商品情報登録データベース検索における PRECIS Framework の応用と評価

根本 貴弘[1,2]　Andrew TANTOMO[1]　杉浦 一徳[1]

概要：本研究では，現在，IETF（Internet Engineering Task Force）にて標準化中である PRECIS（Preparation and Comparison of Internationalized Strings）Framework を応用し，ユーザ入力によるデータベース検索における検索文字列の比較一致の精度の向上を目指す．本研究では，PRECIS Framework の応用として，文字列の変換・正規化に関する処理順序及びアプリケーションで利用可能とする文字コード群を定義し，PRECIS Framework を参照実装した API を文字列の前処理ライブラリとして用い，ユーザ入力による商品情報登録データベースとその情報の共有を行う SNS サービスを対象に評価実験を行った．そして，PRECIS Framework 適応前後で，ユーザが入力した商品名によってデータベースに登録された商品名が正しく参照されるか，また，文字列の比較一致の精度が向上するか評価を行った．
その結果，データベース検索を行った際に，PRECIS Framework 適応前は一致しない記号論ないし意味論上同一とみなせる文字のうち，PRECIS Framework 適応後は全角半角文字及び合成済み文字・結合文字列，言語依存した文字の比較が一致しするようになることが明らかとした．本研究により，PRECIS Framework は国際化文字列を扱うデータベース検索においても有効であることが明らかとなった．

An Application and Evaluation of PRECIS Framework on Data Base Retrieval

Takahiro NEMOTO[1,2]　Andrew TANTOMO[1]　Kazunori SUGIURA[1]

1. はじめに

1.1 研究背景

世界中の人々がインターネット上のサービスを利用するようになり，サービス内のコンテンツは様々な言語で表現し，様々な言語に対応した文字セットや入出力方式が作られてきた．しかし，利用者にとって記号論ないし意味論上同一とみなせる文字がシステム内で複数の異なる数値コードとして存在する場合，入力文字列が利用者の意図した文字列と一致しているか一貫した方法による比較は困難である．そのため，インターネット上の情報資源を参照する際に，利用者が使用する文字入力環境によっては利用者が意図しているサービスに登録されている文字列と入力された文字列が一致せず，意図した情報資源が参照出来ないという問題が起こる．また，同様に他のサービスないしシステムと情報資源の共有を行う場合においても，記号論ないし意味論上同一とみなせる文字が異なる文字コードで登録されている情報資源は正しく相互に活用することが困難である．例えば，日本語の場合，「ダルマ」という単語に対して，合成済み文字を用いた文字列や結合文字列を用いた文字列，半角文字を用いた文字列など複数の表現方法がある．人間にとってはそれぞれ同じモノを指す単語であるが，システムの場合はそれぞれ別のモノを指す言葉として見なされるため，相互参照は困難となる．

1.2 研究目的

本研究では，現在，IETF（Internet Engineering Task Force）[1] にて標準化中である PRECIS（Preparation and Comparison of Internationalized Strings）Framework[2][3] を応用し，ユーザ入力によるデータベース検索における検索文字列の比較一致の精度の向上を目指す．PRECIS Framework とは，通信プロトコルにおいて国際化された識

1　慶應義塾大学大学院メディアデザイン研究科
2　青山学院大学附置情報メディアセンター

別子を利用することによって必要となる文字列の前処理及び比較をするためのフレームワークである．

本研究では，PRECIS Framework の応用として，文字列の変換・正規化に関する処理順序及びアプリケーションで利用可能とする文字コード群を定義し，PRECIS Framework を参照実装した API を文字列の前処理ライブラリとして用い，ユーザ入力による商品情報登録データベースとその情報の共有を行うアプリケーションを対象に評価実験を行う．そして，PRECIS Framework 適応前後で，ユーザが入力した商品名によってデータベースに登録された商品名が正しく参照されるか，また，文字列の比較一致の精度が向上するか評価を行う．

1.3 本論文の構成

本論文の構成は次の通りである．本論文は全 9 章から構成される．第 2 章でアマチュアモーターレースにおける現状及び課題を整理する．その上でドライバー，ピットクルー間のコミュニケーションにて必要とされる要件を定義する．第 3 章ではその情報を即時的に共有するためのネットワーク環境，またデジタルピットボードシステム，及びデジタルピットボードシステムを利用した，ドライバーとピットクルー間のコミュニケーションを可能とするシステムを提案する．第 4 章で提案のシステムを実際のレースイベントで行った際の評価実験の結果を述べる．第 5 章では評価実験の結果を元に考察を行う．第 6 章にて前章での考察結果を踏まえ今後の課題を述べる．第 7 章で本研究の結論をまとめる．

2. 国際化文字列とデータベース検索における現状と課題

本章では，国際化文字列とデータベース検索における現状と課題について述べる．

2.1 国際化文字列における課題

2.1.1 情報入力の課題

利用者を取り巻く文字入力環境は利用者が使用する端末やアプリケーションによって利用される文字セットやエンコーディング方式が異なる．そのため，利用者にとって同一とみなせる文字がシステム内で複数の異なる数値コードとして存在する場合，意図した情報資源を参照するために利用者は自分が使用している文字入力環境を意識する必要がある．しかし，利用者が使用する端末やアプリケーション毎に，視覚的ないし機能的に同等な文字列の比較を利用者が行うことは困難である．そこで，視覚的ないし機能的に同等な文字と文字コードの対応関係を 1 対 1 とするために，端末やアプリケーションによっては半角カタカナ文字を禁止するというようにキーボード入力出来る文字を制限することが行われている．しかし，入力出来る文字を制限したとしても，コピー＆ペーストのように他のコンテンツの文字列を用いることが可能であれば，制限したはずの文字列を入力することが行えてしまう．また，全ての端末やアプリケーションにおいて入力環境を統一することも困難である．そのため，利用者が入力している文字の文字コードを意識すること無く，入力された文字列に対して適切な前処理を行うことで，文字列の比較一致の機会を増やし，意図した情報資源を参照可能とする仕組みが必要となる．

2.1.2 文字列処理における課題

一方で，記号論ないし意味論上同一とみなせる文字には，英字の大文字・小文字や全角・半角文字や日本語の濁点・半濁点やアクセント記号を利用する文字などの合成済み文字・結合文字列，ギリシャ語やトルコ語等の言語に依存した文字変換を行う文字，類義語等様々な文字が存在する．一つのアプリケーションやサービスが様々な国で利用されることが想定される今日において，利用者が英数字同様に正しく母語をサービスやアプリケーションで使用するためには，意味論上同一とみなせる異なる文字コードを正しく同一の文字として扱うための手法が必要となる．

また，言語や文字は政治や文化的な背景により時代とともに変化してきたという歴史的な事実に基づく普遍的なものではないという特徴がある．そのため，世界中の文字を一つの文字コードに全て収録することを目的とした Unicode においても，現状では世界中全ての文字の収録は完了しておらず，現在も改版作業が続けられ，文字セットに新たな文字が追加や改訂が行われている．そのため，システムで利用可能な文字をある特定の文字セットのバージョンに基づき定義していては，文字セットの改版の都度，システムで利用可能な文字の再定義を行わなければならないという問題が起こる．しかし，現在，システムが利用する文字セットに新たな文字が追加された場合，その文字が様々なプトロコルの目的に応じて安全に利用可能な文字であるか定義する仕組みは存在していない．

2.2 データベース検索における課題

本節では，ユーザ入力によるデータベースへの商品情報の登録及び検索を行う Shop-banzai![4] を例にあげ，データベース検索における課題を述べる．Shop-banzai!とは，バーコードを識別子としたユーザ入力による商品情報登録データベースとその情報の共有を行う SNS サービスである．バーコードの他にユーザが登録した商品名によって商品の参照を行うため，利用者の入力文字列と商品名の登録文字列に記号論ないし意味論上同一とみなせる異なる文字コードが存在する場合，利用者は意図した情報資源を正しく参照することが困難となる．

図 1 に Shop-banzai!の検索の現状を示す．図中の登録情報画面の赤四角で囲んだ部分が，登録された商品名「レッドブル」を示している．検索画面（全角文字）及び検索画面

(半角文字) の赤四角で囲んだ部分には検索結果として登録された商品名が表示される．全角文字による検索では意図した情報を参照可能であるが，同様の単語を半角文字を用いて検索した場合，情報を参照することは不可能である．

図 1 Shop-banzai!における検索の現状

また，表 1 に Shop-banzai!を対象に行った，商品情報検索における比較に関する調査結果の抜粋を示す．表中の各調査項目に対し，例の列中の左の文字列を検索キーワードとして入力し，右の文字列を検索することが可能か調査を行った．そして，その結果を，比較結果の列中に示した．「○」は検索であった調査項目，「×」は検索不可能である．この調査結果より，英字の大文字・小文字やギリシャ語のファイナルシグマのような文脈に依存して字形の変わる文字は，意味論上同一の異なる文字でも利用者の意図した文字として検索の対象となっている．一方で，日本語の濁点・半濁点やアクセント記号を利用する文字や，言語に依存した文字変換を行う文字，同義語等，利用者が意図した文字は検索の対象とされないという文字の比較照合に関する問題が起きている．

表 1 Shop-banzai!における検索の結果 (抜粋)

調査項目		例		検索結果
大小文字		TEST	test	○
全半角文字	英語	ｔｅｓｔ	test	×
	日本語	ﾃｽﾄ	テスト	×
	記号	：）	:)	×
合成済文字・結合文字列		ダルマ	ﾀﾞﾙﾏ	×
文脈依存文字		σ	ς	○
言語依存文字		i	i	×
同義語	平・片仮名①	てすと	テスト	×
	平・片仮名②	ば	バ	×
	平・片仮名③	ゔ	ヴ	×
	他言語	test	テスト	×

Shop-banzai!ではデータベースに MySQL を利用しており，MySQL ではより比較一致の機会を向上させるために，the Unicode Collation Algorithm(UCA)[5] で定められたアルゴリズムに従った utf8_unicode_ci という，大文字・小文字や全角・半角文字や合成済み文字・結合文字列を記号論ないし意味論上同一の文字としてみなす，比較手法がある [6]．表 2 に utf8_unicode_ci により同一とみなすことが可能となった文字（抜粋）を示す．これら一部の日本語の他にドイツ語のエスツェットと英字の S やリガチャー（合字）とその基底文字となる英字を同一とみなせるように比較一致の機会を向上させている．しかし，表 2 に示す通り，日本語においては，基底文字が同じであれば大文字・小文字や全角・半角文字や合成済み文字・結合文字列，ひらがな・カタカナ等を同一にみなせる一方で，濁点・半濁点がない文字とある文字を同一視しており，例えば，「ゴハン」と「コバン」が同じ文字列としてみなされる等，正しい情報資源を参照することが困難となっている．

表 2 utf8_unicode_ci により同一視される文字群 (抜粋)

Unicode	3041	3042	30A1	30A2	32D0	FF67	FF71
Character	ぁ	あ	ァ	ア	㋐	ｧ	ｱ

Unicode	3045	3046	3094	30A5	30A6	30F4	32D2	FF69	FF73
Character	ぅ	う	ゔ	ゥ	ウ	ヴ	㋒	ｩ	ｳ

Unicode	304B	304C	3095	30AB	30AC	30F5	32D5	FF76
Character	か	が	ゕ	カ	ガ	ヵ	㋕	ｶ

Unicode	306F	3070	3071	30CF	30D0	30D1	31F5	32E9	FF8A
Character	は	ば	ぱ	ハ	バ	パ	ㇵ	㋩	ﾊ

本章で述べてきたように，国際化文字列を用いた検索の比較一致の機会を向上させるためには，利用者が入力している文字の文字コードを意識すること無く，入力された文字列に対して記号論ないし意味論上適切な前処理を行うことで，文字列の比較一致の機会を増やし，意図した情報資源を参照可能とする仕組みが必要となる．

3. 提案

本章では，商品情報登録データベース検索における PRECIS Framework の応用として，文字列の変換・正規化に関する処理順序及びアプリケーションで利用可能とする文字コード群を定義し，PRECIS Framework を参照実装した API を文字列の前処理ライブラリについて述べる．

3.1 PRECIS Framework 概要

PRECIS Framework とは，通信プロトコルにおいて国際化された識別子を利用することによって必要となる文字列の前処理及び比較をするためのフレームワークである．PRECIS Framework は，下記の 2 つのアルゴリズムと 1 つのプロファイルによって成り立つ．

- システム内で複数の数値コードとして登録されている記号論ないし意味論上同一の文字群を同一の文字として扱うためのアルゴリズム
- 文字セットのバージョンに依存せず，プロトコルで安全に利用可能・不可能な文字を分類するアルゴリズム
- それらアルゴリズムを用いて，従来のプロトコルの仕組みに影響を与えず，文字列の比較照合を行う前処理フレームワークと各プロトコルで利用するためのプロファイル

3.2 システム内で複数の数値コードとして登録されている記号論ないし意味論上同一の文字群を同一の文字として扱うためのアルゴリズム

システム内で複数の数値コードとして登録されている記号論ないし意味論上同一の文字群を同一の文字として扱うためのアルゴリズムとして，PRECIS Framework では 以下の文字の変換と正規化，処理の順序に関する文字列前処理アルゴリズムを定義している．

3.2.1 文字の変換

文字の変換では，1) 大文字から小文字に文字種の統一（Casefolding），2) 各言語で一般的に使用される文字幅の統一（Width mapping），3) プロトコルが独自に定義した文字変換，4) 言語・文脈に依存した変換規則を持つ文字変換 (Local case mapping) を扱う．これら 1)，2)，4) は Unicode Character Database の情報に基づき変換を行い，3) はプロトコルの定義に従い変換を行う．

3.2.2 正規化

正規化は基本的に NFC を扱う．これは，NFKC の正規化処理の中には利用者の混乱をまねく恐れのある文字が含まれているためである．ただし，使用するプロトコルの用途により，NFKC を使用しても問題ない場合は扱えるものとし，その際には NFC 及び Width mapping は扱わない．

3.2.3 文字列の前処理を行う順序

これらの文字の変換と正規化を用い，人間の理解レベルとシステムの理解レベルを補完する前処理の順序を定義している．順序を定義する理由は，処理の順序により文字列の前処理後の結果が異なったり，利用者が意図していない文字に変換されてしまったりという問題を解決するためである．PRECIS Framework では，このような検討を各処理に対して行い安全に文字列の比較照合を行うための前処理として以下の順序が定義されている．

(1) Width mapping（全角半角文字変換）
(2) Additional mappings
  (a) Delimiter mapping（区切り文字変換）
  (b) Special mapping (空白文字変換及び制御文字削除)
  (c) Local case mapping（言語依存文字の大文字小文字変換)
(3) Casefolding（大文字小文字変換）
(4) Normalization (正規化: Unicode 正規化形式 (D，KD，C，KC))

3.3 文字セットのバージョンに依存せず，プロトコルで安全に利用可能・不可能な文字を分類するアルゴリズム

特定の文字セットのバージョン依存問題を解決するために，各プロトコルで安全に使用可能なコードポイントリストへの追加を許容する文字を特定するアルゴリズムを設計する必要がある．この手法は，1) 様々なプロトコルの目的を満たすために Unicode コードポイントの性質に基づいた再整理を行い分類し，2) 分類に基づきコードポイントがプロトコルで利用可能かを記述した派生特性値を判定するアルゴリズムである．

この仕組みより Unicode のバージョン変更が起こっても特性値を算出することが可能である．つまり Unicode プロパティ情報に変更があった場合でも，この仕組みがその都度プロトコルで利用可能な文字を計算するため，Unicode の変更の影響を受けずに文字列前処理フレームワークを利用可能としている．

また，Unicode コードポイントの性質に基づいて分類された文字群を更に様々なプロトコルの目的に合わせて利用可能な文字を分類するために，紛らわしい文字を排除し，比較した際に一致する機会を増やすための IdentifierClass とパスワードなどで，文字が一致する機会を少なくするための FreeformClass の２つのクラスが用意している．プロトコルは，この２つのクラスに基づき使用する文字群を選択し，前処理後の文字が利用可能か不可能かを判定する．

3.4 文字列の比較照合を行う前処理フレームワークと各プロトコルで利用するためのプロファイル

上記の２つのアルゴリズムをプロトコルの目的に応じて利用可能とするためのプロファイルを提案する．プロトコルで本提案フレームワークを利用するために，プロファイル設計者は 1) システム内で複数の数値コードとして登録されている意味論上同一の文字群を同一の文字として扱うためのアルゴリズムにて定義する前処理の順序に基づき利用する前処理を指定し，2) 文字セットのバージョンに依存せず，プロトコルで安全に利用可能・不可能な文字を分類するアルゴリズムに基づき比較対象とする文字クラスを指定する．

3.4.1 PRECIS Framework とユーザ入力によるデータベース登録及び検索の親和性

2 で述べたように，国際化文字列を用いた検索の比較一致の機会を向上させるためには，利用者が入力している文字の文字コードを意識すること無く，入力された文字列に対して記号論ないし意味論上適切な前処理を行うことで，文字列の比較一致の機会を増やし，意図した情報資源を参照可能とする仕組みが必要となる．

PRECIS Framework は国際化文字列を識別子やパスワードとしてプロトコル中で正しく使用するために策定されているフレームワークであり，記号論ないし意味論上同一の文字群を同一の文字として扱うためのアルゴリズムや文字セットのバージョンに依存せず，プロトコルで安全に利用可能・不可能な文字を分類するアルゴリズムなどは，Shop-banzai!のようなユーザ入力により登録した文字列（商品名）を識別子として利用し，データベース内の情報資源を検索するシステムにおいても有効である．また，入

力された文字列に対して記号論ないし意味論上同一の文字として扱うための前処理は，utf8_unicode_ci とは異なり，大文字・小文字や全角・半角文字や合成済み文字・結合文字列を同一に扱うことが可能であり，かつ，濁点・半濁点がない文字とある文字を区別するため，「ゴハン」と「コバン」の様に異なる意味を持った文字列を異なる文字列として扱うことが可能である．

3.5 preciskit を用いた PRECIS Framework のデータベース検索への応用

本節では，PRECIS フレームワークの機能を参照実装した国際化文字列の比較を行うためのライブラリ (preciskit) を用い，データベース検索における PRECIS Framework の応用方法を提案する．preciskit は，インターネットのプロトコルが扱う識別子の国際化が進んでいる今日において，ドメイン名，メールアドレス，メッセンジャーアドレス，ログイン ID/パスワード等，アプリケーションはそれらを適切に扱う必要がある背景を受けて，アプリケーションが複数の国際化された識別子を同時に扱うために必要な機能を実装している．preciskit 構成図を図 2 に示す．

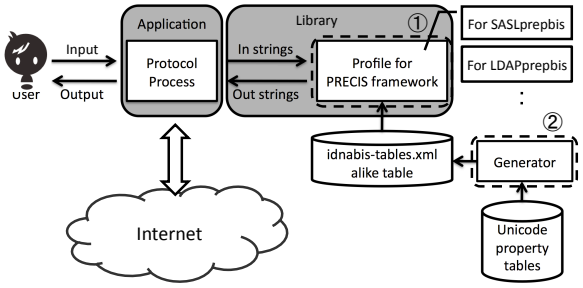

図 2 preciskit 構成図

1 は PRECIS フレームワークで定義した，システム内で複数の数値コードとして登録されている意味論上同一の文字群を同一の文字として扱うためのアルゴリズムに基づき文字列の変換，比較を PRECIS フレームワークプロファイルに基づき行う．2 は Unicode プロパティテーブル群を利用し，文字セットのバージョンに依存せず，プロトコルで安全に利用可能・不可能な文字を分類するアルゴリズムに基づき PRECIS におけるコードポイントの派生特性値を計算し，その結果を記載したテーブルを出力する [7].

本研究では，商品情報の登録及び検索を対象とした，文字列処理の種類と順序のみを定義するが，Shop-banzai!においても，商品情報の登録及び検索の他にログイン ID やパスワードを利用するためそれらの目的に合わせた文字列の前処理が必要となるため，ログイン ID やパスワードにおいては，IETF で標準化中である PRECIS Framework のプロファイルを利用することが可能である．

3.5.1 文字列処理の種類と順序

本研究では，商品情報の登録及び検索を対象に，以下のように文字列処理の種類と順序を定義する．

(1) Width mapping（全角半角文字変換）
(2) Additional mappings
  (a) Special mapping (空白文字変換及び制御文字削除)
  (b) Local case mapping（言語依存文字の大文字小文字変換)
(3) Casefolding（大文字小文字変換）
(4) Normalization（Unicode 正規化形式 (C)）

Width mapping（全角半角文字変換）を使用する理由は，Normalization（Unicode 正規化形式 (C)）を使用するためである．Normalization（Unicode 正規化形式 (KC)）ではなく，Normalization（Unicode 正規化形式 (C)）を使用する理由は，Normalization（Unicode 正規化形式 (KC)）は半角を全角に正規化することで異なる文字幅を同じ文字幅として扱える利点があるため特定の文脈で中心的な意味を確認するためには便利であるが，文字に不適切な変換を行う場合があり，商品情報のような様々な文字入力が期待される環境では避ける必要があったためである．Special mapping（空白文字変換及び制御文字削除）は，空白文字がユーザの入力ミスにより連続して入力された場合において，一つの空白文字へと変換を行ったり，意図しない制御文字が含まれていた場合に，削除するために行う．Local case mapping（言語依存文字の大文字小文字変換）は，ドイツ語のエスツェットやギリシャ語のファイナルシグマ，トルコ語のドット無し i，ドット付き I など，いくつかの文字はその文字を使用する言語圏固有の変換規則を持ち，他の言語圏の変換規則とは異なっているために必要である．Casefolding（大文字小文字変換）は英語等の大文字小文字のある文字を変換するために用いる．また，最後に商品情報の比較一致の機会を向上させるために，利用可能な文字コード群として IdentifierClass を使用する．

4. 評価

本章では，3 章で述べた preciskit を用いたデータベース検索における PRECIS Framework の有効性を評価する．

4.1 評価概要

データベース検索における PRECIS Framework の有効性を評価するために，バーコードを識別子としたユーザ入力による商品登録データベースとその情報の共有を行う SNS サービスである Shop-banzai!において，バーコードの他にユーザ登録による商品名によって商品名の参照を行うため，利用者の入力と商品名の登録に PRECIS フレームワークを利用する．これによりデータベース検索を行った際に，PRECIS フレームワーク適応前は一致しない同等の意味を持つ全半角文字及び合成済み文字・結合文字列の比

較が一致し，参照可能となるか評価を行った．図 3 にシステム図を示す．

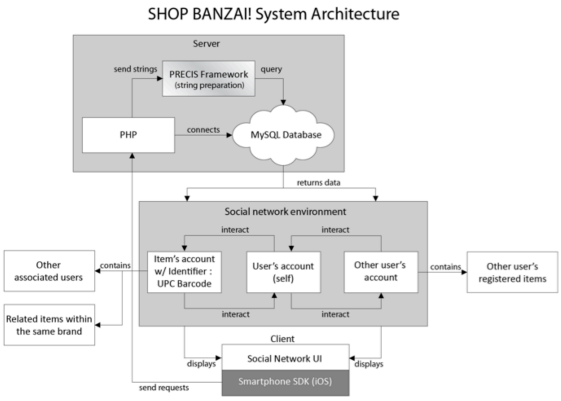

図 3　システム図

4.1.1　PRECIS フレームワークの位置付け

利用者が入力した製品名を商品情報を参照するための識別子として利用し，サーバ側で 3 章で述べた文字列処理の種類と順序及び IdentifierClass を用いて入力された文字列を正規化し，データベースに登録する．また，データベース検索の際の利用者の入力に対しても同様に PRECIS フレームワークを適応し，検索結果が向上することを確認する．

4.1.2　実験結果

表 4 に商品名とテスト項目の抜粋を示す．全角・半角文字の正規化により，製品の比較一致の精度が上がったことが明らかとなった．しかし，例えば「お〜いお茶」に見られる WAVE DASH(U+301C：〜) のような特殊な記号を利用している商品も存在し，これらの文字には前処理は適応されず，IdentifierClass では禁止文字となるため，より多様な商品名を検索可能な形で登録するためには FreeformClass を利用することが望ましいことがわかった．

図 4　データベース検索における PRECIS Framework 適応結果 (抜粋)

登録商品名	大文字			文字幅			結合文字列		
	入力文字列	適用前	適用後	入力文字列	適用前	適用後	入力文字列	適用前	適用後
danboard	DANBOARD	○	○	ｄａｎｂｏａｒｄ	×	○	･	･	･
レッドブル	･	･	･	ﾚｯﾄﾞﾌﾞﾙ	×	○	レッドブル	×	○

また，図 5 に PRECIS Framework 適応後の Shop-banzai! における検索画面を示す．PRECIS Framework 適応前では，表示されていなかった，半角文字列に対する検索結果として，同様の単語を意味する全角文字列の商品名が検索されるようになった．

表 3 に PRECIS Framework 適応後の Shop-banzai!を対象に行った，商品情報検索における比較に関する評価結果

図 5　PRECIS Framework 適応後の Shop-banzai!における検索

を示す．表中の「○」は検索であった調査項目，「×」は検索不可能を示している．この評価結果より，適応前より検索対象となっていた英字の大文字・小文字やギリシャ語のファイナルシグマのような文脈に依存して字形の変わる文字に加え，PRECIS Framework 適応後日本語の濁点・半濁点やアクセント記号を利用する文字や，言語に依存した文字変換を行う文字も検索対象となった．しかし，同義語については検索の対象とされないままであった．

表 3　PRECIS Framework 適応後の Shop-banzai!における検索の結果 (抜粋)

調査項目		例		検索結果
大小文字		TEST	test	○
全半角文字	英語	ｔｅｓｔ	test	○
	日本語	ﾃｽﾄ	テスト	○
	記号	:)	:)	○
合成済文字・結合文字列		ダルマ	ﾀﾞﾙﾏ	○
文脈依存文字		σ	ς	○
言語依存文字		İ	i	○
同義語	平・片仮名①	てすと	テスト	×
	平・片仮名②	ば	バ	×
	平・片仮名③	ﾌﾞ	ヴ	×
	他言語	test	テスト	×

5.　考察

本研究では，PRECIS Framework の応用として，文字列の変換・正規化に関する処理順序及びアプリケーションで利用可能とする文字コード群を定義し，PRECIS Framework を参照実装した API を文字列の前処理ライブラリとして用い，ユーザ入力による商品情報登録データベースとその情報の共有を行うアプリケーションを対象に評価実験を行った．そして，PRECIS Framework 適応前後で，ユーザが入力した商品名によってデータベースに登録された商品名が正しく参照されるか，また，文字列の比較一致の精度が向上するか評価を行い，その結果，データベース検索を行った際に，PRECIS Framework 適応前は一致しない記号論ないし意味論上同一とみなせる文字のうち，PRECIS Framework 適応後は全角半角文字及び合成済み文字・結合文字列，言語依存した文字の比較が一致するようになる

ことが明らかとした．本研究により，PRECIS Framework は母語を扱うデータベース検索においても有効であることが明らかとなった．

6. 今後の課題

本章では，国際化文字列とデータベース検索における現状と課題について述べる．

6.1 同義語検索における課題

本研究が提案する PRECIS Framework ではひらがな・カタカナ，日本語・英語などの同義語の検索精度を向上させることは出来なかった．しかし，PRECISFramework は本来インターナショナライゼーションのためのフレームワークであるため，日本語のひらがな・カタカナや漢字の異体字やアラビア語やペルシア語のアリフマクスーラのようなローカライゼーションに関する処理は，利用者の利便性を向上させるために必要な処理となる．

ただし，利用者が期待する全ての同義語・異体字等を考慮することは困難である．これは，単語や成句の異なるつづりの数は膨大なものになるため，利用者は同じ意味を持つ異なるつづりを等価なものとして処理するよう期待するかもしれないし，等価ではないものとして処理するよう期待するかもしれないため比較が困難であるためである．等価な意味を持つ異なるつづりの例として，アメリカ英語とイギリス英語の”theater”と”theatre”が挙げられる．

また，他の例として，簡体字の名前のつづり (例えば”日本国”) と等価な意味を持つ繁体字のつづり (例えば”日本國”) が挙げられる．”Aepfel”とウムラウト付き”Äpfel”のような，言語固有の等価な意味を持つ異なるつづりの場合，ドイツ語ではしばしば等価なものと見なされるが，他の言語では等価であるとは見なされない場合がある．

このような理由から，本研究では，全ての同義語・異体字の検索結果を PRECIS Framework で解決することは困難であり，他の検索アルゴリズムを用いる必要がある．しかし，異体字問題に詳しい専門家が PRECIS Framework を拡張して異体字比較処理を実現するための方法として，IDNA2008 で用いられる TR46 のような日本語の区切り文字「。」を英字のピリオド「.」に変換する仕組みと同様に実装することが可能であるため今後その拡張法についての研究を行う必要がある．

6.2 コンテンツの国際化と可視化

今回は，サーバ側の処理として PRECIS Framework の適応を行ったが，既存の運用されているシステムをそのまま使うことができ，そこへは一切手を加える必要がなく，アプリケーション側の対応のみで国際化を実現するための仕組みとして，クライアント側の処理として行う必要がある．また，サーバ側の処理として PRECIS Framework を利用した場合，ユーザが入力したと思っている文字とサーバ側で処理されている文字は異なるため，ユーザの混乱を招く恐れもある．そのため，クライアント側で PRECIS Framework の機能を提供し，文字列処理後の結果をユーザが確認出来るように可視化を行うためのライブラリが必要となる．

特に可視化に関する特徴的な課題として，ヘブライ語やアラビア語などのスクリプトは文字を右から左へ表記する双方向性文字がある．このような文字を双方向性文字と呼び文書としてメモリに格納するためには文字の視覚的な順序とは別に論理的な順序を用いることが一般的とされている．しかし，左から右と右から左へ表記する文字が混在するような文書を扱うアプリケーションにおいて，適切な可視化を行う必要がある．

HTML では，Unicode の双方向アルゴリズムを用い，文字コード毎に定められた特性値からこのような表記に関する課題を解決している．この可視化に関する課題はプロトコルで利用する文字列においても重要で，利用者が文字を入力した際に正しいインターネット上の資源を参照していたとしても，参照先を示す文字列の可視化に関する部分が間違っていた場合，利用者は自分が誤ったインターネット上の資源を参照していると思うかもしれない．そのため，可視化に関しする利用者の混乱を招きやすい文字の扱いに関しても考慮する必要がある．

7. 結論

本研究では，現在，IETF にて標準化中である PRECIS Framework を応用し，ユーザ入力によるデータベース検索における検索文字列の比較一致の精度の向上を目指した．本研究では，PRECIS Framework の応用として，文字列の変換・正規化に関する処理順序及びアプリケーションで利用可能とする文字コード群を定義し，PRECIS Framework を参照実装した API を文字列の前処理ライブラリとして用い，ユーザ入力による商品情報登録データベースとその情報の共有を行うアプリケーションを対象に評価実験を行った．そして，PRECIS Framework 適応前後で，ユーザが入力した商品名によってデータベースに登録された商品名が正しく参照されるか，また，文字列の比較一致の精度が向上するか評価を行い，その結果，データベース検索を行った際に，PRECIS Framework 適応前は一致しない記号論ないし意味論上同一とみなせる文字のうち，PRECIS Framework 適応後は全角半角文字及び合成済み文字・結合文字列，言語依存した文字の比較が一致しするようになることが明らかとした．本研究により，PRECIS Framework は母語を扱うデータベース検索においても有効であることが明らかとなった．

参考文献

[1] The Internet Engineering Task Force (IETF), http://www.ietf.org/, (2014.05.01).

[2] P. Saint-Andre and M. Blanchet. PRECIS Framework: Preparation and Comparison of Internationalized Strings in Application Protocols draft-ietf-precis-framework-09, March 2013.

[3] Y. Yoneya and N. Nemoto. Mapping characters for PRECIS classes draft-ietf-precis-mappings-04, August 2013.

[4] SHOP BANZAI! 入手先 ⟨http://www.shopbanzai.com/⟩ (2014.05.01).

[5] Unicode Technical Standard #10 Unicode Collation Algorithm, http://www.unicode.org/reports/tr10/, (2014.05.01).

[6] 10.1.14.1 Unicode Character Sets, http://dev.mysql.com/doc/refman/5.7/en/charset-unicode-sets.html, (2014.05.01).

[7] createtables.rb, 入手先 ⟨http://stupid.domain.name/idna/⟩ (2014.05.01).