言語学習初心者に優しい 多言語入力支援システムの開発とその評価

池田佳代[†]　沼田秀穂[†]　兼子正勝[††]　町田和彦[†††]

多言語情報資源へのアクセスを，OS や対象言語の制約を持たない文字入力手段の提供により実現することを目指し，多言語に対応した入力支援システムの研究を行った．特に，利用したい言語に不慣れな言語学習途上のユーザーでも簡単に所望の言語の情報資源にアクセスできることを目指し，インターネットブラウザで入力操作を行うシステムを構築した．各言語（スクリプト）毎にキー入力文字と出力文字の対応を示す変換辞書を準備し，変換辞書を工夫することにより，すべての言語（スクリプト）に対応可能とした．初心者向けに入力を高速化する手段として，インクリメンタルサーチでの文字変換を導入し，さらに変換候補となる語彙に関する情報を表示する機能を設けた．実装したシステムについては被験者による評価実験を行った．本研究により，言語学習初心者が，パソコン環境にとらわれず文字を入力する環境が実現できた．

Development and Evaluation of a Multilanguage Text Input Support System that is Easy for Beginner Language Learners

Kayo IKEDA[†]　Hideho NUMATA[†]
Masakatsu KANEKO[††]　Kazuhiko MACHIDA[†††]

In this paper, We aimed to allow multilingual information sources to be accessed by using a text input method unrestricted by OS or target language, and we researched an input support system with multilingual support. We also aimed to enable users who are still learning a language and not familiar with vocabulary to be able to easily access information sources to find the word that they are seeking, and we propose a system which performs input operations using a web browser. For each language (script), a conversion dictionary is available which shows how the key input string and output string correspond. By devising a conversion dictionary, this system can support all languages (scripts).We perform text conversion in incremental search as a method to speed up input for users who are beginner language learners, and we have a function which displays information related to the vocabulary items that are among the conversion candidates. The results of evaluation experiments showed that we succeeded in creating an environment in which beginner language learners can input text regardless of their computer's environment.

1. はじめに

IT 化の進展と共に，世界中で多言語・多文字情報資源が蓄積され，インターネット上で発信される時代となった．コンピュータ上で扱うことのできる文字は，コンピュータ黎明期にはラテン文字が中心であり，ASCII における 7 ビット文字コードに代表されるように，少ない文字数しか扱うことができなかった．しかし，各国語対応の IT 環境の進展と，インターネットの普及により，世界中の言語に用いるあらゆる文字を統一的に扱おうとする国際標準規格：ISO/IEC10646 Universal Multiple-Octet Coded Character Set および Unicode が規格化された[1]．これによって，世界中で共通した文字コードにより，多言語・多文字情報資源を蓄積するベースが整備された．

本研究では，利用したい言語に不慣れな言語学習途上のユーザーでも簡単に Web 上にある所望の言語の情報資源に検索エンジンを用いてアクセスできるような，多言語に対応した入力支援システムを提案する．

2. 問題の所在と目的

Web 上に公開された情報資源へのアクセスは，主に Web ブラウザでの操作により行われる．Web ブラウザでの情報資源へのアクセス方法は複数あり，一つは特定の Web ページへの直接的なアクセスであり，この場合は既知の URL へのリンクをクリックすることや，URL を直接入力するなどで所望の情報資源へのアクセスが可能となる．もう一つには，Google や Yahoo!に代表されるポータルサイトと呼ばれるインターネットの入り口となる Web サイトを経由したアクセスであり，ポータルサイトでは，検索エンジンや Web ディレクトリなどのサービスを展開している．このうち Web ディレクトリは，人知によって分野別に分類した Web サイトの索引集であり，分野が階層構造になっていることから，所望の分野の階層を辿ることで Web サイトへのアクセスが可能となる．ただし，索引集に登録されていない Web サイトや，分野が特定できない Web サイトへのアクセスは不可能である．一方，不特定の Web サイトを対象とした検索エンジンの中で，Google のようなロボット型検索エンジンでは，クローラーと呼ばれるプログラムが周期的に全世界の Web コンテンツを取得し自動的に検索用のインデックスを DB に収納する．クローラーが Web コンテンツを収集する範囲や，インデックスの作り方は各検索エンジンで異なるものの，ロボット型検索エンジンに対し，さまざまな言語での検索クエリー（キーワード）を指定することでさまざまな言語の

* [†] (有)エクセリードテクノロジー
Excellead Technology Co., Ltd.
[††] 電気通信大学
University of Electro-Communications
[†††] 東京外国語大学
Tokyo University of Foreign Studies

情報資源へのアクセスが可能となってきた．

しかしながら，各検索エンジンで指定する検索クエリーの文字入力が問題となる．

文字入力方法としては，これまで，ラテン文字やインド系文字，アラビア文字などの表音文字系のスクリプトにおいては，キーボードに文字コードが直接アサインされている直接入力方式がとられている．従って，所望の文字入力を行うためには，その文字に対応したキーボード配列を規定し，それを変更する手段を OS レベルで実装している必要がある．

一方，日本語や中国語，韓国語などの多文字圏では，InputMethod（あるいは FEP：Front-End-Processor と呼ばれる）というソフトウェアが必要となる．これは，(1)キーボード上で入力した文字列を変換辞書を用いて解釈し，候補文字の一覧を画面表示し，(2)ユーザーが表示候補の中からその 1 つを選択する，という処理である[2].

現実的には，直接入力方式または InputMethod による変換入力方式のいずれも，どの言語（スクリプト）を入力できるかは，使用 OS 環境の設定に依るところが大きい．従って，新たに日本語や英語以外の言語を今までの OS 環境のまま利用しようとすると，言語（スクリプト）によっては，相当の困難がつきまとう．

このようなローカルホストの入力環境からの脱却として，Web 上の InputMethod が挙げられる．

すでに，ブラウザ上でインターネットを介して日本語入力ができる日本語 Web IME の研究[3][4]のように，海外においてローカルホストに日本語入力環境がなくても日本語入力が行えるような研究が進められている．しかしながら，これまで多言語をターゲットとした Web 上の InputMethod の研究は存在しなかった．

広大な Web 空間に多様な多言語情報資源が蓄積されていても，Web 検索によりそこにアクセスする手段が簡単には手に入らない状況にある．そこで本研究では，情報資源の対象をインターネットで Web 公開されている Unicode（UTF-8）で記述されたあらゆる言語データと置き，そのデータへアクセスするために必要な，検索クエリーを入力するための多言語 InputMethod を提案し，実装，評価を行う．

本研究では，ユーザー対象を「利用したい言語に不慣れな言語学習途上のユーザー」と置くことで，だれもが簡単に多言語入力を行える環境の提案を目指す．また，ユーザーがその言語を入力するためにローカルホストでの特別なセットアップの必要が無いことを前提としたシステム提案を行う．

3. 提案システム

本研究では，クライアント側の OS 環境に依存せず，多言語情報資源から情報検索を行うために，対象言語の制約を持たない文字入力手段の提供として，Internet Explorer などの Web ブラウザで入力操作を行うシステムを提案する．OS 環境に依存しない文字入力手段とは，言語毎でキーボード配列を切り替えたり，InputMethod を切り替えたりしないということを意味する．そのための提案として後述のように，各言語における入力文字を ASCII 領域内（キーボード上のキー配列の範囲）の文字に置き換える方式を採用する．

3.1 多言語対応

多言語対応の対象を Unicode で規格化されているすべての言語（スクリプト）とする．キーボード上のキーにすべての文字コードをアサインすることは不可能であることは言うまでもない．ここには，多文字を処理してきた日本語 InputMethod の処理工程の 1 つである「辞書との照合」というプロセス[5]を持ち込む．辞書とは，入力文字列と出力文字列を対応させる変換辞書である．

入力文字列には，世界中のどのようなパソコン環境でも間違いなく入力可能な ASCII 領域内の文字（制御コードは含まない）を利用する．従って，キーボード上のキーに入力文字列をすべてアサインすることができる．

出力文字列は，UTF-8 エンコードの文字列とし，Unicode 化されているどのような言語（Script）でも出力可能とする．各言語（スクリプト）毎に入力文字列と出力文字列の対応を示す変換辞書を準備する．

本研究で，各言語における入力文字を ASCII 領域内の文字とするという考え方は言語学における転写（transcription）を活用したものである．転写とは，言語の音声を一定の規則に基づいて文字表記することをいう．これまで，インド系文字やアラビア文字などでは，主に ASCII 領域内の文字を利用して転写表記したもの（ローマ字転写ともいう）を情報資源として利用してきた[6]．日本語のヘボン式や訓令式といったローマ字表記も一種の転写規則とみることができる．

さまざまな言語の転写規則については，規格化されたものがあるわけでなく，各言語の研究者が音声とソーティングなどの利便性の面で工夫した転写規則を各自で定めている．従って 1 言語について，何通りもの転写規則を作ることは可能となるが，本研究では，多くの言語に対応することができることを実証することが目的であるので，転写規則の詳細については議論しない．

変換辞書は，転写規則を元に，入力文字列を ASCII 領域内の文字とし，出力文字列を各言語の UTF-8 エンコードの文字列とした対応表であり，言語毎の変換辞書を工夫することにより，すべての言語（スクリプト）に本システムは対応可能となる．

3.2 語彙辞書

変換辞書に登録する入力文字と出力文字の対応は，直接入力と同じ結果を期待する場合は，入力文字：出力文字＝ 1 文字： 1 文字であるが，日本語等の InputMethod による変換入力に似た結果を期待する場合は，入力文字：出力文字＝n 文字：m 文字(n≧1, m≧1)とすることができる．

後者では語彙を変換辞書に投入することを前提としており，本研究では，次に示す

インクリメンタルサーチにより語彙の綴りをうろ覚えのユーザーに配慮した設計とした．

3.3 インクリメンタルサーチ

インクリメンタルサーチとは，ユーザーが 1 文字入力する度に候補文字を表示していくこと検索手法で，逐語検索，逐次検索とも言う．Jef Raskin(2000)は，インクリメンタルサーチにより検索がすばやく行えるだけでなく一打鍵毎にユーザーにフィードバックが返る点が優れている，と主張している[7].

インクリメンタルサーチは，これまでも GNU プロジェクトによるテキストエディタの Emacs や，日本語のインクリメンタル検索手法である「Migemo」[8]や，携帯電話や Google の検索エンジンに利用されている．

これまでの日本語の InputMethod のような変換辞書を伴う文字変換では，一連のキー入力を終了後，文字変換を指示することで初めて変換候補がリストアップされた．

インクリメンタルサーチでは，最初の 1 文字（C1）を打鍵した段階で，C1 を先頭に含む変換候補がリストアップされる．さらに続けて 1 文字（C2）を打鍵すると，C1C2 を先頭に含む変換候補がリストアップされる．

これにより，完全に入力文字列を打鍵すること無しに，所望の語彙に絞り込まれる．特に，入力しようとしている言語に不慣れなユーザーは，その文字の綴りに自信がないことが想定されるため，インクリメンタルサーチにより，一打鍵毎のユーザーへのフィードバックにより候補文字が表示されることは，正確な綴りに早く導かれるきっかけをあたえるものであると仮説設定する．

3.4 詳細情報表示（支援機能）

本研究の新規性の 1 つは詳細情報表示にある．

本研究では，ユーザー対象を「利用したい言語に不慣れな言語学習途上のユーザー」とおいている．従って，語彙のつづりに自信がなかったり，類似したつづりの単語の区別がつかなかったりすることが想定される．そこで，インクリメンタルサーチにより入力しようとしている語彙が本当に入力したい語彙であることを確認する手段として，変換候補となる語彙に関する情報を表示する詳細情報表示機能を提案する．

詳細情報は変換辞書内に，入力文字列と出力文字列とともに収納する．インクリメンタルサーチにより，出力文字列が変換候補としてリストアップ表示された際に，詳細情報を表示する．詳細情報は Web ブラウザで表示することを前提としているため，データ記述には HTML タグを利用可能とし，詳細情報の表示には多言語（多スクリプト）はもとより音声や画像の提示や他の Web ページへのリンクも行えることとする．

詳細情報を充実させることで，その言語に不慣れなユーザーが間違いなく所望の語彙入力ができ，かつ言語学習の補助としての機能を保有することを仮説設定する．

4. システム実装

4.1 システム構成

Ajax（Asynchronous JavaScript + XML）技術を用いてシステム実装を行った．

開発環境には，非同期通信処理の実現手段として Google Web Toolkit（GWT）を使用し，ミドルウェアとして Apache2.0.59，Apache Tomcat 5.5.26，PHP5.2.2，phpMyAdmin2.10.1 を利用した．

本提案システムで Ajax を用いる理由としては，本提案システムが，ユーザーにローカルホストへのプラグインソフトのインストール等の環境設定における負荷を与えないことを目指しており，Web ブラウザのみがあればよい環境を実現するためである．また，従来の Web ブラウザを使った Web アプリケーションでは，データをサーバに通知して処理結果を得るにはページ全体をロードしなおさなければならず，本提案システムのようなインクリメンタルサーチを伴う入力方法を実現することは不可能であったが，非同期通信を利用し，通信結果に応じてダイナミック HTML で動的にページの一部を書き換える Ajax の技術により，実現可能となった．

各言語（スクリプト）の入力に必要な変換辞書データは，リレーショナル DB である MySQL 5.0.82 に収納した．

4.2 システムの特徴

通常のサーバー・クライアントシステムでは，HTML フォームや URL パラメータといったデータをサーバに送信し，サーバはそのデータに基づいた処理を行い，結果を含めた HTML ページを自動作成してクライアントに返し，表示する，という処理スタイルをとる．本システムでは，Ajax の非同期通信処理を利用しており，ページ遷移を一切伴わずに，クライアント・サーバー間の通信を実現している．

また，本システムでは，HTML ページのエンコード，Java で記述するクライアント・サーバーサイドのソースコード，DB のデータのすべてを UTF-8 でエンコードすることによりあらゆる言語（スクリプト）に対応し，ブラウザ上に多言語を表示することを可能としている．入力文字列として使用される文字は，ASCII 領域内の文字である．

本システムでは，インクリメンタルサーチにより，入力された文字列を元に DB 検索を行い，変換候補をリストアップ表示する．出力文字の確定はユーザー操作により行われるか，入力文字列に対し変換候補となる出力文字列が 1 語も存在しない場合は，一文字前にさかのぼって検索を行い，変換候補が存在するところで，自動的に変換が行われる．

また，本システムは，Web 空間への情報検索を目的としているため，検索クエリーには，スペースを区切りとした複数語の入力が求められる．一方，多言語対応においては，多くの分かち書きを行う言語の変換辞書収納が課題となる．システムとして両者を両立させるために，スペース（U+0020）の扱いが問題となる．本システム実装で

は，変換辞書に登録する入力文字列にはスペースを使用することは不可とし，分かち書き言語で複数語からなる熟語等を 1 つの出力文字列として登録したい場合は，スペースを“_”（アンダーライン，LOW LINE：U+005F）とする規則を設けた．

4.3 変換辞書データ

本システムのデータベースには，変換辞書データを言語毎にテーブルとして収納した．各テーブルは，入力文字列を search_term（varchar (100)），出力文字列を term(varchar(100))，詳細辞書を info (text)として設計した．図 1 にヒンディー語でのテーブル例を示す．

search_term	term	info
aMDaa	अंडा	अंडा<...
aMtaRii	अंतड़ी	अंतड़...
aMd'aa	अंधा	अंधा<...
aMd'aapana	अंधापन	अंधाप...
Ad'eraa	अँधेरा	अँधेर...
aMtaHpura	अंतःपुर	अंतःप...
Ad'aurii	अँधौरी	अँधौर...
akaala_1	अकाल १	अकाल ...

図 1　ヒンディー語のテーブル例
Figure. 1　Example of the database table for Hindi

本システムの実装テストに使用するオリジナル辞書データは，三省堂辞書シリーズや東京外国語大学アジア・アフリカ言語文化研究所の研究者が保有する各言語の語彙辞書を元にして本システム用のオリジナル辞書データ形式に変換したデータである．

これら既存データの多くには，カタカナ発音または転写が含まれているので，出力文字列から入力文字列を生成することは比較的容易に行うことができる．

4.4 ユーザーインターフェース

ユーザーは，Web Browser 上で入力操作をおこない，文字確定後，検索サイトに文字列を検索クエリーとして投げることができる． 一連の入力操作を以下に示す．

(1) 入力言語の選択

Web Browser 上のプルダウンメニューにより，入力したい言語を選択する（図 2）．

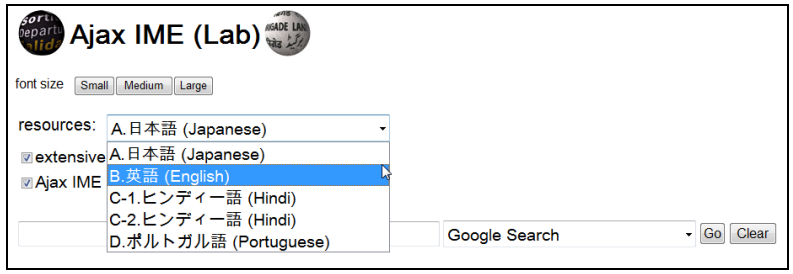

図 2　入力言語の選択画面
Figure. 2　The appearance of selecting the input language

(2) インクリメンタルサーチによる文字入力

ここでは，Devanagari を Script として利用するヒンディー語の入力例を示す．

サンプルとして登録したヒンディー語用の変換辞書には，合計 7,524 語からなる東京外国語大学アジア・アフリカ言語文化研究所のヒンディー語辞典を使用した．

ヒンディー語の「Book：本」は「किताब」であり，変換辞書内の入力文字列は「kitaaba」である．一連の入力操作とクライアント・サーバー間の通信を図 3 に示す．

まず，「kitaaba」の先頭文字「k」を入力する．変換辞書より「k」が先頭文字となる出力文字列候補がリストアップされる．変換辞書に登録されている語のうち「k」から始まる語は 555 語であり，kitaaba は 449 番目である．ここで，リストアップされる出力文字列候補の順列は，文字コードによるソーティングの昇順である．

次に，「i」を入力すると，変換辞書より「ki」が先頭文字となる出力文字列候補がリストアップされる．変換辞書に登録されている語のうち「ki」から始まる語は 44 語であり，kitaaba は 41 番目である．

次に「t」を入力すると，変換辞書より「kit」が先頭文字となる出力文字列候補がリストアップされる．変換辞書に登録されている語のうち「kit」から始まる語は 2 語のみなので，マウスで出力文字列候補をクリックするか，キー操作により出力文字列候補を選択状態にし，エンターキーを打鍵するかにより「किताब」を確定する．

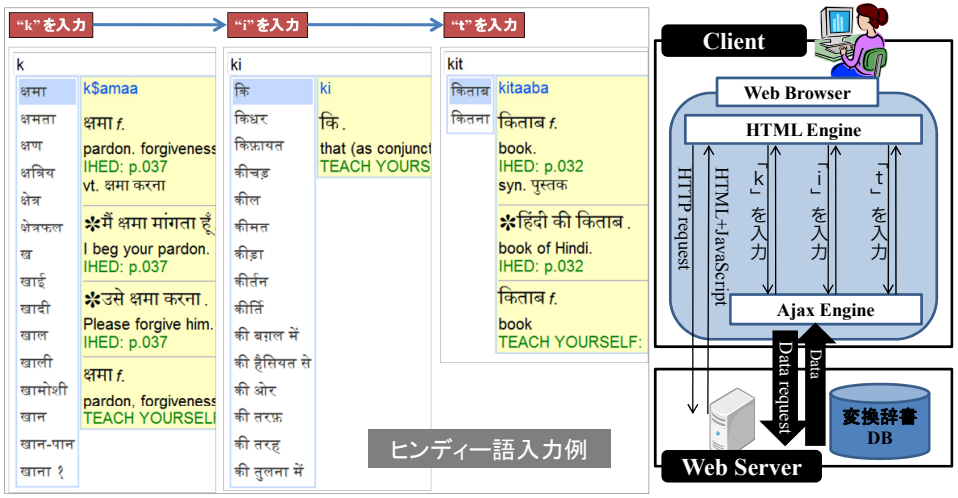

図 3 ヒンディー語入力例とクライアント・サーバー間の通信
Figure. 3 An input example using Hindi and the communication between the client and server

(3) 検索エンジンへの遷移

入力確定した文字列を最終的には，Web への検索クエリーとして利用する．本システムでは，Go ボタンをクリックすることで，検索エンジン（Google で実装）に検索クエリーを引き渡すところまでを一連の操作として構築した．

5. 評価

実装したシステムには，インド系文字 10 言語（Devanagari, Tamil, Telugu, Thai, Bengali, Gurmukhi, Kannada, Sinhala, Sanskrit, Malayalam），アラビア文字 3 言語（Persian, Uyghur, Urdu），日本語，英語，ポルトガル語の変換辞書を登録し，Web 画面上のプルダウンメニューで言語を切り替えることで，所望の言語の入力を行うことができた．

さらに実装したシステムを使って，評価実験を行った．

5.1 ユーザーによる入力実験

被験者による評価実験を行った．実験は，「3. 提案システム」での以下の仮説設定を評価することを目的とする．

① インクリメンタルサーチを用いた本システムにより，その言語に不慣れなユーザーが所望の語彙を正確に早く入力できる．

② 詳細情報により，その言語に不慣れなユーザーが正確に所望の語彙入力でき，かつ言語学習の補助としての機能を保有する．

なお，以下評価実験の中では，本システムを「AjaxIME」と呼ぶ．

(1) 方法

クライアント環境において，Windows Vista OS 上の Internet Explorer 8 を用い，言語毎に，「既存の入力方法」と「AjaxIME」により，解答用紙により提示されたあらかじめ評価用に準備した文字列を入力し，その文字列を検索クエリーとして検索エンジン（Google）で検索した際の結果として検索件数を解答用紙に記録してもらう．本番問題を始める前に，入力方式毎の操作方法の説明および練習問題を 5 問行ったのち，制限時間を 10 分として 30 問の検索と検索結果（検索件数）の記録を行う．

「既存の入力方法」と「AjaxIME」での入力実験が終了したのち，被験者に対し 2 つの方法の操作についての主観評価をヒアリングする．

また，本実験では，本番問題での被験者のキーボードによる文字入力，文字入力以外のキー操作(スペースキー, エンターキー等の打鍵), マウス操作をすべて記録する．

本実験では，結果として，各入力方式での回答数，正答数，誤答数を集計する．回答数とは，検索結果を回答できた設問の数である．正答数，誤答数は，ユーザーが記録した検索件数と，実験当日に実際に正しく文字列を入力した際の検索件数とが一致した場合を正答，一致しなかった場合を誤答として集計される設問数である．

なお，実験当日に実際に正しく文字列を入力した際の検索件数は，同じネットワーク上でも若干の誤差が生じる場合があるため，検索件数が実際に正しく文字列を入力した際の検索件数と異なる場合は，ユーザーのキー操作の記録をもとに正答か誤答かの判断を行った．

(2) 実験対象言語と既存の入力方法および AjaxIME の設定と被験者

日本語，英語，ヒンディー語，ポルトガル語を実験対象として既存の入力方法と AjaxIME の比較を行う．4 言語を対象とする理由としては，本研究の提案方式が多数の言語に対応していることを示すとともに，ユーザーの慣れが異なる言語において既存の入力方法と AjaxIME の評価を行うことで，その言語に不慣れなユーザーにとって AjaxIME が有効である仮説設定を確認することである．そこで，①使い慣れた日本語，②比較的入力が容易であり義務教育レベルである程度の知識を持つ英語，③使用する文字が特殊（Devanagari 文字）で学習経験がないと入力がほとんどできないヒンディー語，④基本はラテン文字であるが学習経験がなく綴りになじみがないポルトガル語，というそれぞれ異なる要素を持つ 4 言語での文字入力実験を行うこととした．

言語毎の「既存の入力方法」，および「AjaxIME」で利用する辞書データと実験用に準備した入力文字列は以下の通りである．入力文字列については，言語間では異なる文字列であるが，言語毎の異なる入力方式間では同じ文字列を用いる．本実験では 1 方式で 1 つの文字列を入力後，異なる方式で同じ文字列を入力するまでに，他の文字

入力を行い，かつ 10 分ほどのインターバルがある．従ってタスクに慣れるということは，短期記憶[9][10]の側面から影響はないものと考える．

① 日本語

「既存の入力方法」としては，OS 標準の Microsoft IME を用いる．

「AjaxIME」で利用する辞書データとして，三省堂大辞林データを元にした約 2 万語を使用する．三省堂大辞林データに含まれる情報は，出力文字列と詳細情報にあたる．入力文字列は，出力文字列のカタカナ読みを元にヘボン式で変換した文字列である．実験用に準備した入力文字列は，上記三省堂大辞林データから乱数発生によりランダムに抽出した 30 語である．

被験者は，H01~H06，P01~P08 までの合計 14 名である．

② 英語

「既存の入力方法」としては，キーボードによる直接入力を用いる．

「AjaxIME」で利用する変換辞書データとして，三省堂デイリー辞典のデータを元にした約 1 万 5 千語を使用する．三省堂デイリー辞典データに含まれる情報は，出力文字列と詳細情報にあたる．入力文字列は，出力文字列とほぼ同じ文字列であるが，複数語で構成される場合にはスペースを“_”（アンダーライン）に変換する．

実験用に準備した入力文字列は，上記三省堂デイリー辞典から乱数発生によりランダムに抽出した 30 語である．

被験者は，H01~H06，P01~P08 までの合計 14 名である．

③ ヒンディー語

「既存の入力方法」としては，オープンソースソフトウェアとして公開されている Virtual Keyboard v3.5.3[a]（以下 VK とする）を用いる．ヒンディー語の入力方法は，日本語 OS 環境に標準的には搭載されていない．そこで，本実験では，バーチャルキーボードとして，モニター上にキーボード配列を表示し，バーチャルキーボードをマウスクリックまたはハードウェアとしてキーボードを打鍵することで各言語の入力が可能な VK を用いた．ヒンディー語を母語として利用する場合の入力はキーボードにすべての文字が割り当てられている直接入力方式であり，VK では，ヒンディー語専用のハードウェアとしてキーボードなしにヒンディー語の入力が可能となる．

「AjaxIME」で利用する辞書データとして，東京外国語大学の所有する 7,524 語が収納されたヒンディー語辞書を使用する．ヒンディー語辞書データに含まれる情報は，出力文字列と詳細情報にあたる．入力文字列は，転写規則をもとに出力文字列から生成した．本実験で使用した転写規則は必ずしも万人に共通するものではない．したがって，「AjaxIME」ユーザーに対しては，転写規則表をあらかじめ提示しておく．

ヒンディー語については他の言語同様に詳細情報を含む辞書に加え，詳細辞書が有効であるという仮説設定を確認するため，詳細情報を含まない辞書を準備し，それぞれを「詳細辞書あり」，「詳細辞書なし」として入力実験をおこなった．

実験用に準備した入力文字列は，上記ヒンディー語辞書から乱数発生によりランダムに抽出した 30 語である．

被験者は，H01~H06 までの合計 6 名である．ヒンディー語を記述する Devanagari 文字はラテン文字とくらべて誰でもすぐにわかる文字ではないため，本実験では，ヒンディー語の学習経験のある 6 名を被験者とした．

④ ポルトガル語

「既存の入力方法」には，VK を用いる．ポルトガル語は ASCII 文字内のラテン文字とポルトガル語特有の記号で構成されるが，その記号を入力するためには，OS レベルでの入力方式切り替えが必要となるため，ヒンディー語と同じ VK を用いる．ポルトガル語を母語として利用する場合の入力はキーボードにすべての文字が割り当てられている直接入力方式であり，VK では，ポルトガル語専用のハードウェアとしてキーボードなしにポルトガル語が入力可能となる

「AjaxIME」で利用する辞書データとして，三省堂デイリー辞典のデータを元にした約 1 万 5 千語を使用する．三省堂デイリー辞典データに含まれる情報は，出力文字列と詳細情報にあたる．入力文字列は，出力文字列とほぼ同じ文字列であるが，ポルトガル語特有の記号は(例:extraordinário)，入力文字列からは削除し(例:extraordinario)，複数語で構成される場合にはスペースを“_”（アンダーライン）に変換している．

実験用に準備した入力文字列は，上記三省堂デイリー辞典から乱数発生によりランダムに抽出した 30 語である．

被験者は，ポルトガル語をまったく知らない P01~P08 までの合計 8 名である．

表 1 実験条件

Table 1 Experimental condition

	日本語		英語		ヒンディー語			ポルトガル語	
実験 No.	A1	A2	B1	B2	C1	C2	C3	D1	D2
入力方法	既存の入力方法：Microsoft	Ajax IME	既存の入力方法：直接入力	Ajax IME	既存の入力方法：VK	Ajax IME(詳細情報なし)	Ajax IME(詳細情報あり)	既存の入力方法：VK	Ajax IME
AjaxIMEで利用する変換辞書データ	-	三省堂大辞林約2万語	-	三省堂デイリー辞典約1万5千語	-	東京外国語大学ヒンディー語辞書7,524語		-	三省堂デイリー辞典約1万5千語
被験者	12名				6名			6名	

表 1 に示した実験 No.（A1～D2）を用いて以下に結果を示す．

5.2 結果

(1) 入力結果の分析

既存の入力方式と本研究が提案する「AjaxIME」方式における被験者の回答の平均比較を行い，日本語，英語，ポルトガル語の結果に対しては，「同一の被験者」が「両

a http://debugger.ru/projects/virtualkeyboard

方式を操作」する評価テスト方式のため，平均値の差の検定「対応のある t 検定（両側検定)」を実施した．

ヒンディー語については，「従来方式」，「AjaxIME 詳細情報なし」，「AjaxIME 詳細情報あり」の 3 方式に対する被験者の回答と操作数の平均比較を行うため，「同一の被験者」が 3 方式を操作する「対応のある一元配置の分散分析（３つの平均値の差の検定)」を実施した．また，「対応のある一元配置の分散分析」の結果，有意差がみられた要因については，多重比較(Tukey 法)による分析を行った．

（ａ）正答数

言語毎に既存の入力方式と AjaxIME の正答数の平均を比較した．

英語では有意な差はなかった．

日本語においては， A1 と A2 を比較した場合，A1 は平均 27.64，A2 は平均 26.00 であり，A1 の正答数が有意に高くなった（t(13)=2.16, p<.05）．

ポルトガル語において，D1 と D2 を比較した場合, D1 は平均 22.88, D2 は平均 28.25 であり，D2 の正答数が有意に高くなった（t(7)=3.80, p<.01）．

ヒンディー語において， 3 方式（C1，C2，C3）の平均の分散分析の結果，有意な差がみられた（p<.01）． さらに多重比較による分析を行った結果，C1 と C2 および，C1 と C3 においては有意な差が見られた（共に p<.05）．しかしながら C2 と C3 においては有意な差は見られなかった．

（ｂ）誤答数

言語毎に既存の入力方式と AjaxIME の誤答数の平均を比較した．

日本語，英語では，有意な差はなかった．

ポルトガル語において，D1 と D2 を比較した場合，D1 は平均 1.13，D2 は平均 0.13 であり，D2 の誤答数が有意に低くなった（t(7)=3.06, p<.05）．

ヒンディー語において，3 方式の平均の分散分析の結果，誤答数では有意な差がみられなかった．

（ｃ）回答率

「既存の入力方式」と「AjaxIME」の回答率（回答数/30）の平均を比較した．

英語では有意な差はなかった．

日本語において，A1 と A2 を比較した場合， A1 は平均 95%，A2 は平均 88%であり，A1 の回答率が有意に高くなった（t(13)=2.90, p<.05）．

ポルトガル語において, D1 と D2 を比較した場合, D1 は平均 80%, D2 は平均が 95%であり，D2 の回答率が有意に高くなった（t(7)=2.87, p<.05）．

ヒンディー語において，3 方式（C1，C2，C3）の平均の分散分析の結果，有意な差がみられた（p<.01）．さらに多重比較による分析を行った結果，C1 と C2 および，C1 と C3 においては有意な差が見られた（共に p<.05）．しかしながら C2 と C3 においては有意な差は見られなかった．

(2) **ヒアリングによる評価**

言語毎の入力実験終了後，被験者に対して AjaxIME についての主観評価をヒアリングした．結果を表 2 に示す．

表 2　主観評価結果

Table 2　The results of the subjective assessment

言語	ポジティブ評価	ネガティブ評価
日本語	・予測変換があるから良い ・同じ読みの異なり語を入力するときに詳細情報があると良い ・外国語を入力するには使えそう ・欧米圏の人には使いやすそう	・ヘボン式は使い慣れない ・読み方がわからないときは難しい ・慣れていないため使い勝手に難がある
英語	・候補がでてくるのがよい ・長い綴りのときに有効 ・日本語よりは既存の入力方式との違いが無い感じがする	・候補としてリストアップされる文字列は入力文字列でのソート順ではなく，類似した単語が良い（要望）
ヒンディー語	・詳細情報がある方があると良い ・読み方が出るから使いやすい ・似た単語でも判断しやすい	・転写規則を覚えるのがたいへん ・転写規則の一部に違和感がある
ポルトガル語	・英語よりもさらに使いやすい. ・言語によって違う可能性がある気がする	特になし

5.3 考察

入力結果の分析によると，ポルトガル語については，「AjaxIME」での入力の方が，正答数，回答率において有意に高く，誤答数においては有意に低い結果となった．また，ヒンディー語についても，「AjaxIME」での入力の方が，正答数，回答率において有意に高い結果となった．したがって，「AjaxIME」は不慣れな言語において，入力の速度および精度において効果的であり，「AjaxIME」により正確に早く入力できるという仮説を確認することができた．

しかし，ヒンディー語において詳細情報の有無が入力に与える影響については，入力結果の分析では充分確認することができなかった．ヒアリングによる評価では詳細情報の存在が高く評価されていることから，入力実験では限られた時間の中で入力数をこなすことに意識が集中するため，学習という視点から語彙の内容を理解するまでの余裕が持てなかったと見ることが出来る．

一方，日本語入力については，「AjaxIME」での入力の方が，正答数，回答率において有意に低い結果となった．日本語入力については，普段から仮名漢字変換をベースとする InputMethod に慣れており，かつ本実験では入力の際の変換規則をヘボン式に固定したため，被験者が使い慣れた InputMethod と比べて，初めて利用する「AjaxIME」では入力がしにくかったと考えられる．

6. おわりに

本研究により，情報資源の対象をインターネットで Web 公開されている Unicode（UTF-8）で記述されたあらゆるデータと置き，そのデータへアクセスするために必要な検索エンジンで指定する検索クエリーを入力するための多言語 InputMethod が実現できた．

本研究では，どのような言語にも対応可能な InputMethod という特徴に加え，変換候補となる語彙に関する情報である詳細情報の表示機能を実装したところに新規性がある．また，インクリメンタルサーチによる文字変換が，入力しようとしている言語に不慣れなユーザーが正確な綴りに早く導かれるきっかけをあたえるものであるという仮説設定のもとシステム実装を行い，評価実験とヒアリングにより，提案システムの有効性を示すことができた．

今後は，変換辞書を追加して対応言語を増やすとともに，入力実験被験者のヒアリングで指摘を踏まえ，変換辞書（転写規則）の違和感を少なくする工夫や，利用頻度や綴りの長さなどを利用して変換候補の出現順を変更するなど，よりユーザーにとって入力しやすい環境を整備していく予定である．

謝辞　本研究は総務省 SCOPE「ICT イノベーション創出型研究開発」「次世代インターフェースとしての多言語コンシェルジュの研究開発（東京外国語大学）」（平成 19-21 年）の支援を受けることにより研究推進が行えました．また，実験用辞書コンテンツは，株式会社三省堂書店様から貸与いただきました．その他，本研究の実験実施にあたり，多くの方にご協力いただきました．ここに深く感謝申し上げます．

参考文献

1)　The Unicode Consortium : The Unicode Standard, Version 5.0, Addison-Wesley Professional; 5th edition (2006)
2)　Ken Lunde : CJKV Information Processing, O'Reilly & Associates Inc(1999)
3)　横山詔一, ロング エリク, 米田純子, 和田志子, 黒田信二郎, 下川和男 ： 日本語 Web IME の開発と図書館情報検索システムへの実装, 情報処理学会研究報告, デジタル・ドキュメント, pp.43-47(2004)
4)　工藤 拓 : Ajax IME: Web-based Japanese Input Method, http://ajaxime.chasen.org/
5)　森健一 ： 日本語情報処理, テレビジョン学会誌, 33(5), pp.380-385(1979)
6)　町田和彦 ： 華麗なるインド系文字, 白水社(2001)
7)　Jef Raskin : The humane interface: new directions for designing interactive systems, Addison-Wesley Publishing Co. (2000)
8)　高林哲, 小松弘幸, 増井俊之 : Migemo : 日本語のインクリメンタル検索, 情報処理学会論文誌, Vol.43(12), pp.3698-3705(2002)
9)　L.R.Peterson and M.J.Peterson : Short-term Retention of Individual Verbal Items, Journal of Experimental Psychology, 58, pp.193-198(1959)
10)　G.A.Miller : The Magical Number Seven, Plus or Minus Two Some Limits on Our Capacity for Processing Information, Psychological Review 63 (2), pp.343-355(1956)

著者紹介

池田佳代（正会員）
千葉大学画像工学科卒業．2006 年東京理科大学大学院修士課程修了（技術経営修士）．2010 年電気通信大学電子通信学研究科博士課程修了．博士(学術)．現在，情報処理推進機構(IPA)研究員，エクセリードテクノロジー取締役．主に文字情報工学，多言語処理の研究に従事．

沼田秀穂（正会員）
1958 年生まれ．電気通信大学大学院電気通信学研究科博士課程修了．博士(工学)．専門社会調査士．現在，武蔵野大学非常勤講師，情報処理推進機構(IPA)研究員，エクセリードテクノロジー代表取締役．主に ICT が社会システムに与えるインパクト研究，多言語処理の研究に従事．

兼子正勝
1953 年生まれ．東京大学大学院人文科学研究科博士課程単位取得退学．パリ第 10 大学文学博士．現在，国立大学法人電気通信大学情報理工学部総合情報学科教授．フランス現代哲学をベースにメディア理論，視覚表現理論を研究し，現在はメディアコンテンツの分析・デザインを研究範囲に収める．

町田和彦
1951 年生まれ．1978 年アラハバード大学修士課程修了．1980 年東京外国語大学修士課程修了．ヒンディー語専攻．東京外国語大学アジア・アフリカ言語文化研究所教授．著書「ことたび　ヒンディー語」「書いて覚えるヒンディー語の文字」「ニューエクスプレス　ヒンディ語」，編著「華麗なるインド系文字」．