

英作文自動添削システム「サッと英作！」のJavaScriptへの移植

西 村 則 久

Porting “Satto Eisaku” (Automatic Correcting System for English Composition) to JavaScript

Norihisa NISHIMURA

要 旨

英作文自動添削システム「サッと英作！」は、和文英訳をコンピュータが自動添削するシステムである。安田女子大学においては最近まで共通教育科目の授業の中でほとんどの学生が利用していた。現在も一部の学科で利用されている。このたびシステムの一部を改良（移植）したので報告する。

I. 英作文自動添削システム「サッと英作！」の概要

英作文自動添削システム「サッと英作！」は筆者が大学院修士課程で開発し、安田女子大学の運用実験を経て実用化された、和文英訳をコンピュータが自動添削するシステムである。ソフトウェアの形態はWebアプリケーションである。北辰映電株式会社によって商用化されており、安田女子大学以外の教育機関でも利用されている。

利用者はブラウザ上に提示（出題）された和文を英訳して入力し、添削ボタンを押す。すると、入力された英文を最大限に活かしつつ正しい英訳とするような最適な赤入れ添削がすぐに表示される（図1）。ただしそれは利用者の英訳が不正解の場合であり、正解のときは当然ながら赤入れは行われない。また、赤入れ添削により正しい英訳へと導くのが困難なほど正解から大きくかけ離れている場合にも赤入れは行われない。

本システムは和文を英訳する機能は持っておらず、正解とされる正しい英訳は問題作成時に人の手によってデータ化される。和文英訳の正解は何通りも存在するものであるが、データ化されていない正解が一つでもあると、利用者がそれを入力したときには不正解の扱いになるため、全ての正解が問題内にデータ化されていなければならないことになる。正しい英訳とされるものは、基本的な構文をどうするか、またどの同義語やイディオムを使うかといったことによって、その数はそれらを掛け合わせた数に増大するため、従来は全ての正解をデータ化することはほとんど不可能と思われていた。本システムは、全ての正解を記述することを可能にするためにBUD言語¹という独自の記述法を備えたことが最大の特徴である（図2）。



図1. 「サットと英作！」の添削画面

記述	he took his [child(ren), kid(s)] to the [forest, wood(s)].
含まれる 正解 (12通り)	He took his children to the forest.
	He took his child to the forest.
	He took his kids to the forest.
	He took his kid to the forest.
	He took his children to the woods.
	He took his child to the woods.
	He took his kids to the woods.
	He took his kid to the woods.
	He took his children to the wood.
	He took his child to the wood.
	He took his kids to the wood.
	He took his kid to the wood.

図2. BUD言語の記述の一例

本システムには問題作成ページも用意されており、教師がブラウザ上でBUD言語の記述によりオリジナルの問題を作成できるようになっている。

古き良き時代の授業では、和文英訳の演習というのは、数十人の中から選択された数人の受講者が黒板に英文を書き、それを教師が一つ一つゆっくりと赤入れをするというものであり、例えば20分以上かけて6問程度が添削されるというものであった。選ばれなかった大勢の受講者は、自分が訳した文は添削してもらえず、他人のものを見ているだけであった。

本システムを利用した授業は、受講者全員がネットワークに接続したパソコンを使用できる環境が前提となるが、受講者全員が自分の英文を添削してもらえるうえ、時間の面でも非常に高速である。また、添削は教師が居なくても行えるので、自習用としても利用することができる。

本システムの重要な機能や特徴には上記のほかにも、部分訳問題機能、並べ替え問題機能、コメント表示、問題の階層化、管理ページなどがあるが、ここでは割愛する。

II. 移植に至る経緯

本システムは1999年頃に安田女子大学で運用され始めた。HTML言語だけで赤入れ添削をわ

¹ BUD言語は問題作成者が利用するもので、当然ながら学習者が知る必要は無い。

かり易く表示するのは困難なので、本システムは教室の黒板をモチーフにしたグラフィック表示を行っているが、運用当初はブラウザ上にグラフィックを表示する手段として当時主流であった Java²を用いていた。しかしその後、Javaに代わりFlashが主流になるにつれ、安田女子大学の教室のパソコンにもJava実行環境がインストールされないことになり、本システムの利用は最初の危機を迎えた。そのため、2009年には本システムに幾つかの機能拡張とともにグラフィック表示を全てFlashで行うための大幅な改良を行った。Flashのプレイヤーは全世界での普及率が99%以上と言われるまでになり、安田女子大学の教室のパソコンにもインストールされていたので、本システムはしばらくの間安定的に運用された。しかし近年になって、インターネットの利用形態が変化し、それによって幾つかの問題が生じてきた。

1. スマートホンの台頭

携帯電話が発達してiPhoneやAndroidといったスマートホンが普及すると、スマートホンでインターネットのブラウザを簡便に利用できるようになった。そして、多くの人がインターネットの利用にはパソコンは使わずスマートホンで済ませるようになった。よって、本システムもスマートホンで利用されて然るべき状況となったが、Flashは動作が重くスマートホンには不向きであるとも言われており、Flashのプレイヤーがインストールされたスマートホンは非常に少ないのが現状である。

2. Flashの退廃

パソコン環境でもFlashに代わってJavaScriptをベースとしたHTML5を推す気運が高まり、最もシェアの高いブラウザであるGoogle Chromeでも内蔵のFlashプレイヤーが無効化されることになった。よって、本システムは現状のままだと近い将来パソコン上でも動作しなくなってしまうことが予想される。

以上のことから、本システムのFlashプログラムをJavaScriptへと移植することを決めた。JavaScriptはかつてはブラウザによって言語仕様が異なったり機能が豊富でなかったりしたため、これまで本システムでは採用しなかったが、近年はHTML5の一部としてJavaScriptの高機能化と異ブラウザ間での高い互換性が実現されてきており、本システムを今後パソコン・スマートホンのいずれでも動作させる上では正に必須の選択と言える。

Ⅲ. 移植の概要

本システムには、サーバー側で実行されるプログラムとクライアント（ブラウザ）側で実行されるプログラムがある。Webアプリケーションであるから、クライアント側で実行されるプログラムであってもクライアントにインストールされる訳ではなく、実行時にその都度サーバーから送られてくることになる。この送られてくるプログラムが主に表示に関わるもので、今回FlashからJavaScriptへと移植した部分である。サーバー側で実行されるプログラムはC言語とPerl言語で作られており、その役割は、添削結果の表示内容を表すデータを生成して、ブラウザ

² JavaScriptと名前は似ているが別のものである。

に表示させるプログラム（従来はFlash、移植後はJavaScript）と共にクライアントに渡すことである（図3）。

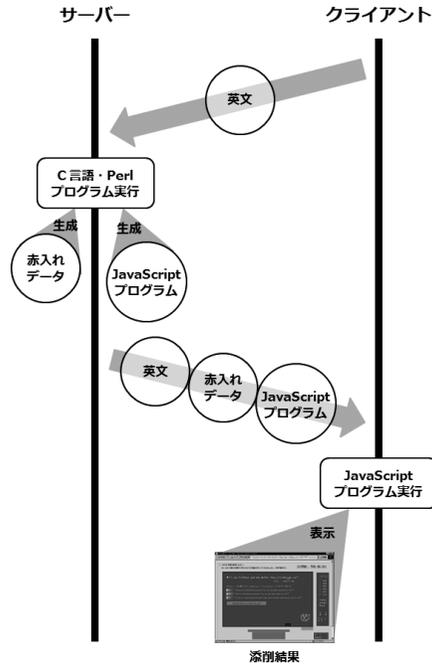


図3. 「サッと英作！」の処理の流れ

IV. 移植の詳細

JavaScriptとFlashのプログラミング言語であるActionScriptはどちらもECMA-262仕様に準拠しており、文法的にも似ているうえ、変数や文字列を操作するライブラリ関数などにもほとんど違いがない。そのため、Flash版のソースコードの大半はそのまま或いは単純な置換をするだけで流用することができた。

よって、移植に必要な開発の大部分は、ボタンなどのUIコンポーネントや文字列を表示するサブルーチンをJavaScriptで作ることであった。Flash版のものと同仕様のサブルーチンであれば、それを利用するメインルーチンには変更の必要がない。

UIコンポーネントはJavaScriptのクラスを利用することも可能であるが、ボタンについては、実行環境（ブラウザ）によって微妙に異なるデザインとなってしまうことや、Flash版にはカラフルで立体的なボタンの作成ルーチンが作ってあったことにより、移植することにした。

サブルーチンの他に必要であった移植作業は、プログラムへの表示データの受け渡しである。FlashではHTMLのタグ内にパラメータを記述してプログラムにデータを渡すが、JavaScriptにはそのような手段がない。そのため、サーバー側のプログラムがクライアントに送るJavaScriptのプログラム内に直に変数の値を記述するようにした。Flashではクライアントのプログラムは常に同じものであったが、今回のJavaScriptプログラムは毎回微妙に異なる内容であることになる。

V. 移植上の難点

移植を行う上で難易度が高かったことや、特に工数を要したことについて述べる。

1. 文字化けの問題

文字列データの受け渡しを行う際に空白文字や制御文字が含まれていると文字列が途切れるなどの問題が起こるので、送り側でそのような心配のない安全な文字列へとエンコードしてから文字列を渡し、受け取り側ではそれをデコードしてから処理している。FlashではActionScriptのescape関数でエンコードし、unescape関数でデコードしていたが、JavaScriptにも同じ役割を持つ同名の関数があるのでそれを使用するようにプログラムを組んだところ、文字化けが発生した。原因を調べたところ、それぞれのescape関数とunescape関数のペアは逆関数となっているが、Flashのescape関数とJavaScriptのescape関数とは異なるアルゴリズムでエンコードをしていることが判明した。そこで、JavaScriptのescape,unescape関数は使用せず、Flashのescape,unescape関数と同仕様の関数をJavaScriptで作成して使用した。

2. ピンチアウト時の文字のぼやけ

スマートホンの特徴の一つにピンチアウト（タッチパネル上で二本の指を広げるように動かす操作）による表示内容の拡大機能があるが、当初、本システムの添削画面をピンチアウトにより拡大させると、文字の輪郭が不鮮明になった。これは、文字列をグラフィック文字で表示させていたためで、いわゆるペイント系の画像となった結果起きた現象であった。輪郭が不鮮明であっても実用上の問題は全く無いが、今回スマートホンでも利用できるようにするためにJavaScriptに移植するのであればこのピンチアウト時の拡大表示にもスマートホンらしい挙動が期待されるはずである。そこで、文字列をグラフィック表示ではなくTEXTAREAというオブジェクトとして表示させて、いわゆるドロー系の表示とすることで、拡大しても輪郭が不鮮明にはならないようにした（ただし後述するように、一部の文字列はグラフィック表示のままにしている）（図4）。

VI. 結 果

英作文自動添削システム「サツと英作！」の表示部をFlashプログラムからJavaScriptへと移植した。これにより将来的に不安のあるFlash環境から離脱することができ、ブラウザにプラグインを追加することなく動作するようになった。また、スマートホンのブラウザ上でも動作するようになったので、利用者にとって、従来よりも非常に手軽に利用することができるようになると期待される。

VII. 今後の課題

本システムの移植はほぼ完了しているものの、細部には未実装の箇所もあるので、完成を急ぎたい。その段階であるため、以下のような細かい課題も残っている。

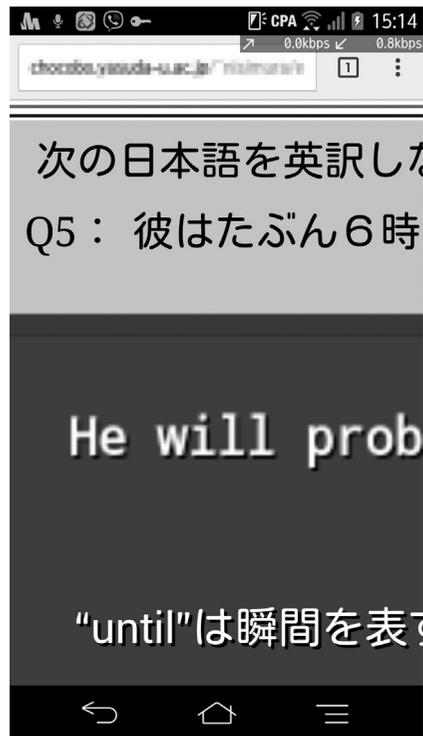


図4. 文字のグラフィック表示（中央がグラフィック表示，その下はオブジェクト）

1. 打ち消しラインの問題

上述のように，文字列は基本的にはドロー系のオブジェクトとして表示している。しかし，添削対象の英文についてはペイント系のグラフィック表示をしている。理由は，赤入れの一つである打ち消しライン（単語に横線を重ねて表示）がグラフィック表示であるために，添削対象の英文をオブジェクトとして表示すると，打ち消しラインが単語の背後に表示されてしまうからである（図5）。打ち消しラインもオブジェクトとして表示することができれば文字の前面に表示させることができるが，打ち消しラインを形作れるオブジェクトが現時点では見つからない。今後この問題を解決する方法を発見して全ての文字列をドロー系オブジェクトとし，鮮明に表示させたいところである。

2. 外部からのアクセス

安田女子大学内のサーバーに筆者がインストールした本システムには，パソコンからでもスマートフォンからでも，学外からアクセスしたときに限り添削することができない（添削ボタンを押してもずっとそのまま，添削結果の画面へと遷移しない）という現象がある。ただしこれは本システムではなくファイアウォールの問題の可能性もあり，原因を調査中である。



図5. 打ち消しラインが単語の背後に表示された状態

参 考 文 献

- 1) 西村則久, 安村通見, 外国語作文における自動添削手法について, 情報処理学会研究報告 (人文科学とコンピュータ), Vol.99, No.5, pp.1-6 (1999 Jan).
- 2) 西村則久, 明関賢太郎, 安村通見, インターネット英作文添削システム, 情報処理学会インタラクシオン'99 論文集, pp.43-44 (1999 Mar).
- 3) 西村則久, 明関賢太郎, 安村通見, 英作文における自動添削システムの構築と評価, 情報処理学会論文誌, Vol.40, No.12, pp.4388-4395 (1999 Dec).
- 4) 明関賢太郎, 西村則久, 安村通見, 状況情報を利用した外国語学習支援システムの設計と開発, 情報処理学会第41回プログラミング・シンポジウム報告集, pp.49-56 (2000 Jan).
- 5) 西村則久, 英作文自動添削システム「サッと英作！」の改良, 安田女子大学紀要 38, 181-186 2010.

[2016. 9. 29 受理]

コントリビュータ：山下 明博 教授 (造形デザイン学科)

