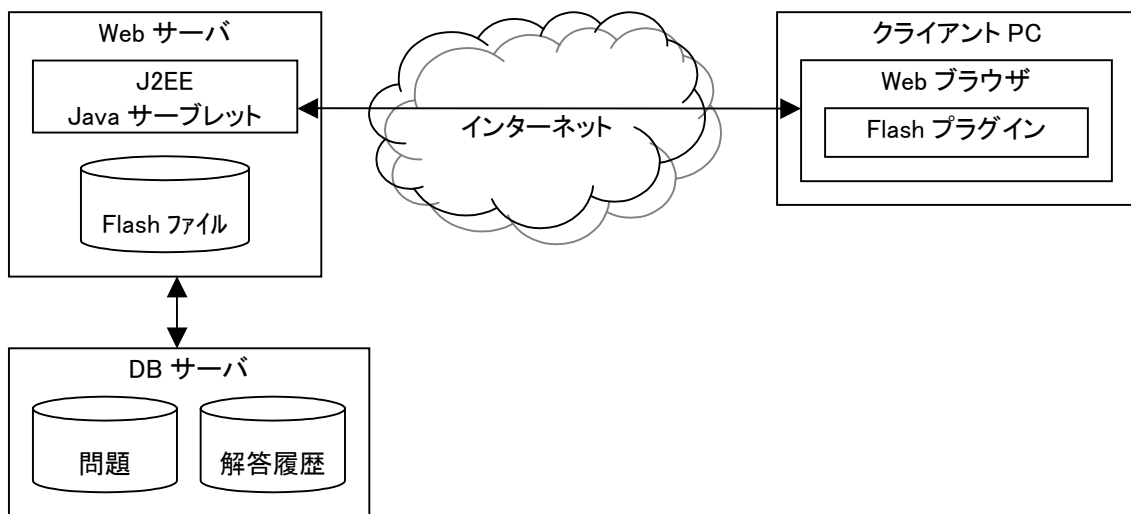


図 1 ソフトウェア構成図

3. 実装方式の検討

ここでは、Macromedia Flash 上に実装するアプリケーションについて検討する。

3. 1 Flash ファイルの構成

Flash では、プラグインに読み込ませるスクリプト(ActionScript)や画像/音声を含むコンパイル済みファイルはすべて、**図 2**のとおり Macromedia Flash プレイヤーファイル(拡張子.swf, 以降「Flash ファイル」と記述)として作成される。

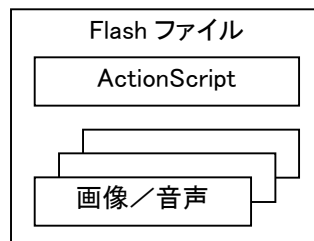


図 2 Flash ファイル構成図

3. 2 ActionScript の利用

Flash に用意されている ActionScript を使用することで、以下のような操作が可能である。

- 画像の表示
- 画像の拡大／縮小
- 音声，アニメーションの再生開始，停止，一時停止
- マウスイベントに反応するボタンの作成
- キーボード入力の検出
- マウスイベントの検出
- タイマの使用

※ActionScript は，JavaScript 言語の仕様である ECMA-262 に準拠している。

これらを組み合わせることによって，本システムに必要な以下の処理が実現できた。

- 音声，画像による出題
- 画像をクリックすることによって，その画像を拡大／縮小しながら移動
- マウス操作によって，画像を移動
- 所定の時間が経過すると自動的に次の問題へ進む

3. 3 画像ファイルの読み込み

前述のとおり，スクリプト，画像，音声などをまとめて一つのファイルに格納したものが，Flash ファイルの基本的な構成である。しかし，本システムでは大量の画像／アニメーションの中から，出題に必要なものを選択して使用する。そのため，これらすべてを1ファイルにまとめるとサイズが大きくなり，ダウンロードにかかる時間が問題になる。そこで，Flash ファイル実行時に動的に画像ファイルを取り込む方法で，必要な画像のみを取り込むことにした (図 3)。

ただし，Macromedia Flash プレイヤーが実行時に取り込めるのは Flash ファイルだけなので，純粋な画像ファイルは取り込むことができない。この点は，個々の画像ファイルをそれぞれ Flash ファイルとしてあらかじめ作成しておき，必要なデータのみ取り込むようにすることで，解決とした。

一応の解決とはなったが，画像ファイルの数が多かったため，各画像ファイルから Flash ファイルを作成するには，かなりの手間がかかった。また，今回のシステムでは関係しないが，URL によって指定されたリソースを取り込めれば，インターネットという巨大なコンテンツの供給源を使えることになり，利用価値は大きい。今後のプラグインには，インターネット上で使われる主なファイル形式を，動的かつシームレスに取り込める機能が望まれる。

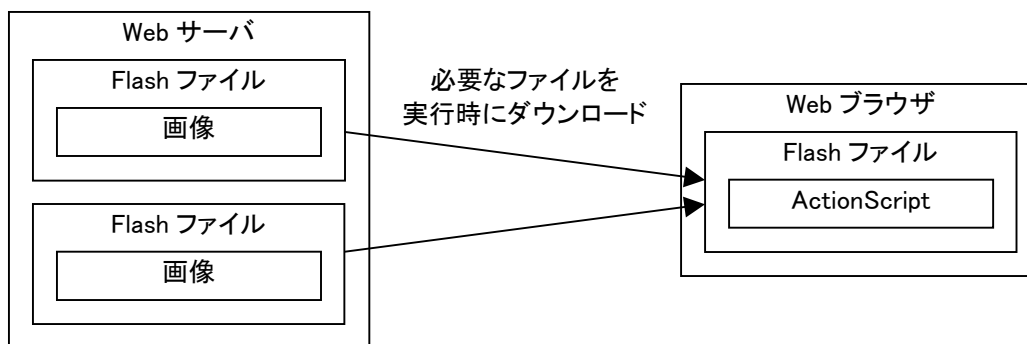


図3 画像ファイルの取り込み

3.4 音声ファイルの読み込み

音声ファイルについても画像ファイルと同様、ファイル一つにつき Flash ファイルを一つ作成することで、必要なデータのみを読み込み可能である (図4)。

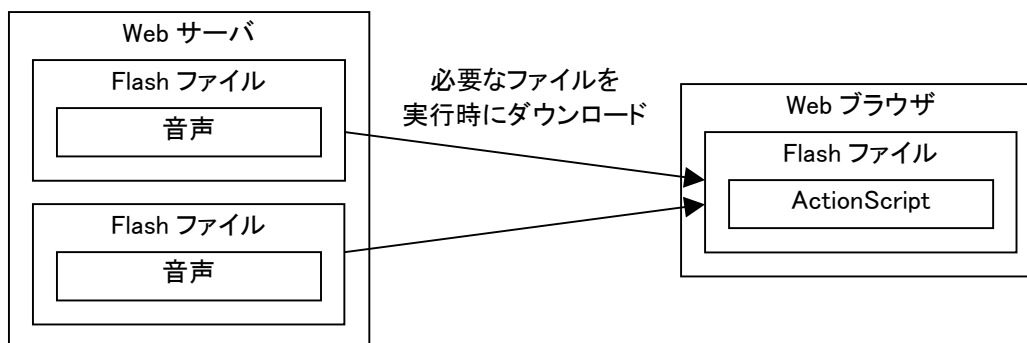


図4 音声ファイルの取り込み

3.5 音声形式

Flash には Flash ファイル作成時に音声を MP3 圧縮する機能があるので、WAV 形式で作成した、元となる音声ファイルよりも、ダウンロード時のファイルサイズを小さくすることができる。

表1は、22[kHz]、16ビットモノラルのWAVファイル25個を変換した結果である。

表1 音声ファイルの圧縮

	形式	ビットレート	合計ファイルサイズ
変換前	WAV	44.1[KB/s]	1013.9[KB]
変換後	SWF (音声をMP3圧縮)	16[kbps] (2[KB/s])	58.7[KB]

Flash ファイルに変換することによって、ヘッダの分だけファイルサイズが増加するが、ビットレートを色々変更して変換したところ、大きくても50~100バイト程度であった。

MP3はビットレートを下げることでファイルサイズを小さくすることができ、16[kbps]では音に歪みが生じるものの、何を言っているか聞き取るだけなら十分な音質が得られた。しかし、本システムには英会話のヒアリングがあり、できるだけ高音質であることが要求

された。音声は長いものになると 30 秒にもなり、ユーザのダウンロード待ち時間が問題となった。これに対処するため、試験を始める前にユーザが ADSL か ISDN かを選択できるようにし、ADSL の場合には高音質の 80[kbps]、ISDN の場合には低音質の 40[kbps]の音声を使用するようにした。

Flash ファイル作成時に予め取り込んである音声は、ActionScript で任意のタイミングに再生、停止させることができる。しかし、単純に音声だけの Flash ファイルを実行時に読み込んだ場合、読み込み完了と同時に自動的に再生が始まってしまう。これでは、ボタンを押したときに再生するような音声も、読み込み完了時に再生されてしまい、都合が悪い。そこで、音声だけの Flash ファイルに ActionScript を記述し、開始と同時に一時停止するようにした。そしてボタン押下時などに、読み込んだ方の Flash ファイルから、音声だけの Flash ファイルの一時停止を解除するようにした。このようにすることで、実行時に動的に取り込みつつ、再生タイミングを制御することが可能になった。

音声を繰り返して再生する、または複数の音声を切れ目なく連続して再生する場合、再生の終わりを認識する必要がある。しかし Flash では、音声の再生終了を検知する機能を持っていない。そのため、サーバから取得するデータに「音声の長さ」を含め、再生開始から指定時間が経過した時点で再生終了とみなすことで、繰り返し再生、及び連続再生を実現した。

3. 6 音声のストリーミング再生

Flash では音声のストリーミング再生が可能であり、音声が始まるまでの待ち時間を短縮する方法として、採用するか検討を行った。しかしストリーミング再生では、想定した通信速度を下回ってバッファが空になった場合など、回線状況によっては再生が途切れてしまう可能性がある。本システムでは、音声のヒアリングによる出題があり、途切れは許されなかったため、待ち時間が長くなってもすべてダウンロードした後で再生することとした。

3. 7 正解の保持方法

本システムでは設問をランダムに作成するので、解答の正否判定を毎回行う必要がある。正解を保持する方法としては、クライアント側で持つかサーバ側で持つかに分かれる。

(1) クライアント側で持つ方法

クライアントに、問題だけでなく正解情報も送信し、ユーザの解答時に、解答と一緒に正解もサーバへ送信する。

(2) サーバ側で持つ方法

Web サーバのセッション管理機能を使用して、作成した問題と正解をサーバ側で保管する。この場合には、ユーザの解答履歴の保管も容易である。また、クライアント側は正解情報を受け取ることがないので、ユーザに対して正解を完全に秘密にすることができる。

検討の結果、セッション管理を行う必要はあるがデータの保管が容易で、正解を完全に

秘密にすることができるので、(2)のサーバ側で正解を保持する方式にした。

この場合に考慮が必要な点としては、セッション ID をサーバ・クライアント間で受け渡す点、及びセッションタイムアウト時間を設定する点が挙げられる。セッション ID の受け渡しは、インターネットを通して不特定のユーザがアクセスする場合を考え、クッキーの使用は避け、URL に問い合わせ文字列の形で乗せて行うことにした。

3. 8 コンテンツのダウンロード方法

本システムで必要なファイルをすべて最初に一括ダウンロードするか、必要な都度ダウンロードするかの検討を行った。一括ダウンロードが可能であれば、解答中は電話回線を切断しておけるという利点がある。以下は、一括ダウンロードをテストした結果の、パフォーマンスと問題点である。

(1) ダウンロード時間

プロトタイプとして作成した複数形式分を出題するための全ファイル容量は、4688[KB]であった。これを ISDN の理論値である 64[kbps]でダウンロードした場合、 $586 \text{ 秒} = 9 \text{ 分 } 46 \text{ 秒}$ かかる計算になる。

(2) メモリ使用量

上記の構成でダウンロードを実行した結果、Web ブラウザと Macromedia Flash プレイヤーが使用するメモリが、50[MB]近く増加した。この使用量では、あまりメモリを積んでいないマシンでメモリスワップが頻発し、OS 自体がまともに動作しなくなるおそれがある。

(3) CPU 使用率

大量のコンテンツを保持し、それらの重なりなどを処理しているためか、CPU 使用率が全体的に高くなり、頻繁に 100%に達して動作がぎこちなくなった。

結論として、問題点が多いためコンテンツの一括ダウンロードは断念し、必要な都度ダウンロードする方式とした。

Flash はファイルを非同期にダウンロードすることができるので、ユーザが解答している間に次の問題の画像ファイルと音声ファイルをダウンロードする仕組みを作成した。これによって通信回線を無駄なく働かせ、各問題の開始前のダウンロード待ち時間を最小にした。

4. Web ブラウザのユーザ操作制限

本システムでは、ユーザが Web ブラウザの「戻る」「進む」「更新」「閉じる」などの操作を行った場合についても、問題／解答の二重送信や欠落などの不整合が起こらないように対処する必要がある。これを実現するにはサーバ側の機能も必要であるが、クライアント側での対処を中心に述べる。

ここでは、JavaScript を使用してユーザが「戻る」などの操作を行う機会を減らす方法

について検討する。これは、Web ブラウザの設定で JavaScript を有効にしておくことを前提としている。

4. 1 別ウィンドウの起動

JavaScript を使って別ウィンドウを起動する場合、新しく作成されるウィンドウのメニューバーやツールバーの、表示／非表示を指定することができる。ここでメニューバー・ツールバーを非表示にすることで、ユーザが故意や不注意で「戻る」「進む」「更新」を行わないようにする。

4. 2 履歴を残さない画面遷移

上記の方法によってメニューバーなどを非表示にした場合でも、右クリックメニューやショートカットキーによって、「戻る」などを行うことができる。JavaScript を使って履歴を残さずに画面遷移することで、これらの方法を使って前画面に戻ることを防ぐ。

この方法は、履歴自体を消去しているわけではないため、本システムのページを表示する前に別のページを表示していた場合、そのページに対する「戻る」までは無効にならない。しかし、上記の別ウィンドウを起動する方法と併用することで、履歴が空の状態から本システムを開始できるので、一切の「戻る」操作を行えないようにすることができる。

5. Macromedia Flash プレイヤーの特徴

5. 1 ユーザビリティ

Web ブラウザの基本機能では、ドラッグ・アンド・ドロップなどの直感的な操作を実現することができず、Web アプリケーションのユーザ・インターフェースが使いづらいものになる要因となっている。また、画面の一部を更新する場合に、画面の再読み込みと再描画を行うことになり、ユーザの操作の流れを遮ってしまう。

Macromedia Flash プレイヤーを使った Web アプリケーションの場合、ドラッグ・アンド・ドロップなど、使いやすいインターフェースを実現できる。また、画面を書き換える場合にも、サーバとの間はデータのみをやりとりし、画面全体の再描画は発生しない。

5. 2 プラットフォーム非依存

Web アプリケーション開発において、プラットフォームは常に問題になる。普及している主な Web ブラウザとして Internet Explorer と Netscape Navigator の二種類があり、しかもバージョンが異なると同じ種類の Web ブラウザであっても互換性が完全ではないからである。そのため、Web ブラウザの種類を限定して開発を行ったり、内部を Web ブラウザごとの処理に分割して実行時に分岐させたりして対応することになる。

Macromedia Flash プレイヤーは、Internet Explorer と Netscape Navigator の両方に対応しており、なおかつ Windows, Macintosh だけでなく、Linux などそのほかの OS 向けバージョンも提供されている。このプラットフォーム非依存のおかげで、対象ユーザを絞り込むことなく、Web ブラウザごとの処理分割や動作確認試験などの開発工数を削減することができる。

5. 3 セキュリティ

Macromedia Flash プレイヤーには、動作しているコンピュータにローカルファイルを書き込む機能がない。また、サーバへ情報を送信する機能があるが、送信先は Flash ファイルと同じサブドメイン内でなければならず、他のサーバへ送信することはできない。このように、情報の破壊・改竄・漏洩を防ぎ、セキュリティを実現している。

6. むすび

Web アプリケーションの操作性や見栄えを向上させるため、本システムでは Macromedia Flash プレイヤーと JavaScript を利用した。これによって要件を解決した方法は、以下のとおりである。

(1) インターネット経由 Web ブラウザ上で動作

プラグインの普及具合について調査した結果、Macromedia Flash プレイヤーはほとんどの OS、ブラウザにプレインストールされており、普及率は想像以上であった。ほとんどの環境で使用できるため、インターネット経由である点が Flash を使う上での障害にはならないことを確認できた。

(2) 解答制限時間

Macromedia Flash プレイヤーでは、描画をプラグイン内部で行い、その際にサーバとの通信や画面全体の書き換えは不要である。そのため、ユーザの操作を遮ることがなく、解答制限時間を厳密に扱うことができた。

(3) 音声、画像による出題

Flash の音声を MP3 圧縮する機能によって、ダウンロードするファイルサイズを小さくすることができた。また、圧縮するビットレートを変えられるので、音質とファイルサイズを調整して、回線速度の違う環境に柔軟に対応することができた。

画像についても、回転、拡大縮小しながらの移動など、Flash の ActionScript を使うことで、画面に動きを付けることが簡単に行えた。

画像や音声などのファイルは一括してダウンロードするのではなく、必要な都度ダウンロードするようにした。その結果、不要なコンテンツをダウンロードする必要がなくなり、ダウンロード時間だけでなく CPU、メモリ使用量を適正にすることができた。また、音声と画像を多用した結果、ダウンロードするデータサイズが大きくなったが、非同期にダウンロードすることで待ち時間を最小にした。

(4) ユーザ操作の制御

JavaScript を使って、ユーザが Web ブラウザの「戻る」「進む」「更新」「閉じる」などの操作を行った場合についても、問題／解答の二重送信や欠落などの不整合が起これないように対処することができた。

今後の検討課題として、以下の項目が挙げられる。

(1) プラットフォームによる違い

今回は、Windows 上の Netscape Navigator と Microsoft Internet Explorer での動作確認を行ったが、Windows のバージョンや Web ブラウザのバージョンによる違いについて細かな検証を行っていない。今後はこのような点についても検証していく必要がある。また、Macintosh など Windows 以外のプラットフォームでの実証が必要である。

(2) セキュリティの考慮

本システムでは、試験成績という重要なプライバシーに関わる情報を扱う。個人情報保護の観点から、サーバ・クライアント間の通信データについて、SSL の実装などセキュリティを考慮する必要がある。

(3) サーバ側実装の検討

今回は、クライアント側の実装を中心に検討した。今後はサーバ側の実装についての検討が必要である。