

# 「XMLによる作文添削情報表示システム」 仕様の発展について —「添削情報電子化」の真の意義を求めて—

宇佐美 洋・鎌水 兼貴

## 1. はじめに

### 1.1. 添削情報を電子化することの意味

2001年に公開した「日本語学習者による日本語作文と、その母語訳との対訳データベース ver.2 (CD-ROM 版)」(以下「作文対訳 DB」)には、日本語教師(日本語母語話者・非母語話者を含む)による添削情報がのべ約 300 編含まれている。これらの添削情報は、「学習者の作文に対し、日本語を母語とする教師・母語としない教師がそれぞれどのような反応を返すか」を見るために収集したものであったが、すべての添削情報は教師が原稿用紙に赤ペンで書き込んだものをスキャナで取り込んだものであり、結果として極めて使いにくいものであった。

添削情報を研究に活用できるようにするためには、何らかの形で添削情報も「検索可能な形で」電子化する必要があった。そのことはまた、大量の添削情報を研究者間で共有し、作文評価・添削方法論に関する共同研究を進めていくためにも不可欠の条件であると考えられた。

そこでわれわれは 2002 年、「XMLによる作文添削情報表示システム ver.1」を開発し、これによって収集した添削情報の電子化を進めるとともに、実際の作文指導や教員指導の場面においてこのシステムを活用する試みをおこなってきた。その過程において、当初のタグの仕様にはさまざまな問題点があることに気づき、それを改善するための改変を続けてきた。

本論においては、2002年に発表された「XMLによる作文添削情報表示システム ver.1」から、2006年3月現在の最新版である ver.3 に至るシステムの改変過程を紹介するとともに、なぜそのような改変が必要であったかについて論じる。

### 1.2. 添削情報電子化に関わる先行研究

「作文対訳DB」の先行データベースである故・中野洋氏の「日中作文コーパス」<sup>1</sup>(国立国語研究所 1998)では、添削情報は図 1のような形で、テキストファイルとして電子化されていた。

---

<sup>1</sup> 平成 10 年度文部省(当時)科学研究費(創成的基礎研究)「国際社会における日本語についての総合的研究」研究班第 3 班:「表記・表現に関する実験的研究」(研究分担者:中野洋)において作成。

ひところ前、日本に名高[く, い, d]、国立国語研究所の学者の中野洋先生が上海に来られて、うちの大学で[講座, 講演, c]を[作って差しあげました, なさいました, d]。「中国流行歌の変化—日中流行歌の対照研究—」というテーマをめぐって、先生は流行歌の三つの特■を取り出して[私たちに, h]説明してくださいました。

<添削コード一覧>

「a」表記, 「b」文体, 「c」語彙, 「d」文法, 「e」敬語, 「f」文化、習慣, 「g」母語の干渉, 「h」表現（間違いではないが、日本語らしくないもの）

図 1 「日中作文コーパス」における添削情報例

「日中作文コーパス」では、誤用と判定された箇所は[ ]で囲まれ、修正案が示されるとともに、その誤用がどのような種類のものであるかが<添削コード>として示されている。

しかし「日中作文コーパス」の方式には、いくつかの問題点がある。最大の問題点は、ある誤用をどの「添削コード」に帰属させるかについて、明確な基準を立てることが困難であることである。また、データがすべてテキストファイルとして保存され、データ入力の際特別なツールを必要としないということはデータ共有上極めて大きな利点であったが、極めて読みにくく、入力には慣れが必要になるものと思われた。さらに、オリジナルの文言と修正後の文言、それに添削コードが電子的にはすべて同じように表現されるため、例えば「修正後の文言に対して検索をかけたい」といったようなニーズに答えることは困難であった。

こうした問題点を解決するものとして、われわれは、当時普及初期であったマークアップランゲージXML<sup>2</sup>に着目した。

添削情報をマークアップランゲージによって表示する試みは、すでに脇田他(1999)<sup>3</sup>などによっておこなわれていた。脇田らが開発したネットワーク型添削支援システムCoCoA<sup>4</sup>では、SGML<sup>5</sup>に基づいて開発されたネットワーク添削用マークアップランゲージCCML(Communicative Correction Markup Language)により、教師が学習者作文に対し付与した添削記号をテキスト内にタグとして埋め込むことが可能となっている。タグの入力は、教師側のタグ入力支援ツール(CoCoA-Editor)によって簡便におこなうことができる。そして学習者側の表示ツール(CoCoA-Viewer)では、タグつき文章を解釈し、従来の赤字添削と同様のものを画面上に表示することができるようになっている。

脇田らのシステムは、電子メールを用いた作文添削が円滑におこなわれるように、という目的のために作成されたものであった。このため CoCoA-Editor, CoCoA-Viewer とともに

<sup>2</sup> eXtensible Markup Language (詳細情報はW3Cを参照。http://www.w3.org/XML/)

<sup>3</sup> http://oriko.cup.com/PDF/9901\_JSISE.pdf

<sup>4</sup> 1998年、株式会社エイペックスから「コレットver.1」という名称で発売された後、1999年に「E-Correct」に名称変更された。現在はWeb版E-Correctも利用できる(詳細情報はhttp://www.apex.jp/index.html)

<sup>5</sup> Standard Generalized Markup Language (ISO8879 / JIS X 4151)

メール送受信機能を備えている。また Viewer では添削前(学習者の書いた原文)・添削後(原文に教師による取り消し線, 修正候補などが付け加えられたもの)・清書後(添削内容を反映させて原文を書き換えて表示したもの)を切り替えて表示できるなど, 電子メール上での添削を支援するツールとして極めて多様な機能を持つものであった。

われわれは, 脇田らのシステムを参考にしながらも, 「添削情報を研究者間で共有し, 作文教育・添削方法論に関する研究を進めていくにはどうすればよいか」ということを考え, 独自システムの開発を進めることとした。

### 1.3. XML による添削情報電子化

添削コードを独自に決め, オリジナルのテキストに付加するという点では, XML 文書は「日中作文コーパス」と変わらない。しかし, XML では, 外部から表示を制御できるスタイルシート(XSLT<sup>6</sup>)というファイルを利用することができる。修正前後が混在している文章であっても, スタイルシートに表示方法を定義することによって, 修正前の文言, 修正後の文言を明示的に区別して扱うことが可能となる。

表示制御が比較的自由に可能であるため, ブラウザ<sup>7</sup>上に「赤ペンによる添削」と非常に近い形で添削情報を表示させることもできる。さらに, そうした美しい表示を可能にすることで, データとして添削情報を活用するだけでなく, 教育現場において現実の「添削支援ツール」として使用することも可能になると考えた。

その反面, テキスト入力, 「日中作文コーパス」のように, 簡単な記号の入力ではなく, XML タグという, 統一規格にのっとった方法で入力しなければならない。そのため直接入力することは非常に困難と思われたため, テキストエディタ<sup>8</sup>のマクロ機能によって入力支援をおこなうこととした。

---

<sup>6</sup> XML Stylesheet Language Transformations

<sup>7</sup> WWW用のブラウザを用いることが想定されている。

<sup>8</sup> 本研究では「秀丸エディタ」(<http://hide.maruo.co.jp/>)を用いている。

## 2. これまでの「XMLによる作文添削情報表示システム」

### 2.1. バージョン1

#### 2.1.1. 初期の目的

添削ツールの開発においてわれわれが企図したことは、教育現場での添削支援ツールとして利用されることを目指しつつも、本質的には添削情報をデータとして大量に蓄積・共有し、添削情報を用いた研究のために用いていくことであった。可能であれば、単に添削記号をタグとして埋め込んでいくというだけでなく、添削研究に役立つような情報をテキスト中に埋め込んでいきたいと考えた。

通常われわれが添削をおこなう際に考えることとして、

- ・「この箇所は明らかにおかしいから、必ず～と直してほしい」と考えて直す場合
- ・「間違いとはいえないが、～と直すとさらによくなる」と考えて直す場合
- ・「～と直すのがいいように思えるが、自信が持てない」と迷いつつ修正する場合

などがある。このような「添削態度」とでもいうべきものをXMLのタグとして区別しておけば、例えば「誤用か正用かの判断に迷う部分」だけを抽出することができ、また同一の誤用箇所に対して、複数の添削者がどのような態度で添削を施しているかを比較対照することもできるようになると考えた。

#### 2.1.2. バージョン1.0のタグの仕様

そこで、2002年に発表した「XMLによる作文添削情報表示システム ver.1」(宇佐美2002)では、以下のようなタグセットを用意することとした。

タグセットは大きく4つのグループからなる。

- A. 明らかな誤用箇所に対するタグ
- B. 明らかな誤用とはいえない箇所に対するタグ
- C. 段落・字下げに対するタグ
- D. 注釈等に対するタグ

それぞれについて、タグの仕様を示す。

#### A. 明らかな誤用箇所に対するタグ

- del(削除) 書式：`<del>削除する文字列</del>`
- put タグ(挿入) 書式：`<put value="挿入する文字列"/>`
- rep タグ(置換) 書式：`<rep value="置換後の文字列">置換前の文字列</rep>`
- moveタグ(移動)<sup>9</sup>

書式：`<movefrom id="IDNo.">移動させたい文字列</movefrom>`  
`<moveto id="IDNo."/>`

#### B. 明らかな誤用とはいえない箇所に対するタグ

- better (誤用とはいえないが、より適切な表現があることを指摘)  
書式：`<better value="より良い表現">元々の表現</better>`
- doubt (執筆者の意図が明らかでなく、暫定的訂正・コメントに使用)  
書式：`<doubt value="添削者のコメント">意図不明表現</doubt>`

#### C. 段落・字下げに対するタグ

- cp (執筆者が入れた改行位置の指定<sup>10</sup>)  
書式：`<cp/>`
- newp (改行すべき場所を指定)  
書式：`<newp/>`
- runon (追い込み：執筆者が入れた改行を無効にする<sup>11</sup>)  
書式：`<runon/> <cp/>`
- sperr (字下げエラー：段落頭で字下げをしていないとき、または字下げをしすぎているときに使用)  
書式：`<sperr value="執筆者が実際に空けているマス数12"/>`

#### D. 注釈等に対するタグ

- remark (添削者から執筆者へのコメント・メッセージ等)  
書式：`<remark value="添削者コメント">コメントの対象となる表現</remark>`
- noteタグ(入力者の注<sup>13</sup>)  
書式：`<note value="入力者注">注の対象となる部分</note>`
- review タグ(文章全体に対する総評)  
書式：`<review>添削者の総評</review>`

---

<sup>9</sup> `<movefrom id="IDNo.">`と`</movefrom>`で挟まれた文字列が、`<moveto id="IDNo."/>`タグのある位置に移動するよう指定されていることになる。IDNo.の部分には任意の数字(1つの作文内で重複しないように設定)が入る。

<sup>10</sup> 添削者が新たに改行位置を指定したり、執筆者が入れた改行を無効にしたりするため、執筆者が本来入れた改行の位置をタグの形で残しておく必要がある。cpタグは添削者が挿入するものではないため、XML化する際に、ヘッダ・フッタとともに自動挿入される必要がある。

<sup>11</sup> runonタグはcpタグの前に挿入することになっている。

<sup>12</sup> 1字の字下げをすべきところでしていない場合は"0"を、2文字下げている場合は"2"を属性として入力。

<sup>13</sup> 手書きの添削情報を入力する際、うまく電子化できない現象があった場合、このタグの中にその現象を言語化して説明する。例えば、漢字の間違いを訂正しているような場合、`<note value="「降」という漢字の書き方の間違いを訂正している">降</note/>`のような書き方にする。

### 2.1.3. バージョン 1.0 タグによる XML 文書例

また、これらのタグを簡便に入力するために、テキストエディタ「秀丸」のマクロを利用したタグ入力支援ツールも作成した。以下に、タグ付きXML文書の一例を図 2に示す。

```
<sperr value="2"/>人<better value="々">達</better>はよく「*<doubt value="?">自分のこのむ</doubt>」と言っています。自分の生活は自分のです。それもいいです<put value="。"/>けれ<rep value="ど">で</rep>もたばこをす<rep value="う">わる</rep>と健康*だけ<rep value="が">は</rep>弱くな<rep value="るわけではありません">りません</rep>。お金もたくさん<better value="使います">用いられま</better>す</better>。それとともに次の世代も悪い<better value="影響">こと</better>を受<del>け</del>ます。私<doubt value="?">が</doubt>感じるのは">の感じによって</doubt>次の世代にいつもいい物を教える<del>の</del>方がいい<better value="ということです">です</better>。その人々は<better value="年をとって">年上になって</better>自分<put value="の"/>両親やほかの人達を例として<remark value="「どんだん」の方が一般的">どしどし</remark>追いかけます。<cp/>
```

図 2 添削 XML バージョン 1.0 による例

こうしたタグ付き XML 文書を、ブラウザ上で手書き添削と同じように表示するためのスタイルシート(correction.xsl)も作成した。

上掲のXML文書を、スタイルシートによって表示させると図 3のようになる<sup>14</sup>。

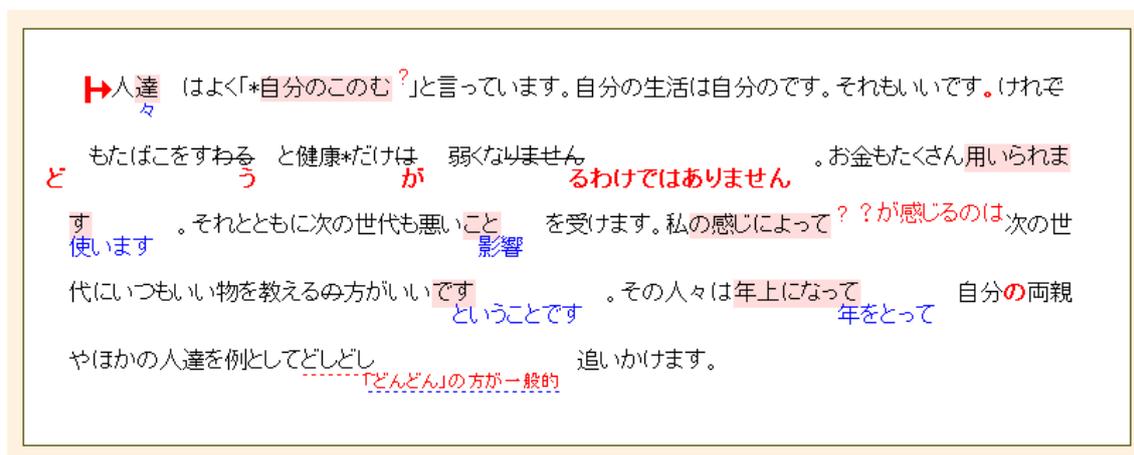


図 3 スタイルシートによる XML 表示例

図 3 は白黒での表示であるが、ブラウザ上では、「明らかな誤用」(del, put, rep タグ使用箇所)に対して示された修正案は赤字によって、明らかな誤用とはいえない箇所に対する

<sup>14</sup> ただしこのような表示を得るためには、スタイルシートcorrection.xslのほかに、画像ファイル(newp.png, note.png, runon.png, sperr.png)が必要となる(例えば図 3 冒頭の、「字下げ」を意味する矢印マークの画像ファイルなど)。

修正案は青字によって示されることになっている。

## 2.2. バージョン 2

### 2.2.1. 仕様の変更

バージョン 1 による、添削 XML の公開後、添削タグの仕様についての問題点を整理して、仕様を改訂することとなった。

手書き添削を電子化することを意識してタグが設定されていたために、直接コンピュータを用いた電子添削をしようとするといくつかの問題点が判明した。こうした経緯からバージョンの改訂作業に入った。

改訂は 2005 年 3 月に行われた。最大の変更は以下の 2 点である。

#### (1) タグの整理

- a. 削除(del)・挿入(put)・移動(movefrom/moveto)を置換(rep)に統合
- b. 置換(rep)タグを、修正(correct)と提案(better)の 2 段階でとらえる

#### (2) タグ挿入時にコメント属性を付与する

添削データの収集を、手書きではなく、電子添削によっておこなうことを考えたとき、手書きの添削の雰囲気を残したままのタグではなく、電子化としてより効率化するために、添削タグの簡素化が計画された。

(1)については、添削作業において修正作業にかかわるものは、文字の置換作業が基本であるとした。削除は空文字への置換であり、挿入は空文字からの置換と等価である。移動も単なる削除後同内容の挿入とみなすことで、すべての修正関係のタグを一つに統一することになった。

移動を廃止することで煩雑になることも予想されたが、1～2文字程度の短い単位での移動は、表現全体の置換として扱われる可能性があり、また、節以上の単位での移動のような文の論理構造にかかわる移動などは制限されると予想し、影響は少ないと考えた。

このように、置換タグが修正に関する全てのタグを担う結果、もともと「より良い表現への置き換え」という意味にしか用いられなかった better にも、挿入や削除の機能が与えられることとなった。本来 better についても挿入や削除が「より良い表現」として示される可能性があったので、この点についてはバージョン 1 の段階で仕様にはいつているべきだったといえるだろう。こうして、correct と better は 2 段階の同一タグとして整理されたといえる。

(2)は、作文添削において、作文の電子添削の効果を考える際に、修正を施した箇所に対する説明を添削者に求めることで、遠隔添削のような添削者と学習者とが話し合えないような状況でも、なるべく添削者側の情報を多く提示することを可能にすることが目的である。

「秀丸エディタ」のマクロへの実装においては、修正タグを入力する際、同時にコメントの入力を促すウィンドウを出すようにした。入力が必要なければそのまま無視してよいが、常に促すことによって細かくコメントがなされることを期待した。

また、コメントが増加することによって、実際の添削画面出力においては、コメントによって画面表示が乱れないように、コメントがある箇所についてのみ示し、そこをクリックすることによって表示されるようにした。

## 2.2.2. バージョン 2.0 の仕様

こうして、変更されたタグバージョン 2.0 の仕様を以下に示す。改行関係のタグは基本的には変更なかった。

### A. 修正関係のタグ

- 存続   • correct (rep からの名称変更)
- better
- unclear (doubt からの名称変更)
- 廃止   • put,del → correct へ統合
- movefrom/moveto → correct へ統合

統合後の修正関係のタグの使用例を示す。以下の correct は better でも同様である。

- 削除(旧 del)  
書式：<correct value="">削除する文字列</correct>
- 挿入(旧 put)  
書式：<correct value="挿入する文字列"/>
- 置換(旧 rep)  
書式：<correct value="置換後の文字列">置換前の文字列</correct>
- 移動(旧 move; movefrom/moveto)  
削除と挿入にて対応
- unclear(旧 doubt)  
書式：<unclear comment="添削者のコメント">意図不明表現</unclear>

### B. 改行関係のタグ

- 存続   • cp
- newparagraph (newp より名称変更)
- runon

### C. 注釈関係のタグ

**存続** ・ review(文章全体に対する総評)

**廃止** ・ sperr → 電子添削化するため廃止。(原稿用紙の書式に関するため)  
・ remark → 廃止。タグ内の comment 属性で対応  
・ note → 電子添削化するため廃止

### 2.2.3. バージョン 2.0 による XML 文書の例

こうして、図 2 と同じ作文例をタグバージョン 2.0 にてあらわしたものを図 4 に示す。

```
人<better value="々">達</better>はよく「*<unclear comment="?">自分のこの  
む</unclear>」と言っています。自分の生活は自分のです。それもいいです<correct  
value="。"/>けれ<correct value="ど">で</correct>もたばこをす<correct  
value="う">わる</correct>と健康*だけ<correct value="が">は</correct>弱くな  
<correct value="るわけではありません">りません</correct>。お金もたくさん  
<better value="使います">用いられます</better>。それとともに次の世代も悪い  
<better value="影響">こと</better>を受けます。私<unclear comment="?">が感じる  
のは">の感じによって</unclear>次の世代にいつもいい物を教える<correct value="の">  
</correct>方がいい<better value="ということです">です</better>。その人々は  
<better value="年をとって">年上になって</better>自分<correct value="の"/>両親  
やほかの人達を例として<unclear comment="「どんだん」の方が一般的">どしどし  
</remark>追いかけます。<cp/>
```

図 4・添削XMLバージョン 2.0 による例<sup>15</sup>

タグの機能が統合化されたことにより、repやput, del といった表現がcorrect となっ  
てしまったため、機能が分かりにくくなった。図 2 におけるバージョン 1.0 の例に比べて  
可読性が低下していることが分かる。

<sup>15</sup> 囲み部のunclearタグは、バージョン 1.0 でremarkタグの部分であった(本文図 2)。バージョン 2 の  
タグセットでは、修正を求めない単なるコメントを示すことはできない。コメントはタグに付随する属性  
としたため、ここではunclearとした。執筆者の意図が明瞭ではないので、betterになる可能性もある。

## 2.3. バージョン 2.1 への改訂と評定情報

### 2.3.1. 電子添削への以降の問題

タグが大幅に整理されたバージョン 2.0 は、電子添削によるデータ収集をめざして 2005 年 3 月以降テストされた<sup>16</sup>。

しかし、電子添削のテストの結果、タグが多くなることによって、XML ファイルが分かりにくくなり、入力も困難になることが分かった。テキストエディタのマクロによってタグ入力が支援されているとはいえ、XML の仕組みに習熟しない添削者にとって、タグ入力はかなりの苦勞を伴う作業であるようだった。

ウェブサイトにおける HTML 文書を、テキストエディタで入力することがわずらわしいといわれるように、元来テキストエディタによるタグの入力には問題があるともいえる。さらに、バージョン 2.0 においては、タグが整理されたために逆に紛らわしさが増したことや、タグにコメント属性が追加されたことでタグの長さが長くなりすぎるということもあったと思われる。

そのため、バージョン 2.0 から添削データの収集を、紙面上での手書き添削から、コンピュータ上での XML 文書による電子添削へと切り替える計画をたてていたが、テキストエディタによる添削は困難と判断した。そのため、2005 年 8 月における添削データの収集では、手書きによる添削を依頼することとした。依頼の際には「添削マニュアル」を作成し、バージョン 2.0 のタグ仕様で容易に電子化ができるような形での添削をお願いした。

添削データ収集作業と平行してタグ仕様の再検討もおこなったところ、「移動」という操作を「削除」＋「挿入」の組み合わせとして表現すると、削除箇所と挿入箇所の間に対応関係がある、ということが分かりにくくなる、という問題点が判明した。そこでバージョン 2.1 では、correct/better タグの中に ID 属性を用いることで、移動機能を復活させた（詳細は p.57 の脚注 9 を参照）。

### 2.3.2. 評定タグの導入

評定タグとは、作文中にどの程度修正の必要がある誤用か、という「誤りの重篤性」に関する 3 段階の評定<sup>17</sup>情報に関するタグである。

添削とは別に評定情報を収集した理由は、添削者が「誤りに気づいてはいたものの、教育効果に配慮したり、あるいは修正候補が思いつかなかつたり、という理由で修正をしなかった」という箇所があるのではないかと考えたからである。当初この評定情報は、添削情報とは別の派生データベースとして公開する予定であったが、同一評定者が同じ作文に対して付与した添削情報と評定情報とはやはり同じファイル上で統一的に示されたほうが

<sup>16</sup> 2004 年度国立国語研究所日本語教育短期研修(第 5 回)「作文添削の電子化・共有と、それを用いた応用研究の可能性」(2005 年 3 月 27 日)において、バージョン 2.0 の公開と、改訂版秀丸マクロを用いたタグ入力のモニターテストが行われた。このほか、2005 年度前半にも非公式のテストを実施した。

<sup>17</sup> 添削作業から一定時間が経過した後、添削者に再度同じ作文を渡し、誤りと思われた箇所に下線を引いてもらい、それぞれの箇所について「誤りの重篤性」の度合いを 3 段階で評価してもらった。

よい、という判断に基づき、添削データベースの中に評価情報も含めることとした。

この思想はバージョン 3.0 へ引き継がれるため後述することとして、ここでは詳述しない。

### 2.3.3. まとめ

以上、バージョン 2.1 への変更点は、

- (1) 移動用タグの復活
- (2) 評価タグの導入

の 2 点である。バージョン 2.1 において追加された部分について以下に示す。

**追加** ・ id 属性の追加

移動に用いる(correct/better どちらも)

(例)

移動元 <correct value="" id="ID No.">移動箇所</correct>

移動先 <correct id="ID No."/>

**追加** ・ rating (評価)

<rating value="重篤度" s="開始位置" e="終了位置">該当箇所</rating>

重篤度： 1～3 の 3 段階で示す

開始位置・終了位置： タグが挟んだ範囲のどの部分に対する評価かを示す<sup>18</sup>

---

<sup>18</sup> 添削とは別途おこなう調査であるため、添削情報と組み合わせる際には、評価タグと添削タグの範囲がずれることがある。こうした状態を回避するために、評価タグは添削タグの外側に置くようにして、囲んだ範囲の何文字目から何文字目までに対する評価かを指定できるようにした。

### 3. XMLによる作文添削情報表示システム バージョン3

#### 3.1. 「添削」という行為自体の問い直し

##### 3.1.1. 許容度の判断基準

バージョン 2.1 までの修正タグでは、「明らかにおかしいから直せ」という意味のタグ (correct)と、「誤りとはいえないがより適切な表現があるから直したほうがよい」という意味のタグ (better)とが区別されてきた。つまり、学習者が産出した「逸脱表現」を、添削者はどの程度「許容できるか」が問われていたわけである。

しかし、現実に学習者作文に見られる誤用や、教師の添削行動を分析していくと、correct と better の境目は決してひとつの要因によって決まるものではなく、複数の変数が絡み合っていることが分かってくる。

その変数のひとつとして、「意味の分からなさ」というものが挙げられる。読んでいて意味が取れない箇所というのは、文章として最低の目的を果たしていないわけであるから、通常これは「許容できない」箇所となるだろう。しかし場合によっては、「明らかに規範から外れており、不自然ではあるのだが、しかし意味は明解に理解できる」という場合もある。例えば、

私は学校に勉強します。

という文があったとする。この文では、助詞「で」を使用すべきところ、誤って「に」を使ってしまっているもので、極めて初歩的な文法ミスであるということが出来る。しかし普通に日本語を理解できる読み手であれば、頭の中でたやすく「に」を「で」に置き換えて理解することが可能だろう。この誤用によってコミュニケーション上の支障が生じることはほとんどないといってよい。

こういう場合、「に」の許容度をどのように判定するか、ということについては、かなり大きな個人差が出るものと考えられる。「文章では書き手の意図が伝わるのがもっとも大切である」と考える添削者であれば、これは十分許容できる範囲の誤用ということになるだろうし、「初級で学んでいるはずの文法項目が習得できていない」ということを重く見る添削者であれば、必ず修正されるべき重篤な誤用、と判定することになるかもしれない。

もちろん、添削態度にはさまざまなものがあってよい。しかしもし、添削者が「意味は分かるけれど重篤な誤用だ」と判断したのか、「誤りではあるがコミュニケーション上問題はないから軽微な誤用だ」と判断したのかが学習者にはっきり示されれば、それは学習者にとってさらに有効な情報であるといえようし、また添削者本人にとっても、「いま自分はこの『逸脱表現』に対し、どういう基準によって重篤・軽微の判断を下しているのか」に対し意識的になることは、非常に意義あることといえるであろう。

### 3.1.2. 非修正箇所への注目

また添削者は、許容できない箇所すべてに対し修正を施すわけではない。石橋(2002)は、添削者が作文添削をおこなう際のプロトコルを分析することにより、「添削者は、学習者の逸脱表現に気づき、留意<sup>19</sup>しても、その逸脱を『学習者の日本語習熟度では許される範囲のものである』と判断したり、その逸脱に対し適切な対案が思い浮かばなかったりした場合には、逸脱箇所を非修正のまま残すことがある」と述べている。

通常の添削では、このように「問題がありながら非修正箇所とした箇所」と、「問題がない箇所」との間には、表面上まったく区別はなされないことになる。「学習者の日本語習熟度では許される」と判断して修正しないのはよいとしても、「適切な対案が思い浮かばないために直さない」ということは、教育上問題となるかもしれない。対案は示せないまでも、「この部分は～という理由で許容できない」ということを学習者に伝える手段は用意されてしかるべきであろう。

さらに、許容できない箇所に対する修正案が思い浮かんだとしても、執筆者の執筆意図が必ずしも明確に把握できないために、その修正案が適切なものであるかどうか確信が持てない、ということもあるだろう<sup>20</sup>。そのような場合、執筆者と面談できる立場にいる添削者であれば、執筆者から直接意図を聞き出すことによってより適切な修正をおこなうことが可能であるが、しかし遠隔地にいる学習者とインターネットなどを通じて指導をおこなっている教師や、学習者と直接面談する時間が取れない教師にとっては、「十分確信は持てないが、そのことを明記した上でとりあえず修正案を示す」ということも必要であることが予想される。

---

<sup>19</sup> ここで「留意する」とは、学習者の「逸脱表現」に対し、「それを修正すべきものと認める」ということを意味している。

<sup>20</sup> バージョン2ではunclearタグで示していた事項である。

### 3.2. バージョン3.0における添削タグの入力順序

そこでわれわれは、具体的には以下のようなことが可能となるように、添削情報表示システムを改訂することとした。

- 学習者の文章の中で「逸脱表現」と認定した箇所について、「意味が分からない」という理由での逸脱なのか、それ以外の要因による逸脱なのかを明示できるようにする。
- 仮に修正案が思い浮かばなくても、「逸脱」と認定した箇所は明示できるようにする。
- 修正案に確信が持てない場合も、「確信が持てない」ことを明記した上でとりあえずの修正案が示せるようにする。

こうした複雑な判断を添削者に課す上で、われわれは、添削者が入力支援ツールを用いて添削タグを挿入していく際、以下のような「一定の手順」を設定することにした。

- (1) まず、文章中の「逸脱箇所」を範囲指定し、その箇所に対する「許容度」（どの程度許してよいか）を3段階で評定する。すなわち：「必ず修正すべき」を3とし、「逸脱してはいるが、必ずしも修正しなくてもよい」を1、その中間を2とする（この数値は、修正タグ内の属性値のひとつとなる）。
- (2) その箇所に対し、対案が提示できればそれを入力し、できなければ入力しないままにする。対案が提示できない場合、範囲指定された箇所に対しては「許容度情報」のみが属性値として表現される。
- (3) 対案を提示できた場合は、その対案に対する「確信度」を2段階で指定する。すなわち：確信をもって対案を示している場合は2を、確信はないが暫定的に対案を示している、という場合は1を入力する。対案が提示できた場合、範囲指定された箇所には修正タグが付与され、(1)で指定した「許容度情報」が属性値として表現されるとともに、対案の「確信度情報」もまた属性値として表現されることになる。

この過程をフローチャートの形で表すと以下のようなになる。

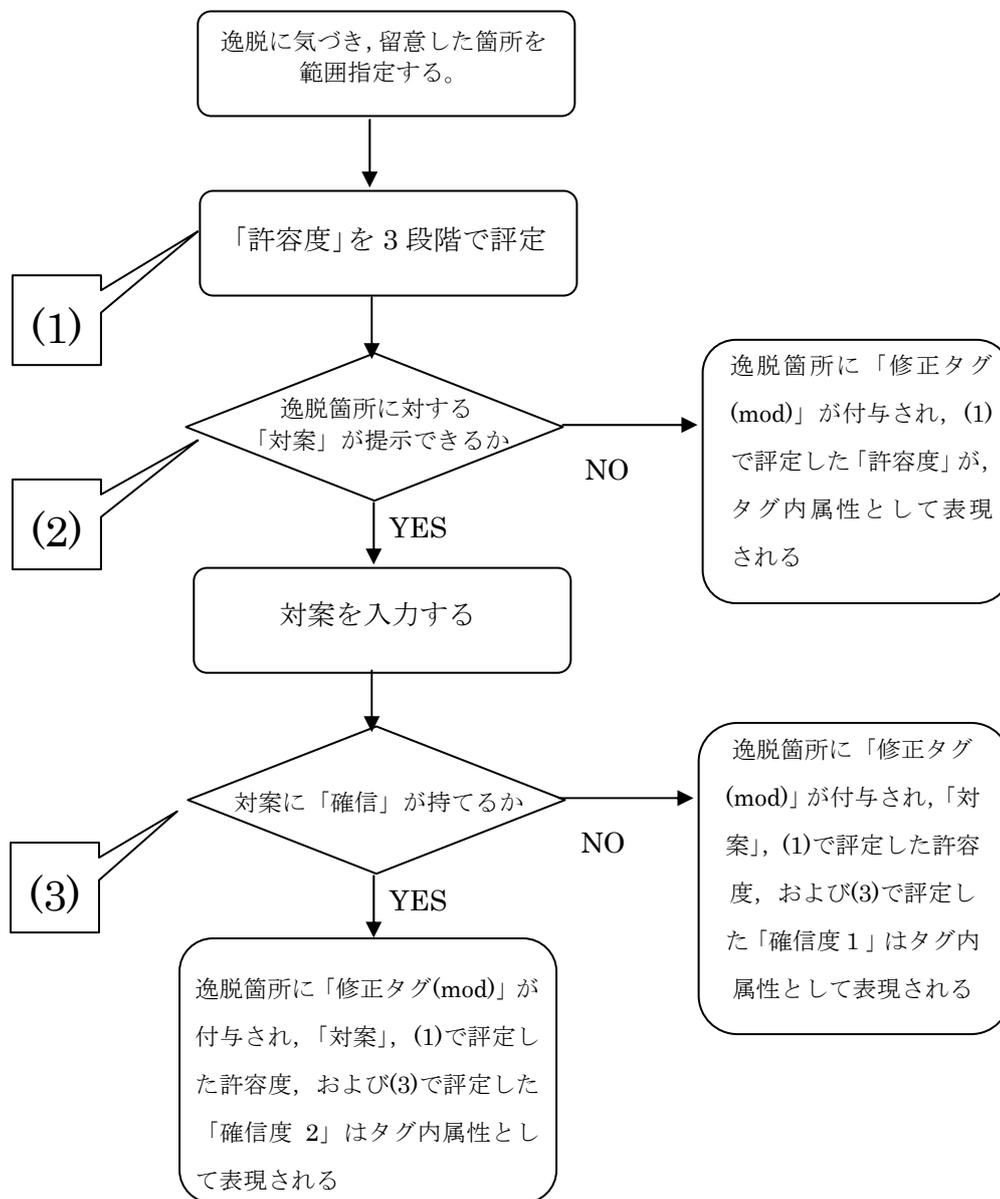


図 5 バージョン 3.0 における添削タグ入力のフローチャート

上記(1)~(3)の各ステップについて解説する。

### (1) 「許容度」

添削者はまず、全体的な「許容度」の判定を求められることになる。この段階において

は、「分かりにくさ」「不自然さ」などに関する分析的な判断はおこなわず、総合的判断としての「許せる度合い」が評定される。

## (2) 「分かりにくさ」

「意味の推察が可能であるか」、つまり「分かりにくさ」の判定が求められることとなる。意味の推察が不可能なほど分かりにくい箇所であれば、当然対案提示は不可能となる。

一方で、「意味内容推察」はできるが「対案提示」は不可能、という場合もありうる。「いたいことはよく分かるのだが、それを日本語でどのように表現していいかまったく思いつかない」、という場合である。

つまり、「対案提示不可」という場合には、「意味内容が分からないからできない」という場合と、「分かっているができない」という場合とがあることになるのだが、いずれの場合も教師として「対応不可能」という点では同じであり、学習者から見た場合、この両者を区別することにはあまり意味はないものと考えられる。そこで「対案提示」が可能か不可能か、という観点での判定だけを求めることとした。

## (3) 「確信度」

対案を提示できた場合には、その対案にどの程度の確信を持っているのか、ということをも2段階で評定してもらったこととした。

手書きによる実際の添削情報を詳細に見ていくと、「執筆者の真意が十分に分からないため、対案に十分な自信が持てないのだが、それでもとりあえず対案は示しておく」という態度での添削が決して少なくないことが分かる。例えば、クエスチョンマークつきで対案を示したり、対案を示したあとで「～ということですか?」という「執筆者に対するメッセージ」を示したり、というような形での添削である。もちろん、「このようなあやふやな態度での修正はすべきでない」という考え方もありえよう。しかし実際にこういう添削行動が見られる以上、電子添削においてもそれに対応できるような準備は必要であろうし、また執筆者と直接対面できない状況での添削においては、「確信は持てないがとりあえず対案を示しておく」ということの必要性も決して否定はできない。ただしその際、対案が「確信が持てない状態で提示されたものだ」ということは、学習者に対し明確に示されるべきであろう。このような考えに基づき、「対案の確信度」を添削タグの属性値として表現することとした。

次に、こうした手順によって入力されるさまざまな添削タグの仕様について、詳細を説明する。

### 3.3. バージョン 3.0 の仕様

#### 3.3.1. 属性仕様

バージョン 3.0 では、添削過程よりむしろ思考過程の整理に重点がおかれたため、これまで以上に修正関連のタグは整理され、共通属性が体系化された。

修正関連のタグは実質的に 1 つとなり、属性値によってその添削機能を示す形へと変化した。そのため、タグ自体の重要性が低下し、むしろ属性の解説を中心に行うこととする。

修正関連のタグがとりうる属性値は以下の 5 種類である。

- a 属性(Acceptability) 当該箇所に対する許容の度合い
- s 属性(Sureness) 添削者が、自ら提示した修正案に対し持っている確信の度合い
- r 属性(Replace) 提示した置き換え文字列
- c 属性(Comment) 添削者によるコメント
- id 属性(Identification Data) 移動やコメント時に用いる参照用記号

#### (1) a 属性

【属性名】 acceptability, 許容度

【対象タグ】 mod, newp, runon

【機能】 許容度の評定

【値】 数値(3段階: 1,2,3)

a 属性は「acceptability(許容)」, すなわち、文章の「許せる度合い」に関する評定をあらわす。添削者にとって、該当箇所の「許せる度合い」を 3 段階で評価する。最も許容できない状態が 3 で、ある程度許容できる状態が 1 となる。

従来タグとして使用されてきた **correct** と **better** も、許容度という点以外での違いはなく、修正タグとしての動作は、挿入・削除・置換・移動とすべて同じである。そのため、タグの区別ではなく、属性による違いで示すことが可能であると思われる。

段階については、**correct**, **better** の 2 段階ではなく、迷った場合も考慮して 3 段階とした。旧タグからの変換に際しては、**correct** が 3, **better** が 1 とした。

#### (2) s 属性

【属性名】 sureness, 確信度

【対象タグ】 mod

【機能】

【値】 数値(2段階: 1,2)

s 属性は、「sureness(確信)」, すなわち、提示した対案への確信の度合いをあらわす。確信度は 2 段階とし、確信の有無のみで判断する。確信がある場合が 2 で、ない場合が 1 である。そもそも対案が提示できない場合には、この属性自体が存在しないことになる。

修正候補が明確な場合(バージョン 2 での **correct**, **better** タグ)には、確信度については許容度にかかわらず 2(確信あり)となる。

問題となるのは、バージョン 1 の **doubt** タグや、バージョン 2 の **unclear** タグのコメントとして示されていた対案である。意図が不明瞭な箇所について、添削者が文脈などから推定して修正案を提示することもある。しかしそうした添削者にとって確信のもてない修正案も、コメントに含めており、参考程度の扱いであった。確信がもてない理由は複雑で、学習者の作文側に問題がある場合もあれば、添削者が修正案の提示に自信がもてない場合もあり、通常どちらか一方に原因を求めることは難しい。添削者側も明確にどちらが原因か判断できるとは限らない。

しかし、たとえ修正案が不確実であったとしても、その提示は、置換・挿入・削除という、従来の **correct** や **better** タグと同じ働きをしていると思われる。そのため「対案提示の際の確信」という観点から、従来の **correct/better** と同等に扱うことにした。中間段階を設けず 2 段階としたのは、不明瞭な状態をさらに個人の中で分類することは困難であると想像したからだが、実際のテストでの状況次第では段階が追加されることもありうる。

### (3) r 属性 — replace

【名称】 **replace** (置換)

【機能】 修正の提示

【値】 修正候補の文字列

【対象タグ】 **mod**

従来の **correct**, **better** タグにおける **value** 属性からの名称変更である。置換後の内容であるため、**replace** とした。この属性が空である場合、自動的に削除とみなされるが、もし修正候補が思いつかないまま入力を忘れてしまった場合に削除と区別がつかないことになる。教育効果を考えて、あえて対案を示さないというような場合には、削除と誤解されないようにしなければならない。

そのため、入力ツールで実装する場合には、置換の場合には、置換後の内容が入力されるまでは、空ではなく何かダミー記号を与えておく必要があると思われる。

### (4) c 属性 comment

【名称】 **comment** (コメント)

【機能】 添削者註

【値】 文字列

【対象タグ】 **mod**, **newp**, **runon**, **com**

コメント属性には、添削者のコメントが入る。添削タグや、コメント専用のタグにおいても用いられる。修正関連のタグには全てつけられるほか、コメント専用のタグももうけた。これは特定の箇所に対する感想・メッセージ等に用いられる。そのためプラス評価の

コメントもここに属する。プラスの評価については、作文添削の効果を考える上で重要な要素になると思われるため、改良の余地があるだろう。

#### (5) id 属性 ID, identification data

【名称】 ID

【機能】 参照場所の指定

【値】 文字列

【対象タグ】 mod

mod タグでは、移動時に用いる。移動は、バージョン 2.1 と同様に、ID 付の削除・挿入となる。挿入時には r 属性はなしで、移動元と同じ ID 属性を示しておけば自動的に移動元でタグに挟まれた内容を挿入することができる。

移動については、長い作文の場合には、文章の構成にともなって、複数の文や、段落といった単位での移動も必要になることがある。これは文中での添削とは別のレベルの問題となるため、このような移動にどのように対応するかは今後の課題である。

### 3.3.2. タグ仕様

以上の属性値をとるタグについて解説する。

#### A. 修正関連タグ

##### mod

【名称】 modification, 改良

【属性】 a 属性, s 属性, r 属性, c 属性, id 属性

【使用法】

mod は、改行関係以外のすべての箇所に関する何らかの修正をおこなう場合に使用する。バージョン 2 までの correct, better, unclear の全てが mod タグに統合される。従来の correct で、

```
<correct value="置換内容" comment="コメント">該当箇所</correct>
```

と表示されていたものは、

```
<mod a="3" s="2" r="置換内容" c="コメント">該当箇所</mod>
```

へと変換される。

バージョン 2.0 における修正関連タグを mod タグで表現すると、

```
correct : <mod a="3" s="2" r="置換" c="コメント">
better  : <mod a="1" s="2" r="置換" c="コメント">
unclear : <mod a="1" s="1" r="修正候補の部分" c="修正候補以外の部分">
unclear : <mod c="コメント">
```

となる。

## B. 改行関連タグ

### cp

【名称】 Change Paragraph, オリジナルの改行

【属性】 なし

【使用法】 <cp/>

XML では、原文テキストに多くのタグが挿入されるため、みかけは大きく変更される。タグの構造を明確にするために改行を用いてしまうと、もともと存在した改行記号との区別がなくなってしまう。そのため、表示上の改行は、別に cp として定義されている。

### newp

【名称】 New Paragraph, 改行挿入

【属性】 a 属性, c 属性

【使用法】 <newp a="許容度(1-3)" c="コメント"/>

### runon

【名称】 Run On, 追い込み

【属性】 a 属性, c 属性

【使用法】 <runon a="許容度(1-3)" c="コメント"><cp/></runon>

newp と runon は、改行タグを挿入・削除するという点では、通常の文字の添削行為と同様であり、mod タグで扱うことも可能である。しかし、r 属性(置換)内にタグを用いなければならないことや、対案の提示という点では、対案内容が「改行」に限定されるため、s 属性(確信度)の値は 2 に固定されてしまう。そのため現状では、従来のタグを存続することにした。

ただし runon については、cp タグを消去することを明確にするため、cp タグの前に置くのではなく、cp タグを runon タグで挟むことにした。このため、runon については、mod タグで表現可能な状態にある。この点は今後の課題である。

## C. コメント関連タグ

### **com**

【名称】 comment, コメント

【属性】 c 属性

【使用法】 <com c="コメント"/>

<com c="コメント">該当箇所</com>

コメント専用のタグである。文全体の総評を述べる場合と、文中の注のような形で述べる場合とがある。階層的には、総評タグは文全体を囲んだコメントタグと考えることができるが、現段階では別のタグとした。

コメントの場合には、必ずしも誤りであることが前提ではないが、自然さの属性をつける。プラスのコメントの場合も存在するため、独自の属性が必要になる可能性もある。

### 3.3.3. バージョン 3.0 による XML 文書例

以上が、バージョン 3.0 のタグセットである。図 2～図 4 で示したのと同じ添削情報を、バージョン 3.0 の XML 文書にした例を示す。

```
人
<mod a="1" s="2" r="々">達</mod>
はよく「*
<mod c="?">自分のこのむ</mod>
」>と言っています。自分の生活は自分のです。それもいいです
<mod a="3" s="2" r="。"/>
けれ
<mod a="3" s="2" r="ど">で</mod>
もたばこをす
<mod a="3" s="2" r="う">わる</mod>
と健康*だけ
<mod a="3" s="2" r="が">は</mod>
弱くな
<mod a="3" s="2" r="るわけではありません">りません</mod>
。お金もたくさん
<mod a="1" s="2" r="使います">用いられます</mod>
。それとともに次の世代も悪い
<mod a="1" s="2" r="影響">こと</mod>を受けます。私
<mod a="1" s="1" r="が感じるのは">の感じによって</mod>
次の世代にいつもいい物を教える
<mod a="3" s="2" r="">の</mod>
方がいい
<mod a="1" s="2" r="ということです">です</mod>
。その人々は
<mod a="1" s="2" r="年をとって">年上になって</mod>
自分
<mod a="3" s="2" r="の"/>
両親やほかの人達を例として
<com c="「どんだん」の方が一般的">どしどし</com>
追いかけます。
<cp/>
```

図 6 バージョン 3.0 による XML 文書の例

もはやXML文書から原文や添削文を読み取ることは不可能に近い。そのため、むしろXML文書としての分かりやすさを優先して、タグごとに改行を挿入した。スタイルファイル(correction.xml)により、図 3と同じ表示は得られるが、入力方法については検討が必要となる。

### 3.4. ソフトウェアへの実装

バージョン 3.0 の改訂により、XML ファイルは原文を読むものではなくなった。また、属性の増加や、入力順序の規則ができたため、入力方法も複雑化した。このため、電子添削はもとより、手書き添削の電子化においてもテキストエディタ(秀丸エディタ+入力マクロ)を用いた入力は困難になってきたと思われる。

このため、入力のための専用ツールが必要となる。実装方法としては、CoCoA(E-Correct)のような、WYSIWYG方式<sup>21</sup>が望まれる。しかし、専用のソフトウェアは、利用者に配慮したインターフェースの設計や、使用するコンピュータの種類やバージョンに考慮したプログラミング、また、継続的な保守のコストなどの問題点がある。

また、添削システムがどのような場面で利用されるのか、なども考慮しなければならない。

以上はバージョン 3.0 のタグセットを評価する上で急務の課題である。2006年5月頃までに Windows 上で動作する添削入力用プログラムの試作版を作成し、タグセットの有用性のテストや、プログラムの利用上・作成上の問題点の整理をおこなうことで、添削システム全体の改良を続けていく予定である。

---

<sup>21</sup> 編集画面に表示されるものとほとんど同じ表示が、最終結果としても得られるということ。通常、テキスト表現で、グラフィカルな表現を生成することは困難である。そのため、入力時にグラフィカルな表現を可能にすることは、使用者に対して優しい作りになる反面、フォーマットなどの表示制御情報を隠すようにプログラムを作成する必要がある。

#### 4. まとめ

バージョン 1.0 から現在の 3.0 までのタグの変遷を、以下に示す。

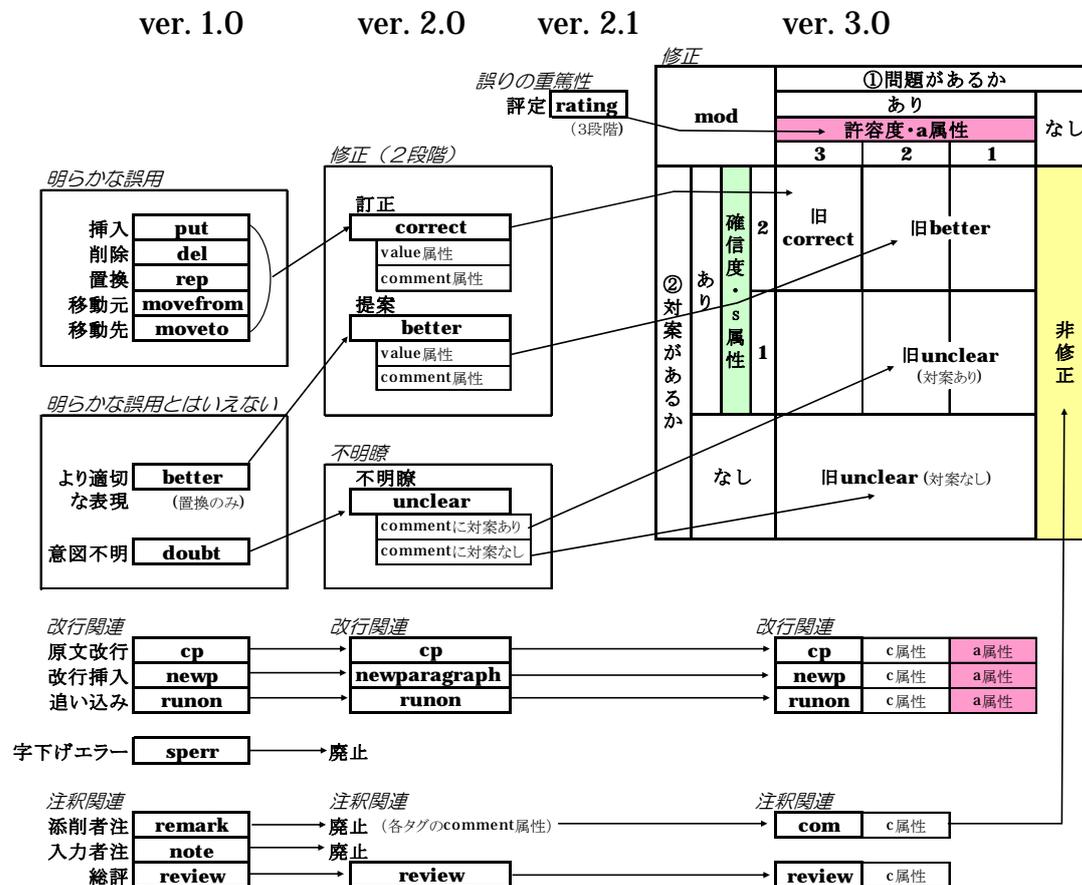


図 7 バージョン 1.0~3.0 の添削 XML タグの変遷

タグセットの変遷を時間を追って見ていくと、「逸脱表現」を大きく二つに切り分ける際の境界が少しずつずれてきていることが分かる。バージョン 1 では、`put`, `del`, `rep` といった「明らかな誤用」を表すタグ群と、`better`, `doubt` のように「それ以外」のタグ群との間に大きな境界線があった。しかし、「明らかな誤用」とは何かを決める際にはさまざまな要因が複雑に関わっていることが予想され、この境界は必ずしも明瞭なものではなかった。

そこでバージョン 2 では、「確信のある対案提示」ができる場合(`correct`, `better`<sup>22</sup>)と、そうでない場合 (`unclear`)との間に境界を移すことになった。添削者本人にとって「確信ある対案提示」ができるかどうかは、明らかな誤用かどうか、という基準よりは明確なものであると思われるからであった。

<sup>22</sup> `better` タグは、「誤りとはいえなが直したほうがよりよい」ということを示すタグであって、添削者は提示された対案に対しては確信を持っていることになる。「確信のない対案」を `better` としているわけではないことに注意。

しかしながらバージョン 2 においては、unclear は「対案があるが確信が持てない場合」と「対案がまったくない場合」の両方をカバーすることになっていた。対案が出せるか出せないか、という境界は添削者側にとっても学習者側にとってもかなり重要なものであり、これらを区別できないことはやや大きな問題であると思われた。

このためバージョン 3 では、「対案がまったくない」のか、「確信の有無にかかわらず、曲がりなりにも対案が出せる」のか、というところに大きな境界線が移動することになった。さらに、対案がある場合はその対案に確信があるのかどうか、ということが区別され、加えて、そもそも元の逸脱箇所をどの程度許せるのか、という度合いが 3 段階で区別されることになった。

こうしたタグセットの変遷は、「誤用・逸脱表現」というものをどうとらえるか、ということの考察が徐々に深化してきたことを示している。バージョン 3 にいたっては、「どの程度許容できるか」「対案が出せるか」また、「対案を出せる場合、その対案に確信が持てるか」という複数の判断軸を立て、添削者にはそれぞれについて適切に判断をしてもらうような「手順」までが指定されることになっている。その手順を追うことによって添削者は、「なんとなくおかしい」ということでなく、「どういうところが、なぜおかしいのか」ということを明確に意識しながら添削を進めていくことができるようになり、かつ「誤用・逸脱表現」に対しどのような判断を下したかということが、記録として明示的に残せるようになった。そのことは、従来の添削方式ではまったく期待できないことであった。

バージョン 2 までのタグシステムは、「手書きでもできることをコンピュータ上で再現できるようにしたに過ぎない」という見方もできるかもしれない。それがバージョン 3 に至って初めて、コンピュータ添削ならではの利点を、教育・研究の両面にわたって提示することができたものとわれわれは考えている。

## 参考文献

- 石橋 玲子(2002)「日本語学習者の産出作文に対する教師の修正及び非修正行動」『言語文化と日本語教育』23, 1-12, お茶の水女子大学日本言語文化学会
- 宇佐美 洋(2002)「XMLによる作文添削情報表示システムと、その応用」『日本語教育学会春季大会予稿集』, 231-232.
- 国立国語研究所(1998)『日中作文コーパス』,「国際社会における日本語についての総合的研究」研究班第3班:「表記・表現に関する実験的研究」文部省科学研究費 創成的基礎研究(研究分担者:中野洋)
- 脇田 里子・緒方 広明・矢野 米雄(1999)「作文教育のためのネットワーク型添削支援システム CoCoA の実践と評価」『教育システム情報学会誌』15(4), 270-275.

## 謝辞

このシステムは、国立国語研究所 日本語教育部門の研究補佐員・アルバイトの多大なる協力を得て開発され、発展してきました。特に井上義勝氏、中山智哉氏、傳智之氏の3氏に対し、この場で篤くお礼を申し上げます。