

英作文自動添削システム「サッと英作！」の改良

西 村 則 久

Reform of "Satto Eisaku" - Automatic Correcting System for English Composition

Norihisa NISHIMURA

概 要

英作文自動添削システム「サッと英作！」は、和文の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1年生の授業「英語ライティングⅠ・Ⅱ」¹⁾で採用されているため、殆どの学生が利用している。このたびシステムの一部を大幅に改良したので報告する。

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、安田女子大学での運用実験を経て実用化されたシステムで、同大学が北辰映電株式会社を通して販売している「Yasuda System」の一部であり、外部の教育機関でも利用されている。まず本システムの主要な部分を簡単に説明する。

本システムを利用する学習者は、表示される問題文（和文）を英訳して入力し、添削ボタンを押す。すると自動添削処理が施され、入力した英文が最大限に活かされつつ、正しい英訳となるような赤入れが一瞬にして表示される。ただし、入力した英文が正解の場合や、逆に全くの的外れである場合には、その旨の表示のみで赤入れはない。



図1. 「サッと英作！」の添削画面

1) 計32クラス開講、筆者は担当していない。

一般に和文英訳の正解は一通りではなく、数限りなく存在することも多い。正解／不正解の判定や、入力された英文を最大限に活かした赤入れが常に正しく行われるためには、問題作成者は想定される全ての正解文を作成しておかなければならないことになるが、通常これは不可能な作業と考えられるため、多数の正解に対応した添削システムは他に類を見ない。BUD言語という特殊な記述法²⁾を開発し、想定される全ての正解文の作成を可能にしたのが本システムの技術的特徴である。

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

本システムはWebアプリケーションであり、一台のWebサーバーにインストールするだけで複数のパソコンから同時に利用することができる。CALL教室の授業では、本システムを利用して受講者が自分のペースで次々に添削を受けられる。コンピュータを使わずに教師が添削する授業だと添削できる問題の量に限りがあるので、非常に大きなメリットがある。

ほかに、本システムの重要な機能や特徴には、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. 従来の問題点

本システムは1999年頃から安田女子大学にて運用されてきたが、時の流れとともにパソコンのソフトウェア環境・ハードウェア環境および利用形態が進化すると、いくつかの問題が生じてきた。

1. 環境依存の問題

HTML言語だけで赤入れ添削をわ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。このグラフィック表示には、本システムの運用開始当初にこのようなものを実装する手段としては主流であったJavaアプレットを用いていた。JavaアプレットやJavaアプリケーションは通常のアプリケーションソフトとは異なり同一のバイナリで機種を選ばず実行可能であると謳われ注目されたが、複数メーカーが実行環境を配布したことなどにより、同一のJavaアプレットであっても実行環境が異なると表示が微妙

2) 記述ルールは多岐に渡る。ここでは触れない。

に異なったり、処理速度に大きな差が生じたりした。このことは一般的なJava利用ソフトにおいては大きな問題ではなかったが、本システムのJavaアプレットでは特に顕著で、利用に支障をきたすようなバグ同然の不都合もあった。その結果、本システムはJava実行環境のバージョンを限定してしまうことになった。

そのため、安田女子大学においては、CALL教室および情報教室のパソコンのJava実行環境のバージョンを本システムの都合だけで決めてきたが、より新しいバージョンが次々とリリースされる中、やや難のある運用であった。また、安田女子大学は近年、段階的にノート型パソコンの所有を学生に義務付けてきており、ノート型パソコンの利用が必須の授業もある。よって、学生が空き時間などに自分のノート型パソコンで本システムを利用したいというニーズが生まれたが、ここでもJava実行環境のバージョン限定が問題であった。

平成21年度前期から、導入する他ソフトの都合により、教室のパソコンのJava実行環境を最新のものにすることが決められた。よって、「英語ライティングⅠ・Ⅱ」の授業において、重要な教材である本システムが使えなくなる見通しとなった。

2. 表示サイズの問題

本システムの運用開始当初は、パソコン画面の画素数は横800×縦600ピクセルのものが多かった。この大きさに収まりかつ見やすいように、本システムのグラフィック表示の大きさは横760×縦540ピクセルとしていた。その後、パソコン画面は横1,024×縦768ピクセルが主流となり、更に現在は横1,280×縦1,024ピクセル以上が主流となっている。そのため、本システムのグラフィック表示は相対的に小さく感じられるようになり、パソコン画面の広さを有効に活用していないと言える状態になった。

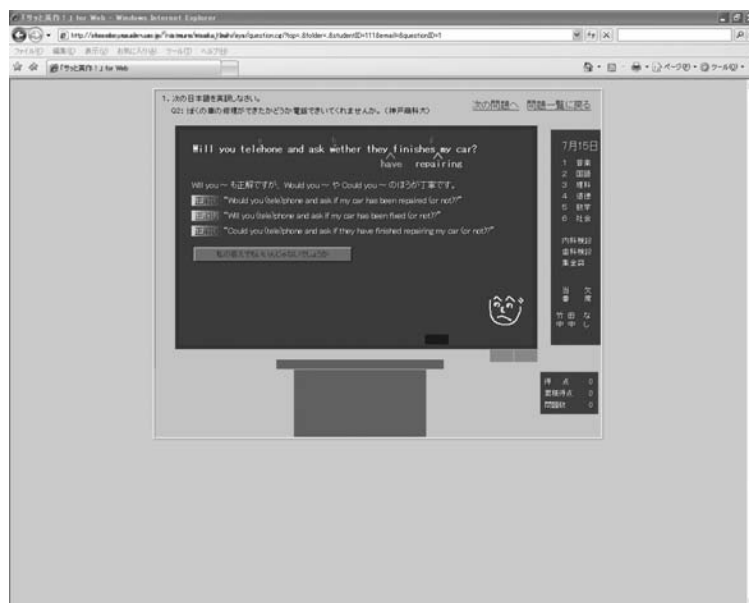


図3. 横1,280×縦1,024ピクセルでの添削画面

Ⅲ. 改良の詳細

環境依存の問題を解決するため、本システムのグラフィック表示部をFlashで作直した。Flashはインターネットのブラウザ上で動作させることが可能で、アニメーション機能などの高い表示能力があり、用途はJavaと重なるところがある。本システムの運用開始当初はFlash実行環境の普及率は低かったが、現在は99%程度と高い。また、未インストールのパソコンにインストールする作業が容易である。Flashの環境には本システムのJavaアプレットが従来抱えていたようなバージョン問題は存在せず、上位互換のバージョンアップを重ねる中でスクリプトによるプログラミング環境が強化され、Javaの用途の一部を奪いつつある。

開発環境はやや古いがMacromedia Flash MXを用いた。開発言語はActionScriptである。本システムのJavaアプレットのソースプログラム（Java言語）の流用は困難で、ゼロからの開発を余儀なくされたが、そのため表示サイズの問題をこの機会に同時に対応すれば工数は増加するが別々の機会に開発するよりも効率が良いので、同時に対応した。

パソコン画面が横1,280×縦1,024ピクセルなら、グラフィック表示は横1,200×縦850ピクセル程度で良いと思われたが、今後のパソコン画面の拡大化や、多様な利用環境を想定し、横608×縦432ピクセルから横1,520×縦1,080ピクセルまでの七段階の何れかを利用者が何時でも選べるようにした。なお、グラフィック表示のサイズを大きくしても、表示される文字やボタンなどの大きさは不変であり、全ての表示内容を単純に拡大しているわけではない。そうすることにより、サイズの大きな画面だと多くの内容を表示できるようになる。これには、単純な拡大表示と比較すれば、多くの開発工数を要する。

Flashはswf形式ファイルで提供される。1個のswf形式ファイルで七段階の表示サイズへの動的な変更は実現できなかったため、7個のswf形式ファイルを組み込んでいる。

Ⅳ. 開発上の難点

JavaとActionScriptは言語仕様が近くないので、本改良は単純作業的なものにはならない。また、論理的処理の移植とは異なり、表示関連の移植には、数多くの解決しなければならないプラットフォーム間の違いが伴うものである。

1. 文字コードの問題

本改良は当初、本システムの中の表示に関わる部分のFlash化のみで終わる見通しであったが、JavaアプレットとFlashはサーバーとのデータ受け渡しにおいて日本語文字コードが異なることが判明し、表示に関わる部分のみならず、その周辺にまでプログラムの変更が大きく及んだ。学習者が利用を途中で中断した際の状況を保存したファイルの中味の文字コードに至るまで互換性を持たせる必要があった。このような開発は、一般的に、動作確認の作業量が非常に多くなる。そのため、当初の予想以上の開発工数を要した。

2. UIコンポーネントの問題

ボタンなどのUIコンポーネントは通常は開発環境に用意されているものを利用するところであるが、Flashに用意されているボタンはモノクロでデザイン的に優れず、本システムにおいて

は不都合と判断したため、これを利用せずに従来Javaで作成していたボタンに近い色とデザインのボタンを独自に作成した。これには影のような縁が描かれており、立体感がある。5個の長方形をずらして描画することで立体感を出し、その上に文字を表示した。ボタンが押されたときには凹んだボタンを表示させるため、更に5個の長方形と文字が必要となり、1個のボタンのために12個のオブジェクトを作成した。

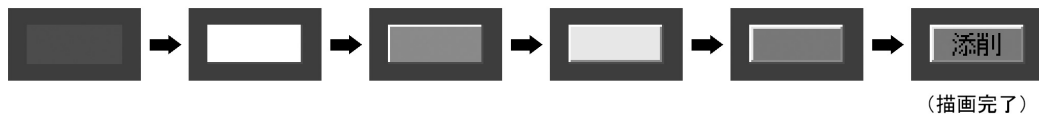


図4. 独自のボタンの描画

独自のボタンなので、押されたときのクリックイベントを検出するためのプログラムも独自に作らなければならない。画面全体のクリックイベントがあったとき、その位置情報を元に、それがボタンの中であるかを調べ、ボタンの中であれば、押されたときに行うべき処理へと移る。ボタンの中でなければ何もしない。

ボタンの他にも、同様の方法で、様々なUIコンポーネントを独自に作成したり、既存のUIコンポーネントを装飾したりした。

V. 結 果

Javaアプレットで行っていたグラフィック表示をFlashによるものへと変更した。これにより、実行環境依存の問題が解消され、ほとんどのパソコンで利用できるようになった。

グラフィック表示のサイズが小さいという問題を解消した。横608×縦432ピクセルから横1,520×縦1,080ピクセルまでの七段階の中から利用者が自由に選択できるようにした。

なお、本改良の前後で表示や使用感の変化は少なく、グラフィック表示部にプラットフォームの変更があったことを利用者に感じさせないものとなっている。

改良に要した期間は約6か月、開発ステップ数は最大の変更箇所であるグラフィック表示のActionScriptだけで1,937ステップ、その他の箇所にも変更がある（ステップ数は不明）。

VI. 今 後 の 課 題

本改良は、平成21年度前期の授業開始に間に合わせる緊急の開発であったため、納期を優先し、あまり使用されない機能（時間制限機能など）は未サポートとした。この問題への対応が今後の課題である。

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, 安村通晃, 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ョ
ン'99論文集, pp.43-44 (1999 Mar).
- 3) 西村則久, 明関賢太郎, 安村通晃, 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, 安村通晃, 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).

[2009. 9. 28 受理]