唐代人物知識ベースについて
～ 漢字文献知識ベース構築へ向けて ～

The Tang Person Knowledge Base: Towards a Knowledge Base of Chinese-Character-Based Documents

山本 一登
Tadato YAMAMOTO

京都大学／人文科学研究所、京都市左京区北白川東小倉町 47
Institute for Research in Humanities, Kyoto University, 47 Higashiogura-cho, Kitashirakawa, Sakyo-ku, Kyoto

あらまし：「唐代人物知識ベース」とは、京都大学 21 世紀 COE「東アジア世界の人文情報學研究教育據點」[1]の一環として行なわれているプロジェクトのひとつである「漢字文献ナリッジベース構築」のサブプロジェクトとして進めてきたものである。本システムは、ネットワークを介したウェブアプリケーションとして利用者にサービスを提供するデータベースであり、以下の URL から利用できる。

http://coe21.zinbun.kyoto-u.ac.jp/knowledge/person/index.html

本論では、現在実装している検索機能の紹介、および使用しているデータベースエンジンなどについて順次述べ、今後の展望についても述べる。

Summary: The Tang Person Knowledge Base advances as sub project of "Construction of a Knowledge Base of Chinese-Character-Based Documents" which is one of the project promoted as part of the Kyoto University 21st Century COE, "East Asian Center for Informatics in Humanities". We release the Tang Person Knowledge Base service as the web application which utilizes Internet, the URL is following.

http://coe21.zinbun.kyoto-u.ac.jp/knowledge/person/index.html

In this paper, we describe the search function which is implemented now, about the spec of our database system which we have used, and concerning future views.

キーワード：知識ベース、唐代、漢字文献、人文科学

Keywords: knowledge base, Tang dynasty, Chinese character literature, humanities

1 はじめに

本知識ベースは中国学を研究対象としている大学の学部生、院生や若手研究者などを主な利用対象者に想定しているため、検索できる事項は中国学を勉強する際の基礎となるような項目を選んだ。現在、唐五代の人物、約 4000 件の人物データ [9] に対して人名（姓、諱、字、排行、名号、謚、廟号）、関係地（生地、貫籍、郡望、寓居、卒地）、在世時（在世時、生年、没年）、家系（祖、父、祖先、子孫、家族、姻戚）、経歴（科挙、官職、著作、記事）などの情報が格納されている。データ量としては、まだまだ不十分ではあるが、これらの項目に対して検索が可能である。

本知識ベースの Web デザインには W3C の XHTML 1.1 [4]、W3C の CSS 2.1 [5] の標準規格に準拠し、Mozilla Firefox での閲覧に最適になるように開発を行ってきている。Web の標準仕様に従うようにデザインを進める理由は、特定の Web ブラウザに依存しない Web サイト作りを強く意識しているからである。

また、本データベースシステムでは、データフォーマットは XML 形式、データベースエンジンにはネイティブ XML データベースのひとつである eXist [7] を採用し、検索システムの実装を行なった。

本報告では、現在実装している検索機能の利用方法、および eXist 固有の特徴などについて順次述べ、最後に今後のプロジェクトの展望について述べる。

[1] http://coe21.zinbun.kyoto-u.ac.jp/

2 利用上の注意

格納しているデータの文字コードは Unicode 標準 [6] の 1 つのエンコード方式である UTF-8 を採用した。従って、Unicode 標準で定義されている漢字はすべて使用対象となっている。そのため、使用するコンピュータに Unicode 標準で定義されている漢字をすべて表示できるようなフォントがインストールされていることが望ましい。現状ではフォントに含まれていないグリフ（字形）は表示できないことに注意してほしい。フォントが使用できない利用者向けに画像を表示したりする機能は現在は考えていない。幸い、Linux、Mac、Windows などで使用できる Unicode 標準の漢字がすべて含まれたフリーなフォントがあるので、それをインストールすればよいだろう（入手方法は付録参照)。ただし、古い OS を使用しているユーザ等は Unicode 標準のサロゲートペアをサポートしていないものもあり、フォントをインストールしても表示されない場合がある。また、Windows で Internet Explorer で使用するにはレジストリを書き換える等の作業も必要になる場合がある。ウェブデザインに使用している CSS もブラウザによっては一部こちらが意図した通りに表示されないものもある。

推奨環境

1) サロゲートペアをサポートしていてる OS を使用
2) Unicode 標準の漢字がすべて含まれたフォントをインストール
3) 最新の Mozilla Firefox を使用

ハードウェア（CPU やメモリ）に関しては Mozilla Firefox のシステム要件[2]を満していればよいだろう。

3 利用方法

検索方法としては、「簡易検索」と「詳細検索」の 2 種類を提供している。

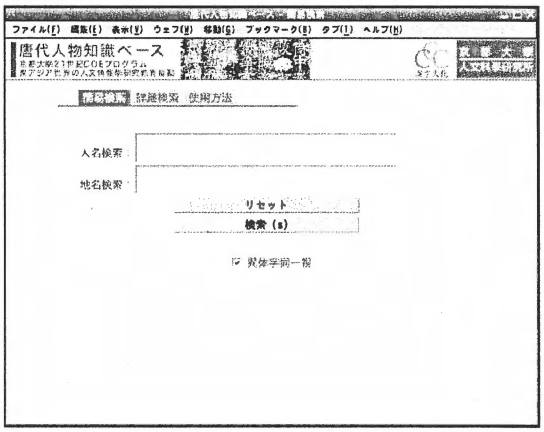
すべての語彙検索に XQuery の正規表現[3] を使った検索が可能である。

また、本システムで採用している異体字を確認するサービスも併せて提供する。

データベースに格納されている人物情報は現在以下の項目についてである。

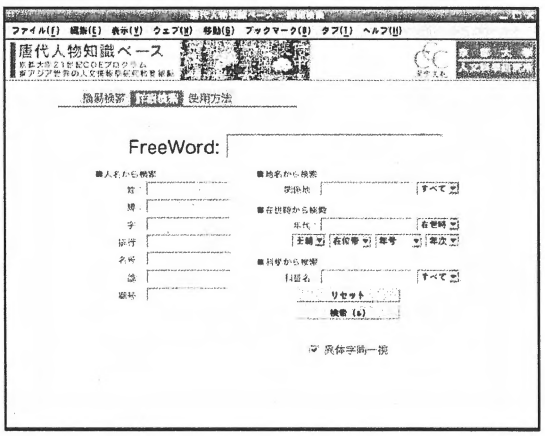
項目：	内容
人名：	姓　諱　字　排行　名号　謚　廟号
関係地：	貫籍　郡望　寓居　生地　卒地
在世時：	在世時　生年　没年
家系：	祖　父　祖先　子孫　家族　姻戚
経歴：	科挙　官職　著作　記事

簡易検索



簡易検索では、人名、関係地、および人名と関係地の AND 検索ができる。例えば、名前に「韓」を含み、関係地に「河陽」を含む人物を探したければ、人名欄に「韓」、地名欄に「河陽」と入力して検索を実行する。検索項目は中国学で需要が高いと思われる人名と関係地からの検索だけに絞った。

詳細検索



詳細検索は「FreeWord」検索と「人名」、「関係地」、「在世時」、「科挙」の詳細項目検索の 2 種類を提供する。

[2] http://www.mozilla-japan.org/products/firefox/system-requirements.html
[3] http://www.w3.org/TR/xpath-functions/#regex-syntax

FreeWord 検索：
人名、関係地、在世時、家系と経歴の科挙、官職から検索する。本システムで最も広範囲を対象とする検索が可能である。

詳細項目検索

名前から検索：
「姓」、「諱」、「字」、「排行」、「名号」、「諡」、「廟号」の検索をする。

地名から検索：
プルダウンメニューの「すべて」、「貫籍」、「郡望」、「寓居」、「生地」、「卒地」から選んで検索する。

在世時から検索：
プルダウンメニューの「在世時」、「生年」、「卒年」から選んで検索する。

科挙から検索：
プルダウンメニューの「すべて」、「登第」、「不第」から選んで検索する。

「人名」、「地名」、「在世時」、「科挙」の詳細項目検索は検索したい複数の項目に語彙を入れれば、複数の項目による AND 検索を行うようにしてある。例えば、姓が「王」、貫籍が「京兆」で西暦「690 年」生れの科挙の「進士」に登第した人物といった検索ができる。

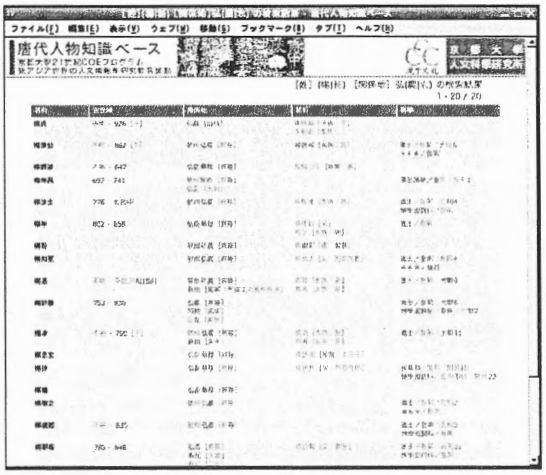
検索機能の補足

現在、本システムは Google などで採用されているような、検索欄に複数の語彙を半角スペースで区切って入力すると AND 検索を行なうような仕様にはなっていない。その代わりに、「| (半角の縦棒)」で複数の語彙を区切ることで OR 検索ができるようになっている。例えば、詳細検索の名前から検索の諡の検索欄に「文|貞|懿」と入力して検索を実行すれば、諡が「文」もしくは「貞」もしくは「懿」を含む人物を探すことができる。この半角の縦棒「|」は正規表現の演算子であり、演算子の左右に置かれた表現のいずれかにマッチを意味する。NOT 検索には現在対応していない。検索方法については、利用者からの意見や要望が多ければ変更を検討する。

詳細検索の「名前から検索」のみ現在実験的に「検索対象項目にデータが入力されてない人物を捜す」ことを可能としている。入力欄に「^$」(これも正規表現演算子であり、「^」は行の先頭、「$」は行の最後の意味となり、「^$」とすることで行の最初と最後、つまり、その行に文字データがないものと解釈される) として検索を実行する。例えば、姓の情報がない人物を探したければ、姓の欄に「^$」と入力して検索を実行するだけである。

詳細検索の「在世時から検索」には西暦と年号からも検索できるようにしてある (ただし、西暦と年号の対応処理は、まだ完全なものではないため、うまく検索できない場合もある)。数字に関する部分はすべて算用数字で入力する (半角・全角の区別はしない)。ただし、「元年」のみ「1 年」と同一視するようにしてある。また、JavaScript 使用の許可をしてあれば、「王朝」、「在位帝」、「年号」、「年次」のプルダウンメニューを使用した入力ができる。この JavaScript の特徴は、プルダウンで選択できる語句を動的にチェックし、存在しない年代表記を生成しないような工夫をしてある (現在、「隋」と「唐」のみ対応)。選択した語句は年代欄に入力される。

検索結果一覧



上図は詳細検索で姓が「楊」で関係地が「弘農」、プルダウンメニューは「すべて」を選択して検索した結果である。図の右上に検索した語彙、そのすぐ下にマッチした件数を表示している。一画面に表示される最大件数は 100 件である。一覧には「名前」、「在世時」、「関係地」、「家系」、「科挙」について表示するようにしてある。「名前」に表示されている名称は、その人物の代表的な表記を見出し人名として表示している。

「名前」列の各人名をクリックすると、その人物の個人情報へ移動する。

「在世時」列は、左側に生年、右側に卒年を表示しており、生年をクリックすれば、同じ年に生れた人物、卒年をクリックすれば、同じ年に亡くなった人物の一覧へ移動する。

「関係地」列は地名をクリックすると、その地名と関連のある人物の一覧へ移動する。

「科挙」列は科目名をクリックすると、その科目名と関連のある人物の一覧へ移動する。

また、一覧テーブルのヘッダ部分をクリックすると並び換えができるようにしてある。現在、動作するのは、「名前」、「在世時」、「関係地」、「家系」の項目である。デフォルトの状態からクリックすると降順、もう一度クリックすると昇順に並び換える。空白データはどちらも一番後ろに配置されるようにしてある。「sort reset」をクリックすれば初期状態に戻る。現在並び換えのルールはUnicode標準の文字コード順で行っている（eXistの仕様でサロゲートペアの文字よりFxxx番台の文字のほうが文字コードが大きくなってしまう）。

個人情報

上図は「韓愈」の個人情報である。まず左上に見出し人名、右上にその人物の異称を並べてある。その下の左側に人物略歴、右側に関係地、著作、記事といった情報を表示している。人物略歴には、生没年、科挙、官職歴をそれぞれ表示している。また、生没年が両方判明している人物は棒グラフで在世期間を視覚的に表示してある。女性に関しては背景色を変えるようにした。科挙の科目名が、データ入力した文献に記載されていない場合は科目名を「＊＊＊」としてあり、「＊＊＊」で検索できるようにしてある。官職の官職名の前に「＊」が付いているものがある。これは入力した文献のみでは正式な官職名が判らない場合を意味している。官職名の前の「（　）」は兼官を意味している。

[4] http://kanji.zinbun.kyoto-u.ac.jp/~yasuoka/CJK.html

4　異体字同一視

本システムではデフォルトで異体字同一視機能が有効になっている。異体字同一視機能を無効にしたい場合はチェックを外してから利用してほしい。

異体字テーブルは安岡氏等[4]によって製作されたものを採用した。ただし、このテーブルはUnicode標準のBMP（基本多言語面）領域に定義されている漢字のみで、完全なものではない。例えば、このテーブルには「曆」の異体字は3つ定義されている。そして、その3つの異体字に対し、以下のような異体字関係になっている。

曆	：	历	歷	曆			
历	：	厤	歷	曆	曆	歷	歷
歷	：	历	曆	曆	歷	歷	
曆	：	历	厤	歷	曆		

上の表を見れば明らかなように、この異体字テーブルは1つの漢字に対し、異体字関係は一方向で定義されていることに注意してほしい。相互形式になっていないため、利用者が入力した文字によって対応する異体字に違いがでる場合がある。要望があれば異体字テーブルは随時更新するつもりである。

検索処理の仕組み

ここでは書道で有名な「欧陽詢」を例に異体字処理の仕組みを簡単に解説する。この場合、「欧」、「陽」、「詢」に対して

欧	：	歐
陽	：	阦　阳
詢	：	询

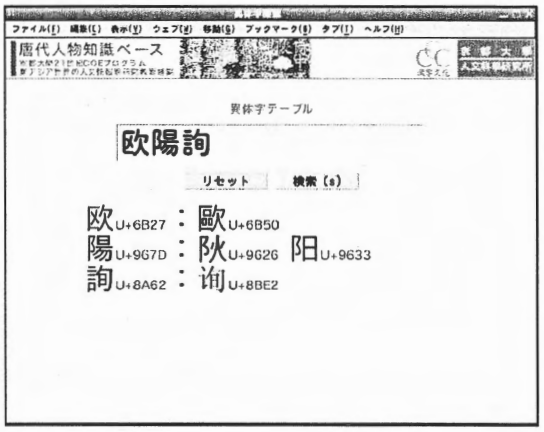
の異体字が対応している。これらを内部で以下の正規表現に置き換えてからデータベースを検索する。

検索処理の流れ

検索文字列	欧陽詢			
▼	▼			
正規表現置換	（欧｜歐）（陽｜阦｜阳）（詢｜询）			
▼	▼			
データベース検索対象文字列	欧陽詢	歐陽詢	欧阦詢	歐阦詢
	欧阳詢	歐阳詢	欧陽询	歐陽询
	欧阦询	歐阦询	欧阳询	歐阳询

この結果、「欧陽詢」に対して2×3×2＝12通りの組み合わせで検索を可能とする仕組みである。

検索結果一覧の検索した語彙の箇所は異体字処理を施した文字列を敢て表示している。これは内部でどのような異体字処理がされているのかを利用者に実際に目で見て確認してもらうためである。「簡易検索」と「詳細検索」にある「異体字同一視」をクリックすると、本システムで採用している異体字を確認できるサイトに移動する。



使い方は単純で、異体字を調べたい文字を入力欄に入力して検索ボタンを押すだけである。

異体字同一視機能は、「全国漢籍データベースの設計とWWWでの運用」(安岡、2002) [10] を参考にして実装を行なった。

5 データベースエンジン

本システムには、eXistというネイティブXMLデータベースエンジンを選択した。eXistを選択した主な理由は

1) ネイティブXMLデータベースエンジンである

2) CJK統合漢字拡張Bが扱える

の2つがある。

本データベースシステムはデータベースエンジンに、世間で利用実績の多いリレーショナル・データベースではなく、利用実績がまだ多いとは言えないネイティブXMLデータベースを敢えて採用した。ネイティブXMLデータベースとは、格納するデータはXMLの文法に従っていればよく、XMLの持つデータ構造の柔軟性を活かしやすく、複雑なデータ構造を持つXML文書でも効率的に扱うことができ、格納できる情報量も多いといった特徴を持っている。この自由度の高さは、システム設計時に格納するデータ構造が決定していなくても開発を開始できるという利点でもあり、リレーショナル・データベースでは実現困難な複雑なデータベースを容易に構築できるのである。現在、本データベースには約4000件の人物データが登録されており、データベースの規模でいえばおそらく小規模になるだろう。また、XML形式にすることで、データを見ただけで視覚的に人物データモデルの構造を的確に把握できる。また、データ交換なども容易に行なえ、人物モデルの仕様の変更およびデータベースシステムの変更などにも柔軟に対応できることなどの理由から本システムにネイティブXMLデータベースを採用した。また、eXistはオープンソースで活発に開発が進められており、内部の仕様、およびソースコードがすべて公開されていることも開発する上での利点である。また、Javaで作られているため、Javaが動作する環境であればOSに依存せずに使用でき、インストールも容易である。

そして、我々にとって最も重要な要素として、Unicode標準 [6] のCJK統合漢字拡張B (CJK Unified Ideographs Extension B、補助漢字面、Supplementary Ideograph Plane、SIPなどと呼ぶ) が扱えることである。本データベースの対象が中国古典資料であるため、Unicode標準の規格で扱える漢字はすべて使用できる状態が望ましいのである。XMLデータベース選びで悩むのは、Unicode標準の基本多言語面の文字だけであるのなら選択肢は広いのだが、CJK統合漢字拡張Bが扱えるとなると途端に選択肢が狭まってしまう。W3CのXMLの仕様 [1] では、CJK統合漢字拡張B領域の漢字も文字として使用できることなっているのだが、対応していない製品もある。

また、これまでに何度か出てきたXQuery [2, 3] とは、リレーショナル・データベースにおける標準の問い合わせ言語がSQLであるように、XMLデータベースにおける標準の問い合わせ言語がXQueryである。XML文書に対してさまざまな問合せを行うことが出来るように開発された言語であり、W3Cで仕様の確定が進められている。XQueryでは問い合わせの結果として、XMLを作成する。問い合わせの仕方によっては複雑なXMLを出力することも容易である。また、XQueryはXMLデータベース用の問い合わせ言語としてだけではなく、XML形式なテキストファイルに対して処理が行なえるXQuery処理器がある。本システムではSaxon-B [8] を使用し、eXistに格納するためのデータの加工や結合といった作業に利用している。

システム構成

本データベースシステムは以下の構成で運用している（主要なもののみ提示）。

サーバー：	Dell PowerEdge 2800
CPU：	Intel Xeon 3.80GHz × 2
メモリ：	8GB (4 × 2G 2R DDR2/400)
OS：	Red Hat Enterprise Linux ES v.4
Java：	Sun Java 1.5.0_04_b05
DB：	eXist 1.0rc

現在、eXist は 1.0 系と 1.1 系の 2 種類あり、それぞれ最新バージョン（2006 年 11 月 23 日現在）は 1.0.1 と 1.1.1 であるが、本システムでは 1.0rc 版を使用している。最新版には本データベースの実装に対して我々が意図した通りに動作しないバグがあるため最新版を使用できない状況である。開発者には既に報告してあるので次のバージョンで改善されていれば最新版に変更する。開発者達とのやり取りが気軽にできることもオープンソース開発の利点である（筆者の個人的な感想だが、eXist のメーリングリストにはかなり好感を持っている）。

6 データ構造および問い合わせの最適化

実際、どんなネイティブ XML データベースでも「複雑なデータ構造を持つ XML 文書でも効率的に扱うことができる」という、このデータベース最大の長所が実現されているのだろうか？ 我々が採用した eXist に関していえば、データベースに格納してあるデータの構造、いわゆる物理データモデルに依って、問い合わせなどの処理が十分な時間で処理できない場合がある。検索サービスを提供する上で結果が表示されるまでの時間を極限まで短縮させること（検索処理を軽くすること）は利用者に対して一番重要な事項であるため、論理データモデルからの大幅な構造変更も必要となる。XML 文書をそのまま扱えることがネイティブ XML データベースの魅力であるはずなのだが、物理データモデル、および XQuery 問い合わせをどのように設計すれば効率的な処理ができるのかは、いろいろ実験を繰替えしながら見極めていくしかないのが現状である。今後、eXist の性能が向上していけば、このあたりの設計の手間は軽減するだろうと期待している。しかしながら、ハードウェアの性能などから、満足な処理速度が得られない場合などは、限界まで物理データモデル、および XQuery 問い合わせの設計を吟味する必要があるだろう。これは他のネイティブ XML データベースでも同様である。

データ構造の最適化

ここでは、人名に関する検索を例にして最適化を具体的に見てみよう。まず、人物データモデル（論理データモデル）[11] では以下のような XML で人名部分は構成されている。

a) 人物データモデル

```
<個人 id="7-336">
  <人名>
    <姓 value="韓">
      <諱 value="愈"/>
      <字 value="退之"/>
      <排行 value="十八"/>
      <諡 value="文"/>
      <名号 value="昌黎"/>
      <名号 value="吏部"/>
      <名号 value="文公"/>
    </姓>
  </人名>
</個人>
```

このモデルの特徴は、<姓>とそれに付随する<諱>や<字>等の関係を明確にしている点である。しかし、「諱」や「字」は「value」属性の値としてそれぞれ格納されている。この状態のままでは、簡易検索の人名検索で「姓＋諱」の「韓愈」と入力してマッチするように実装するとなると属性値同士を結合したりしなければならず無駄が多くなる。従って各種検索を効率的に行なえるような XML にモデルを変換する必要がある。簡易検索用に最適化したデータは以下の形式である。

b) 簡易検索用に最適化したデータ

```
<個人 id="7-336">
  <人名>
    <名前 type="姓">韓</名前>
    <名前 type="姓＋諱">韓愈</名前>
    <名前 type="姓＋字">韓退之</名前>
    <名前 type="姓＋排行">韓十八</名前>
    <名前 type="姓＋諡">韓文</名前>
    <名前 type="姓＋名号">韓昌黎</名前>
    <名前 type="姓＋名号">韓吏部</名前>
    <名前 type="姓＋名号">韓文公</名前>
  </人名>
</個人>
```

このように、最初から<姓>とそれに付随する<諱>や<字>を結合した形式にしておけば、「姓＋諱」の「韓愈」と入力してマッチするように実装するのが容易になる。

逆に、詳細検索の場合は、「姓」のみ、「諱」のみ、「字」のみといった名前の部品毎で検索できなければならないため、この形式のままでは困るのである。それに対しては次のようなものを用意する。

c) 詳細検索用に最適化したデータ

```
<個人 id="7-336">
  <人名>
    <名前 type="姓">韓</名前>
    <名前 type="諱">愈</名前>
    <名前 type="字">退之</名前>
    <名前 type="排行">十八</名前>
    <名前 type="諡">文</名前>
    <名前 type="名号">昌黎</名前>
    <名前 type="名号">吏部</名前>
    <名前 type="名号">文公</名前>
  </人名>
</個人>
```

このようにして、<諱>や<字>を<姓>の子要素にせずに、並列化することで、それぞれの名前の部品を検索するための実装が容易になる。

このように、検索に特化したデータ構造を持つXMLファイルを個別に用意することで検索処理の効率を上げることが容易にできるのである。

問い合わせの最適化

XQueryによる問い合わせ結果が同じであっても、記述方法の違いによって結果が返ってくるまでの時間に大きな差が生じる場合や、使用するデータベースエンジンによって同じ問い合わせであっても処理時間に差が生じるような場合がある。ここでは、本システムの「在世時から検索」のプルダウンメニューで「在世時」を選択した場合の実装を例に具体的に見てみよう。

「在世時」は生きていた時間帯であるため、幅を持っている。現在は入力データの「生年」と「卒年」がそれぞれ「西暦年」表記で入っている場合のみ、変換テーブルを利用して「生年」と「卒年」の間の可能な「年号＋年次」表記をすべて<在世>要素の下に格納している。従って、長生きした人物は必然的にデータ量が増すことになる。紙面の都合から、以下のサンプルデータは実際のデータベースに格納しているデータから検索時に使用しない部分を省略したものである。

「在世時から検索」に使用するデータの構造

```
<個人 id="7-336">
  <在世>
    <時>768 - 825</時>
    <時>唐代宗大暦 2 年</時>
    <時>768 年</時>
          ⋮
    <時>唐敬宗寶暦元年</時>
    <時>唐敬宗寶暦 1 年</時>
    <時>貞元 8</時>
    <時>貞元 12 年 7 月</時>
          ⋮
    <時>長慶 3</時>
    <時>長慶 3 年 10 月</時>
  </在世>
</個人>
```

「在世時から検索」でプルダウンメニューで「在世時」を選択した場合は上記の構造の<時>要素から検索する。欲しい結果はマッチした<時>要素の値ではなく、<個人>要素であるため、以下のようなXQuery問い合わせで目的は達っせられる。

```
個人 [在世/時 [matches(., $txq)]]

$txq は検索語彙を代入したもの
```

しかしながら、eXistではこの記述方法だと検索にマッチする<時>要素が多くなる（例えば検索欄に「.（ピリオド）」を入力）につれて処理が重くなる傾向がある。これまで、処理速度が大幅に遅くなるわけではなかったため、eXistの仕様であるとしてあまり気にしていなかったのだが、試しに以下のように記述してみたところ、上述のような傾向が見事に解消した。もちろん、問い合わせ結果は同じである。

```
個人 [在世 [時 [matches(., $txq)]]]
```

両者の違いは、「在世/時」と「在世 [時 …]」だけである。ところが、Saxon-Bで両者を比較しても差が現れない。つまり、これはeXist固有の問題になるわけである。言いたいことは、問い合わせ結果が同じであっても、データベースエンジンや記述方法の違いによって処理速度に大きな差が出る場合がある。シンプルな記述より、複雑なほうが処理が軽くなるといったこともあるかもしれない。実装する際には、いろいろな記述を試み、比較検討することが非常に重要なのである。

7 今後の予定

京都大学 21 世紀 COE「東アジア世界の人文情報學研究教育據點」は 2007 年度でプロジェクトは終了する。我々が現在開発を行っている「唐代人物知識ベース」は、平行して行なわれている「唐代官職知識ベース」、「唐代地理知識ベース」と統合され、「唐代知識ベース」として最終的に公開される。主要な開発は 2007 年の 8 月くらいで終了し、後はメンテナンス程度になる予定である。従って、統合された「唐代知識ベース」の実装は残された時間で実現可能なものになるだろう。現在は、まだ統合された「唐代知識ベース」をどのようなデータベースサービスにすべきか議論している段階であり、具体的なデータベースの設計等はこれからである。現段階で決定していることは、「唐代知識ベース」の実装は筆者が担当することだけである。つまり、筆者の能力次第で「唐代知識ベース」がどこまで出来上がるかが決ってしまうことである。特に「漢字文献ナリッジベース構築」部門は本 21 世紀 COE のメインであるため、順調にプロジェクトが遂行していくことを願うばかりである。

謝 辞

本プロジェクトは、独立行政法人日本学術振興会の 21 世紀 COE プログラムの援助を受けて行なわれている。そして、本データベースシステムの構築には、データモデルの策定、データ入力、データベースの設計等に本プロジェクトに関っている多くの方々からの協力を得て成り立っている。ここに心から感謝の意を表す。

参考文献

[1] Extensible Markup Language (XML) 1.0 (Third Edition): http://www.w3.org/TR/2004/REC-xml-20040204/

[2] XQuery 1.0: An XML Query Language: http://www.w3.org/TR/xquery/

[3] XQuery 1.0 and XPath 2.0 Functions and Operators: http://www.w3.org/TR/xpath-functions/

[4] XHTML 1.1 - Module-based XHTML: http://www.w3.org/TR/xhtml11/

[5] Cascading Style Sheets, level 2 revision 1 CSS 2.1 Specification: http://www.w3.org/TR/CSS21/

[6] Unicode 4.1.0: http://www.unicode.org/versions/Unicode4.1.0/

[7] eXist：http://exist-db.org/

[8] Saxon-B：http://www.saxonica.com/

[9] 周 祖撰 主編：「中国文学家大辞典 唐五代巻」、中華書局、1992

[10] 安岡 孝一：「全国漢籍データベースの設計と WWW での運用」、平成 14 年度全国文献・情報センター人文社会科学学術情報セミナー『データベースの活用と人文社会科学』、2002

[11] 秋山 陽一郎、白須 裕之、永田 知之：「中国古典学知識ベースにおける信頼性評価モデルの一試案」、第 17 回東洋学へのコンピュータ利用、2006

[12] 山本 一登：「唐代人物知識ベースの実装 ～ eXist による試み ～」、第 17 回東洋学へのコンピュータ利用、2006

付 録

Unicode 標準 4.1 で定義されている漢字がすべて含まれたフリーなフォントに「HAN NOM」がある。フォントの作成過程等の詳細はわからないが、実用に耐えられる品質を持ち、且つマルチプラットフォームで利用できるフリーなフォントは「HAN NOM」しかないのが現状だ。このフォントは http://vietunicode.sourceforge.net/fonts/fonts_hannom.html か http://prdownloads.sourceforge.net/vietunicode/ からダウンロードできる。hannomH.zip をダウンロードし、解凍すると HAN NOM A.ttf と HAN NOM B.ttf という 2 種類のフォントが入っている（説明書は同封されておらず、フォントしか入っていない）。TrueType フォントは 1 つのフォントに含むことのできるグリフ数の最大値に制限があるらしく、1 つのフォントでは収まりきらず、2 つに分割されている。HAN NOM A には Unicode 標準の BMP（基本多言語面）領域の漢字、HAN NOM B にはサロゲートペア領域の漢字がそれぞれ収められている。もちろん、漢字以外のグリフもかなり収められている。フォントが 2 つに別れているため、同時に 1 種類のフォントしか表示できないようなアプリケーションでは、画面に表示した際に文字化けする場合もあるだろう。HAN NOM A.ttf には一部コードポイントがずれているバグがある。具体的には FA4C～FA6A のグリフが 1 つずつ後ろにずれていて、FA4B のグリフが FA4C に重複している。幸いこの領域の漢字は CJK 互換漢字と呼ばれていて、使用しないほうがよいと言われているため、さほど気にする問題ではない。