

共同翻訳作業システムの構築について

0448057 辻 奈津子

目次	
1 序論	1
1 はじめに	1
2 利用対象者	1
3 利用価値	1
4 類似サイトについて	2
5 制作を進めるための環境と技術	4
6 初期の作業計画	5
7 作業計画の変更	6
2 本論	7
1 サイト構成	7
2 新規登録システム	8
3 文献一覧・目次システム	11
4 詳細表示システム	12
5 編集システム	14
6 更新順一覧システム	17
7 削除システム	18
3 結論	19
1 アンケート結果から得られた客観的評価	19
2 改善点	21
3 最後に	24

1 序論

(1) はじめに

大谷大学文学部人文情報学科の福田ゼミでは、「人の役に立つ Web サイトを作る」という基本テーマを掲げている。そこで私は、ゼミの基本テーマに基づき「共同翻訳作業システムの構築について」という個人テーマを設定し、卒業制作及び卒業論文の執筆を進めることに決めた。

(2) 利用対象者

私が作成するシステムは多言語の文献を取り扱う研究者や意見の交換の場を持ちたいと思う研究者、所持していない文献の閲覧をしたい研究者を対象とする。

(3) 利用価値

研究者は多くの文献を利用し研究を進める。特に多言語の研究を行なう者は、日本語以外の文献も多く利用するはずである。私は研究者が研究を進める際に翻訳をまとめるサイトがあると便利ではないかと考えた。またシステムを構築しサイトにすることで、利用したい文献を所持できなかった研究者にも文献の資料として提供できるのではないかと考えた。

しかし、一人の研究者が翻訳を行い、それを表示させただけでは信憑性に欠け、独りよがりな翻訳が出来上がる可能性が出てくる。私はその状況を避け、確実性を持たせるために wiki のような参加者が正しい情報を提供し合えるサイト構築ができれば良いと考えた。wiki の要素を取り入れることにより、利用価値を高めることができ、卒論制作として進めることに意義が生じるのではないかと考えた。

(4) 類似サイトについて

(i) 文献翻訳サイト・自動翻訳サイト

私が構築するシステムと類似する機能を持つサイトとして、アーネストウェイ⁽¹⁾やエキサイト翻訳⁽²⁾などが挙げられる。

先に挙げたアーネストウェイは翻訳会社のサイトである。翻訳の依頼を受け納期までに翻訳を行ない、依頼人に翻訳原稿を渡すという請け負い作業を行なっている。翻訳会社のサイトの利用は、自分で翻訳しない分労力を使わず研究が進められるという利点がある。しかし、その代償としてコストが掛かるという難点が生じる。また完成した翻訳原稿が手元に届くまで時間が掛かるという不便もある。

次に挙げたエキサイト翻訳のような自動翻訳サイトも類似するサイトとして挙げられる。自動翻訳サイトは私が構築するシステムと同様自分で翻訳行為を行なうが、翻訳自体はシステムが自動で行なうようになっている。しかし、自動翻訳サイトの翻訳は文章として成り立っていないことが多い。なぜならシステムが単語単位で翻訳を行い、それをそのまま結果として表示させるからである。自動翻訳サイトの利用は、短文や単純な文法の文章時に役に立つが文献のような内容が難しく、なおかつ膨大な量の翻訳資料に対しては向いていない。

私が構築するシステムでは、自分自身で翻訳作成を行なうが、研究者が必要としている箇所から翻訳を始める事ができる。また、すぐに内容を確認できるシステムを構築しようと考えた。もちろん自分自身で翻訳作成を行なうだけなのでコストもかからない。

(ii) wikipedia

wikipedia⁽³⁾はオープンコンテンツの百科事典であり、賛同する人々で記事を編集・新規作成したりできるサイトである。wikipediaHP に

よると他の類似プロジェクトと比較した場合、次のような特徴があると記されている。

- ・ 百科事典の作成に目標を限定していること。この点で、例えば議論や情報交換を行う場である掲示板や USENET などと異なっている。
- ・ ウィキを利用したプロジェクトであること。他の多くのウィキがそうであるように、多くの利用者が簡単に執筆、編集できる仕組みになっている。つまり、使いやすいサイトなのである。ウィキペディアでは参加者の資格制限などを行っていないため、年齢、職業、国籍などの点で多様な執筆者が参加する傾向にある。

上記のように wikipedia は、wiki を採用しているサイトであり、私が構築しようとする参加型サイトが展開されている。参加者が知識を持ち寄り正しい情報を作り上げていくのである。wikipedia の欠点は、参加型だから現れる嘘の情報の流出だと考えていた。しかし、良識ある参加者により、嘘の情報は即時に訂正・削除されると wikipedia のガイドラインから知った。多くの利用者があるからこそ正しい情報の百科事典が完成されるのである。私は、wikipedia への不信感を消し、wikipedia のように多くの参加者に信頼され、その参加者の協力によって正確な翻訳を完成させられるサイトを制作したいと考えを改めさせられた。

しかし、wikipedia は百科事典であり、文献の全文が記載されているということはない。私が構築しようとしているシステムでは文献を扱うので、百科事典よりも1つの項目に対する情報が多くなる。さらに述べると、wikipedia のように多くの方に閲覧してもらえるサイトではなく、多くの方の役に立つ共同作業サイトの構築を目指すため wikipedia とは利用価値が異なっているといえる。

また、wiki には特有の記入規則があり、規則を理解していないと表示の際に不都合が生じてしまう。wikipedia 内では記入規則の公開を行なっているが、その記入規則が表示されているページへのリンクが明確ではなく、すぐに必要とする情報のページに到達できない構造になっている。wikipedia のオープンコンテンツは利点であり卒論制作でぜひ取り入れたいと思うが、分かりやすく利用しやすいシステムという点を重点に考えると、システムの利用価値を上げるための考察をさらに行なっていきたいを考える。

上記で挙げた類似サイトを自分自身の目で確認することにより、見習うべき点や翻訳作業から表示までの一連作業の簡素化・迅速化を追求した無料システムの構築の必要性を見出し、取り組み意義を見出せたように感じた。

(5) 制作を進めるための環境と技術

制作において私は CD-ROM で起動できる Knoppix を採用した。記述は、Web アプリケーションを開発するためのプログラミング言語の 1 つである PHP を利用し、さらにリレーションデータベース管理システムの一つである MySQL でデータベース化を図ることにした。MySQL は PHP から取得したデータを蓄積したり、保持したデータの検索、削除を可能にするシステムである。また、PHP 側にはデータベースにアクセスするための関数が用意されているため制作の際に便利であると判断した。

また、ホームページの作成・PHP の埋め込みのために HTML の技術を必要とした。さらに CSS を利用してデザインを変更し、参加意欲を向上させるサイトに仕上げることにした。

(6) 初期の作業計画

システム構築の制作を始める前に、作業をスムーズに進めるための作業計画を立てた。作業計画は次の通りである。

1. 迷いを削減し効率よく作業を進めるため、PHP・MySQLの知識を得るための勉強を行なう。その後、必要な機能やその機能を効率良く生かすための配置をツリー構造で作成する。(図1参照) 必要な機能には、新規登録・編集・削除・一覧・文献目次・検索・編集時に必要なパスワードを挙げた。さらに、新規登録の際にはPDFファイルを読み込み、文献の原文が表示できるようにしようと考えた。
2. ツリー構造を元に制作を進める。Knoppixを起動しPHPでプログラムを作り上げる。その作業と同時に、データベースを作り必要であればテーブル同士をリレーションさせ、PHPで組み立てたプログラムをより良いものになるよう考察していく。
3. Internet Explorer または Firefox で制作したシステムが Web サイトとして表示させられるか、プログラムが正常に機能しているかを確認する。福田ゼミ3回生の協力を得て、構築したサイトについてアンケート調査を行なう。アンケート調査を行なう前にはHTML文法チェック⁽⁴⁾も行なう。表示ができ、プログラムが正常に機能していても、正しい記述でなければ勉強不足であり、卒業制作として未完成であると考えたためである。私は文法チェックで各ページ90点以上を取ることを目指し作成を進めた。
4. 最後にアンケート調査を元にシステムの改善を行い、改善後正常に機能するか最終確認をし、卒業制作を完成とする。

上記のような流れを意識し、卒業制作を始めることにした。

(7) 作業計画の変更

PHP・MySQLの勉強した後、私は改めて構築するシステムの内容について考え直した。さらに制作を進め、システムについて改めて考察した結果、以下の内容に変更を加えることに決めた。

1. 初期の計画時、研究テーマは「チベット文献共同翻訳作業システムの構築について」であった。私は、制作に移る前にゼミ監督である福田先生にツリー構造を確認してもらい、意見をいただいた。その結果、現在の研究テーマ「共同翻訳作業システムの構築について」に変更した。図1のツリー構造を見ていただいたら分かるが、基本構造はブログと類似したものになっている。チベット文献特有のシステムを必要とし、構築する予定がなかったのである。故にチベット文献に限定せず、日本語以外の多言語の翻訳作業が可能なシステム構築に卒業制作内容を変更した。
2. 新規登録時に表示させるPDFファイルは利用しないことにした。対象者を主に研究者としているため、翻訳しようとする文献を所持している方が多いのではないのかと考えた。手元に文献があれば文献を見ながら翻訳行為を行なうと判断し、PDFファイルを登録する手間と、PDFファイルを開くまでの時間を削減させ、制作作業の単純化と時間短縮を図った。
3. 検索の機能を排除した。検索は、文献名検索のみの制作を考えていた。しかし、文献目次ページを作る際、トップページと文献目次の間に文献選択ページを経由させる必要があると判明した。文献選択は文献名を50音順に配列したものである。文献50音一覧と元から制作を考えていた更新順一覧があれば検索は必要性に欠けるのではと考察した。その結果、検索機能を排除することにした。

4. パスワード機能を排除した。wikipedia ではパスワードを取得して記事の編集を行なうようにしている。パスワードは、編集者を明確にし、荒らしを防止するという点から存在価値が生じる。しかし私は、研究テーマの中でも「参加しやすい・利用しやすい」という点を重視していた。そのため、より多くのユーザーにシステムを利用してもらい、翻訳作業の担い手として参加してもらうためにパスワード機能を排除した。自分自身の経験でパスワードや ID 番号の取得と聞くだけでそのサイトの利用をやめようと判断することが多々あったからでもある。私と同じ思考を持つ方は少なくないと判断し、パスワードの利用は中止することにした。

上記の変更項目を考慮し、構成を直したツリー構造が図 2 である。この図 2 のツリー構造を元に今回の卒業制作を進め、実際構築したシステムを完成させた。

2 本論

(1) サイト構成

今回作成した Web ページは、全 16 ページで構成されている。各ページには関連するページへ移動するためのボタンやリンクを用意した。サイト構成はシンプルにしようと心掛け、WWW ディレクトリ内に 16 個の PHP ファイルと 1 個の CSS ファイルのみ用意した。(図 3 参照)

PHP ファイルの内訳を示すと、トップページに関するファイル 1 つ、新規登録システムに関するファイル 5 つ、文献一覧・目次システムに関するファイル 2 つ、詳細表示システムに関するファイル 1 つ、編集システムに関するファイル 3 つ、更新順一覧システムに関するファイル 1 つ、削除システムに関するファイル 3 つとなっている。

トップページは全てのページにリンクを繋げている。どのシステム作業が終わっても、またトップページからスタート出来るように配慮した結果、全てのページにリンクを繋げることにした。

CSS は、基本レイアウトの統一を図るためと文字の強調のために用意した。レイアウトに関して述べると、落ち着いた目に優しいサイトになるように心掛けた。初め、背景色は白にしようと考えていた。文字を認識しやすい背景色は、白だと考えたためである。しかし、白は光を反射させる性質を持っているため長時間コンピュータの画面を見ていると目に負担をかけてしまう。そのため、文字が表示される中央部分のみ配色を白にした。そして、中央部分を取り囲む部分の背景色を温かみのある淡い色を採用し、目の疲れを和らげるように考慮した。さらに白と暖色との背景の境界線と各ページのバナー部分に気持ちを落ち着かせる効果を与える青を取り入れ、基本レイアウトを作成した。

(2) 新規登録システム

新規登録システムへのサイト移動はトップページ(資料1参照)のみに限定した。文献名登録ページ(資料2参照)が新規登録の入り口であり、その入り口を1つに限定することでサイトの単純化を図った。資料2を見ていただいたらわかるが、文献名登録ページには1つのリンクと1つの入力フォームを用意した。この2つのうちどちらかの手段を選択し、新規登録を行なっていく流れになる。ひと括りに新規登録と述べているが、登録作業は文献名登録と文献内容登録の2段階の作業を用意した。

文献名登録は、文献の書名登録を行なうために設置し、当サイトに翻訳したい文献が登録されていない場合のみ利用する。文献名登録ペー

ジの「文献名登録をして下さい」の下にあるフォームに文献名を入力することにより MySQL の title データベースにデータが登録される。また、文献名の登録と同時に文献内容入力ページ (資料 3 参照) へ進むようにした。このページでは文献内容の入力作業を行なう。見ていただいたら分かるが、それぞれの項目にフォームを設置し登録データを直接入力、確認ボタンで登録完了とし、登録確認ページ (資料 6 参照) に進むようにした。ここで登録したデータは MySQL の honyaku データベースで管理される。

新規登録の手段は上記で挙げた文献名登録後サイト移動する方法の他、文献名一覧から文献名を選択しサイト移動する方法がある。文献名登録ページの「一覧から文献を探す」というリンクから移動した先に設置した文献選択ページ (資料 4 参照) が該当するものである。文献選択ページは、文献名データがデータベースに保持されている場合に利用し、翻訳作業の続きを行いたいというユーザーのために用意した。文献名を選択すると文献内容入力ページ (資料 5 参照) に移動する。

文献内容入力ページのアプローチ方法を 2 通りに分けたため、文献内容入力ファイルは 2 つ用意した。しかし、ページの見た目はどちらも同じものにした。理由は、同じ作業を行なっているという意識をユーザーに与えるためである。アプローチ方法は異なるが新規登録を行なうという概念に変化がないため見た目の統一を図った。またアプローチ方法を分けることで、文献名登録作業の短縮化を図り、データベース内の重複登録を避けた。章ごとの登録にしたのは、区切りを明確にするためとアプローチ方法を 2 種用意したためである。

上記でも述べたが title テーブルは MySQL を利用して作成した。(図 4 参照) このテーブルには、文献名登録ページのフォームから登録した

データが管理されていく。文献名登録ページでフォーム入力を選択した場合、フォームに入力したデータを PHP の機能である from action で文献内容入力ページに反映させる。そしてデータを受け取った後、文献名の項目に入力したデータを echo で表示させるのである。

文献選択ページに進んだ場合、title テーブルでデータ管理を行なっている ID 番号を利用し文献名を表示させる。ID 番号は重複を避けるように MySQL 内で設定している。そのため、文献選択ページで指定した文献名には重複しない ID 番号が割り当てられており、その ID 番号と共に管理されている文献名が必ず表示されるのである。指定するだけで文献名が表示されるので、ユーザーの手間を減らし、同時に入力間違いが生じる可能性を低下させることが出来るのである。

文献内容入力ページで確認ボタンを押すと、honyaku データベースにデータが登録され、同時に登録確認ページに移動する。図 4 の createhonyaku.sql を見ていただいたら分かるが、honyaku データベースだけでは文献名は表示できない。なぜなら、honyaku データベースでは文献名に値するデータを数値管理しているためである。そのため、honyaku データベースの title_id と title データベースの t_id をリレーションさせ、文献名の表示を行なった。

サイト表示する際、私は改行・空白も記述通りに反映させたいを考えた。そこで、登録確認ページでは本文と注釈に nl2br() を登用した。nl2br 機能は、後に続くカッコ内に書かれた情報を記述通りに指定する性質を持っている。print や echo の後に記述することで、全て記述通りにサイト内で表示することが出来る。

(3) 文献一覧・目次システム

トップページにある文献一覧を選択すれば文献一覧ページ(資料7参照)が開く。文献一覧は、文献名のデータをMySQLから抽出して50音順に中央表示させている。データは文献名だけを利用できれば良いのでtitleテーブルを利用した。また、各文献名にはリンクを貼り、選択した文献の目次ページにサイト移動するようにした。(資料8参照)

文献一覧ページからサイト移動する際、選択した文献の目次のみを表示することに苦戦を強いられた。目次ページで苦戦を強いられたのは、MySQLのtitleデータベースのID番号を取得し、目次ページに反映させながらサイト移動を行なうこととtitleデータベースで得たID番号の文献の目次のみを表示させることであった。

1つ目の問題は、a hrefで移動先のファイルを指定し、文献名と共にtitleデータベースで管理しているID番号を次のリンク先である目次ページに送信するように指定をした。2つ目の問題は、文献一覧ページから送られてきたtitleデータベースのID番号の情報を受け取り、受け取った情報とhonyakuデータベースの文献名に当たる数値とリレーションを組み合わせることで表示データの限定を可能にした。

文献一覧ページへのサイト移動はトップページと目次ページ、後ほど説明する削除ページの3つだけに設定した。当システムは、文献名を限定し作業を行なうページが大半を占めている。そのため、目次ページへのサイト移動を多くし、同文献の別作業を即時に行なえるようにした。ユーザーである研究者は、一度に1つの文献の翻訳しか行なわないのではないかと考察し、文献一覧ページのリンクを少なくした。

目次ページは文献一覧ページと同様、中央表示をさせ各項目にリンクを貼っている。各ページのバナーには機能に関する名前をそれぞれに与

えていたが、目次ページではバナーに文献名を表示するようにした。バナーはサイトの内容を明確に伝えるページの顔である。そのため、文献名をバナーに表示するようにし、どの文献の目次なのかを明確にした。

資料 8 を見ていただいたら分かるが、バナーに入る文献名は文献一覧ページで選んだものによって変化する。そのため、文献一覧ページで指定した ID 番号の情報を目次ページで受け取り、MySQL で管理されている ID 番号と同一であると認知させるプログラムを組んだ。このようにすることで MySQL で管理されているデータの中から文献一覧で指定した ID 番のみを取り込める。そして PHP の echo の機能を使い、指定した文献名を表示するようにした。

目次ページは文献一覧ページと異なり、目次 NO. で配列している。目次 NO. は 2 桁の数値とハイフンで成り立ち、「1 章 2 節」であれば「01-02」と新規登録作業で入力してもらう。しかし、福田ゼミの 3 回生にシステムを利用してもらった際、1 章のような 1 桁の情報を表示させる場合に「0」を入力し忘れ、目次が乱れるという事態が何人かで起こった。「0」がない場合、1 章の「1-1」の次に 11 章の「11-1」が表示され、その後に 2 章の「2-1」が表示されるという状況になる。これは配列を数値の昇順で指定しているためであり、「1」の情報を先に拾い出し、表示させてしまうためである。この状況を改善するため、1 桁の章を記述する場合には「0」を加え、2 桁の数値にして登録をしてもらうよう文献内容入力ページの説明文を改正した。

(4) 詳細表示システム

目次ページで目次 NO. と目次見出ししか無かった情報をさらに詳しく表示させるのが詳細表示ページである。(資料 9 参照) 詳細表示ペー

ジへサイト移動する際、目次ページの各リストに与えられている重複しない ID 番号を認知し、その ID 番号と共にデータベースで登録されているデータを抽出、さらに表示出来るようにした。

ID 番号は MySQL の honyaku データベースのものである。しかし、honyaku データベースだけを利用して文献名を数値管理しているので文献名の表示がうまくいかない。そのため、title データベースとリレーションを組み、文献名を文字列で表示させることを可能にした。

詳細表示システムは新規登録システムや後ほど述べる編集システムと密接に関係している。登録した内容をそのまま表示させるのが詳細表示ページの役割になるからである。そのため、サイトに表示させる各項目名は 3 つのシステム全てで統一しておく必要がある。

しかし私は、各ページでの確だと思ふ項目名をそれぞれに与え、統一させることを皆無していた。別の項目名を与えていたのは、現在の「目次 NO.」と「目次見出し」の 2 項目である。項目名を統一する前は、新規登録システムで「章 NO.」「章見出し」という名前を与えていた。新規登録システムは目次システムを作成するよりも先に構築作業を行なった。その際、後に作成する目次システムの事を考慮せず、ユーザーには文献の章番号と章見出しを登録してもらおうという意識しかなかった。さらに目次システム構築の際、新規登録システムでの項目表示のことを意識せず、honyaku データベースで管理されている情報を取り出し登録番号通りに表示させることのみを考慮し、システム構築を行なった。

システム全体の状況を考えず個々のページ構成について考察した結果このようなミスが発生させてしまった。このミスは自分では発見できず、アンケートをお願いした福田ゼミ 3 回生に指摘されて気が付いた。サイト構築の際には個々のページ構成を考慮すると共にシステム全体の

結びつき、全体を見通して最良のシステム構築を考察していく必要があるのだと実感した。

(5) 編集システム

編集ページへのサイト移動は、2パターン用意した。2パターン用意したのは、自分のミスを正す場合と他人のミスを正す場合の2つの異なる視点から最良の方法を考えた結果である。

1つは、新規登録後の内容確認ページからの移動である。mixi⁽⁵⁾など既存するWebサイトを参考にすると新規登録でデータを正式登録する前に確認ページを用意しているWebサイトがいくつかあった。しかし、PHPファイルの数を少なくし、共用できるページを最大限に生かせるページ構成を考えた結果、私は確認ページを用意しないことにした。そして、登録後に内容を確認し、誤字などがあった際即時に編集が出来るよう編集ページへのリンクを用意したのである。

もう1つのサイト移動方法は、詳細内容ページの翻訳情報の下にある[編集・削除]をクリックして移動する方法である。内容を確認した後で、違和感がある部分の訂正が即時に出来るよう考慮した結果、編集へのリンクを設けた。

翻訳システムに関するファイルは3つ制作している。簡単に述べると編集記述ファイル(資料10参照)、確認ファイル(資料11参照)、登録ファイル(資料12参照)の3つである。以下からはファイルそれぞれの説明を記述する。

(i) 編集記述ファイル

編集記述ファイルは、新規登録システムと同じフォームにした。しかし、3点新規登録システムと異なる仕様を取り入れた。1つは、記述説

明と記述例の排除である。前記でも記述したが、ユーザーが編集を行なうのは2つの場合がある。自分もしくは他人のミスを直す場合である。どちらの場合も記述形式を視覚情報として得て編集行為を行なう。つまり、1度視覚で得た情報と類似した形で記述すれば良いと脳で理解出来るという事でもある。そのため、私は記述説明と記述例は必要ないと判断し排除した。また、記述説明と記述例の表示を無くす事でページスクロールを短くし、編集フォームに到達するまでの億劫さを削減した。

2つ目の異なる点は、フォームである。編集システムのフォームには、編集するデータの内容が表示されないといけない。編集を行なうということは、元からある内容に手を加えるという事であるため、元の情報を表示させていないと意味がない。そのため、honyaku データベースで管理されている情報を取り出し、PHP の機能である echo で表示させるように設定した。また編集システムでも、文献名を表示させる必要があるので title データベースとリレーションを組ませた。今まで述べてきたシステムでは、単にデータベースで管理している情報を print や echo を利用し、表示させていた。しかし、編集システムではフォーム内に表示させる。そのため input type を利用し、text と指定することでフォームを作った。さらに value 値に echo を指定し、表示させたい変数を記述した。この変数とは、honyaku データベースで取得したデータに変数を与えたものを指している。

3つ目の異なる点は、input type の reset 機能である。資料を見ていただいたら分かるが、記述では新規登録システムと異なる所はない。しかし、Web サイトで表示した場合その機能に変化が現れる。新規登録システムでは、クリアボタンを選択すると記述した内容が全て削除されるが、編集システムでは、記述前の状況つまり honyaku データベ

スで管理されている情報に戻るのである。これは、value 値で honyaku データベースの内容を表示させるように指定したためである。クリアボタンは指定した状況に戻す機能である。そのため、新規登録システムでは白紙に、編集システムでは honyaku データベースで管理されている情報表示に戻るのである。前記でも述べたが、編集は元からある情報に手を加える機能であり、一から情報を作り上げる機能ではない。新規登録と同様に元の情報を削除するのは編集の役割とは異なると判断し、新規登録システムとは異なる使用にした。

(ii) 確認ファイル

確認ファイルは、編集記述ファイルで訂正した部分を反映させ、表示するページである。そのため編集記述ファイルの情報を POST で受け取り、確認ファイルに受け渡しを行なっている。POST で受けた情報にそれぞれ変数を与え、それを echo で表示させている。また、表示させる必要のないデータは input type の hidden を使い表示させないようにした。このように設定したのは、移動先のページによって受け渡す必要があるデータだからである。

ページの下の方には、編集した内容でよいか判断するボタンを用意した。「はい」を選択すると登録ファイルへサイト移動し、「いいえ」を選択すると編集記述ファイルに戻るようにしている。「いいえ」を選択した場合は、登録の行為を行なっていないため honyaku データベースで管理されている情報に戻り、翻訳記述ページに表示されることになる。

(iii) 登録ファイル

登録ファイルでは、実際に honyaku データベースへの更新を行なう。POST 機能を利用し、サイト移動する前のファイル、つまり確認ファイルの情報を受け取り、UPDATE SET を実行することで更新を行なう。

そして、更新が成功すれば資料 12 のように「更新しました」と Web サイト上に表示される。また、更新が失敗した場合には「更新できませんでした」と表示され、視覚から状況判断が出来るように設定した。この判断は、if 文を取り得れることで可能にした。

(6) 更新順一覧システム

トップページとリンクを繋げた更新順一覧システム (資料 13 参照) は、私が卒業制作でぜひ取り入れたいと考えたシステムの 1 つである。システム構想段階で、私は wikipedia を参考にすることにした。前記でも述べたが、wikipedia はオープンコンテンツであり、多くのユーザーの指示を得ているサイトである。多くの指示を得るということは、それに見合うシステムが構築され、ユーザーにとって利用しやすいサイトが実現されているということである。更新順一覧システムは、そんな wikipedia を参考にしたシステムの 1 つである。

システムを利用しやすいものに仕上げようと思案した際、すぐに最新情報を確認できる「更新」という wiki の機能が役に立つのではと考えた。最近ではブログなどでも取り入れられ、ユーザーにも浸透しているので、即時に使用してもらえらる機能であると判断し、卒業制作の機能の 1 つに追加した。

honyaku データベースに登録・編集された順にリスト表示させるのが更新順一覧システムである。ページの基本レイアウトは、前セクションで述べた文献・目次一覧システムと同じ仕様にした。しかし、更新順にするにあたりリスト表示させる情報を追加した。追加した情報は、honyaku データベースに登録した日付である。そして、この追加した日付情報を元にリスト表示を実現させた。また、文献一覧システムと目次

システムの情報をまとめてリスト表示することで、ユーザーが何のデータか認知しやすいように配慮した。

リストの配列は、日付の降順とした。日付を降順に設定することで、最新の情報がページの最上段の位置に表示される。このように上段から新たな情報を表示させることで、ユーザーに視覚的に分かりやすく最新情報を即時に提供できるのである。

(7) 削除システム

削除システムには、文献内容削除と文献削除の2種の削除方法を用意した。ひと括りに削除と述べても、削除したいと考える状況は十人十色だと考える。そのため、多くの状況に対応できる削除方法を配慮した結果、上記で述べた2種の削除方法を作成することにした。

資料10を参考にさせていただいたら分かるが、文献内容削除は編集記述ファイルのフォームの下に設置している削除ボタンから実行できる。文献内容削除は、honyakuデータベースで管理されている情報の削除を行なうシステムである。PHPのDELETE FROMという記述で削除の実行を行なっているが、honyakuデータベースのID番号と削除を行なう前のページ、つまり編集確認ファイルで使用したID番号が一致するように指定することで初めて削除が可能になる。このようにID番号の指定を行わなければ、ID番号を持つ全データの削除が実行されてしまう。文献内容削除は、文献の一部削除を実行する際に役に立つ。当システムでは章単位で文献を細分化しているため一部削除を可能としているのである。もし章より細かい単位での削除を行ないたい場合であれば、編集システムを利用して文章の削除を行なってもらう。

文献削除は、文献一覧システムの下のある位置にある「文献削除」という

リンクを指定すると実行できる。文献削除では、まず削除を行なう文献を選ぶ必要がある。資料 14 を確認していただいたら分かるが、文献削除選択ページのレイアウトや見た目は文献一覧ページと同様の仕様にした。しかし、見た目の変化がないままではユーザーがページ移動をしたと把握しにくいと判断し、「削除」という言葉がページ内に表示されるように配慮した。削除の実行は、削除したい文献を選択すると即時に行なわれる。同時に削除が実行されたことが証明されるページに移動するように設定した。(資料 15 参照) また、文献内容削除でも資料 15 と同じ削除が実行されたことを証明するページ(資料 16 参照)が表示される。

文献内容削除と文献削除の違いは、データベースの選択である。前記でも述べたが、文献内容削除の場合は、honyaku データベースのみの削除である。これに対して文献削除では、honyaku データベースの削除と同時に title データベースの削除を実行する。PHP の記述では、title データベースに対して DELETE FROM の実行を行なった後、if 文で削除が失敗した場合と成功した場合のパターンを用意した。そして、削除が成功した場合、honyaku データベースに対しても DELETE FROM を行なうようにした。2 つの削除を同時に行なうことで両データベースの削除を可能にし、文献全てのデータが削除されるのである。

3 結論

(1) アンケート結果から得られた客観的評価

序論でも述べたが、今回卒業制作で作成した共同翻訳作業システムを福田ゼミに所属する 3 回生 5 名に利用してもらい、アンケートの記入をお願いした。(図 5・6 参照) アンケート実施の目的は、第 3 者の意見を知るためである。どんなに自己評価を良くしていても、ユーザーにとっ

て役に立つものが仕上がっているとは限らない。つまり、主観的に考えて作成している自分自身には、客観的な視線で制作物を見ることが出来ていないということになる。これは福田ゼミのテーマから著しく遠ざかっており、卒業制作を「完成品」と呼ぶことが出来ない。そのため、制作に全く関わっていない福田ゼミの3回生に協力をお願いし、率直な意見と感想をアンケートとして記述してもらった。結果は以下の通りである。

利用しての感想、レイアウトについては全員から良い評価をいただき、さらに「本システムは『協力型翻訳作成』となっているが、参加しやすかったですでしょうか」という質問4の結果では、全員から「はい」という評価をいただいた。この質問は私の卒業論文の根本課題であったため、第3者から良い評価をもらったことに安堵した。またこの結果で当システムの利用実用度が増したと感じ、制作物に対して自信が得られた。

しかし、サイトの利用説明や新規登録の使用説明については意見が分かれ、平均評価で「なんとなく分かる」という結果になった。著しく悪い訳ではないが、万人に理解してもらえる説明文を提供できなかったことが窺える。実際、自分自身で説明文を記述する際、頭の中では整理が出来ているのに文章で表現できないということがあった。この結果は自分自身の文章力の無さが原因だと判断できる。

また、アンケートでは以下のようなシステムの改善を指摘した意見をいただいた。

- ・書き込んだ内容が間違っていた時にコメントで指摘出来るようにしたら良いと思います。
- ・新規作成のページが top にあるなら編集ページも別に top であっ

た方が分かりやすいかなと思いました。

- ・メニューの「文献一覧」と「更新順一覧」は同じような項目だと思うので、「文献一覧」内に「作成順の並べる」というリンクを貼るか、メニューの順番を変えたり分けたりした方がいいと思います。

この3つの意見は、私が考えていたシステム構築とは違う視点でサイトを見ていただいたと実感できる意見であった。上記の意見も含め、私はさらにシステムを良いものに出来ないか思案してみることにした。

(2) 改善点

卒業制作の作業を無事終わることが出来たが、作業後のアンケートについて考えを巡らせると何点かの改善すべき点があったのではないかと思案した。その結果、以下のように改善するとユーザーにとって参加しやすい、利用しやすいシステムになったのではないかと考える。

(i) 新規登録システム

まず挙げるのは、新規登録システムである。文献内容入力ページでは、ユーザーに対してどのようなアプローチをすれば利用方法を把握してもらえるか考えを巡らせた。その結果が作成したシステムの形として存在しているが、現段階ではe +の登録ページ⁽⁶⁾(図7参照)のように制作すればより利用方法を分かりやすく説明できたのではないかと考える。

ポイントとして挙げるのは、必須事項の明確化、説明文とフォームの近距離表記である。私が制作した共同翻訳作業システムの新規登録では必要がなければ注釈の記述を行なわなくてもよい。しかし、Webサイト上でその情報を公開していないためユーザーに理解してもらうことが困難であった。そのため、赤の表記でアスタリスクマークを項目名の

横に表示し、必須記述事項として明確に記すべきであったと考えた。また、新規記述フォームと密接する位置に 1 行程度で記述説明を表示させるべきであったとも考える。短文でなおかつ近距離に配置することでページ幅が短縮されスクロールバーを短くさせることが出来る。さらに明確に記述したいフォームの説明をユーザーに提供することが出来るのである。また、フォーム部分をテーブル設定に変更することで、CSS の設定も簡潔にでき、見栄えの良いものに整えられた可能性がある。

(ii) 編集システム

制作したシステムには、文献名の編集が用意されていない。これは致命的な失敗であったと考える。翻訳を行なうのは文献の内容だけではなく文献名も含めてである。つまり、文献名の翻訳間違いが発生する可能性があるということである。削除システムを使い文献名を一度排除し、新たに作成し直すことは可能であるが、その場合文献内容も削除され、再び入力し直すという 2 度手間が発生してしまう。そのため、新規内容登録時に文献名が変更できるようにフォーム表示にする事、文献名一覧の下に位置する「削除」と並べて「編集」のリンクを設定しておくべきであったと考える。

他に、編集記述ページに「クリア」ボタンを配置したが用意しなくても支障は無かったのではないかと考える。編集前に戻すという設定は、Internet Explorer や Firefox の 1 つ前のページの戻るという機能と類似しているからである。元から設定されている機能を利用し、「クリア」ボタンを排除すれば操作ミスを削減できるのではないかと考える。

もし「クリア」ボタンを維持するのであれば、位置をフォームより上部に移動させ、さらに右隅に設置するなどの配慮を加える必要がある。また、「クリア」という表記ではなく「元に戻す」という表記に変更する

べきであるとも考える。「クリア」という言葉を目にすると一般的に無に帰すという判断を下すと考えるからである。表記が「クリア」のままではユーザーは使用することを躊躇し、存在価値を低下させる結果になると考える。

編集システムでは、直接入力による編集とは別に新たな機能が必要なのではないかと思案した。このように考えたきっかけは、3回生のアンケート結果を拝見したためである。前記でも述べたがアンケートでは、コメント機能の追加を指摘された。この意見をみた時、私は wikipedia のノート機能を思い出した。ノート機能というのは wikipedia に記載されている記事情報の構成などについて話し合うための機能です。つまり、ページの間違いを編集という形ではなく、コメントで知らせるというものである。wikipedia のノート機能を参考に思案してみると、この機能を取り入れることで自分の翻訳に自信が無いというユーザーにも気軽に利用してもらえるのではないかという考えが生まれた。詳細表示システムから編集とは別にコメントというコマンドを用意し、フォームがあるページに移動、コメント登録後は詳細ページの下に内容情報が追加されていくようにすればサイト利用の効率が良くなると考えた。

(iii) 更新順一覧システム

更新順一覧システムは、日付順に配列をしている。しかし、年月日の情報までの指定しか行っていなかったため情報が不十分であったと考える。同じ日付の内に更新をした場合、更新された順に配列されない場合が発生する可能性がある。このため、年月日に加え時間の定義づけを行なう必要があったと考える。

(iv) 削除システム

現システムでは、削除する文献もしくは詳細内容を選択すると削除が即時に実行されるように設定されている。このように設定したのはページの単純化と縮小化を図った結果であったが、このままでは操作ミスをした場合の対処が皆無にされてしまう。そのため、削除を実行する前に確認ページを用意する必要があると考えた。特に文献削除で操作を誤ると膨大なデータが削除され、同時に多くのユーザーに迷惑が掛かることになる。これを回避するには編集確認ページのような視覚で内容確認するページを確保する必要がある。レイアウトや表示内容は編集確認ページと大差なく制作すれば良い。同じような仕様にすることで確認事項を行なっているという意識をユーザーに与えられるからである。

以上が改善すべき点であり、このように仕上げることにより良いシステム構築が可能になったのではないかと考える。さらに考えるならば、システムの構成上「翻訳サイト」と限定する必要があるのだろうかという点である。研究者を対象としたサイトと考えるのならば、このシステムで本の執筆も出来るのではないだろうか。例えば、共同で本を書く事も可能になり、研究者当時のコミュニケーションを取り合う場としても利用できるサイト構築が可能だったのではないかと考える。

(3) 最後に

今回私が制作した共同翻訳作業システムは、ユーザーが必要な文献の翻訳を自らの手で必要なページの内容から制作、編集、削除などができ、さらに他のユーザーが作成した翻訳に対しても編集を行なうことができ、相互して正確な文献翻訳作成が出来るというものであった。

実際に卒業作成を進めていると自分の勉強不足を痛感し、何度も制作

することを放棄してしまおうかと考えた。卒業制作物が完成しても、前記のように改善すべき点を考えれば考えるほど思いついてしまい、本当にこのシステムで良かったのだろうかと考えてしまう。しかし、今回の制作が失敗だと思ふことは無い。

私は今回の卒業制作のように一からシステムを構築、制作し、さらに利用してもらうという経験を体験したことが無かった。卒業制作は、人生経験としても自分のスキルアップとしても良い経験であったと考える。普段はあまりしない他人の視点から物を考えることを意識し、どのようにすれば役立つか十分に思案する実践が出来ることは幸福であったとも思う。学生最後に自分にも出来るという経験をすることで自分自身に自信を持たせた気がする。

システムの構築を経験し、ユーザー全てが満足するサイトを構築することは不可能であると再認識した。しかし、多くのユーザーが必要とするサイトにさらに近付けるシステムを模索し、構築していくことは可能であると理解した。万人が妥協できるラインにどれだけ近づけるかが参加しやすい・利用しやすいサイトへの近道だと私は考える。それを考えると、私が構築したシステムにはまだまだ改良の余地があり、実際にWebサイトとして運営するにはムラがありすぎると考える。

一からシステムの構想を考え、構築し、一定人数ではあったが利用してもらう。必要な技術の基礎知識も持っているか分からなかった私が、多くの問題を解決し、無事卒業制作を完成させることが出来たことには意義があると思う。また、今まで無かったサイトの構築を考察したことによって多くの方に利用してもらえる新たなサイトを作り出せる可能性も見出せたのではないだろうか。

注

- (1) アーネスト : <http://www.earnestway.com/>
- (2) エキサイト翻訳 : <http://www.excite.co.jp/world/>
- (3) wikipedia : <http://ja.wikipedia.org/wiki/>
- (4) HTML 文法チェック : <http://htmlint.itc.keio.ac.jp/htmlint/htmlintl.html>
- (5) mixi : <http://mixi.jp/>
- (6) e + お客様基本情報登録 : <http://north2.eplus.jp/sys/main.jsp>

文献表

神谷 歌寿彦

『はじめての人のためのかんたん PHP+MySQL 入門』

秀和システム, 2003