

日付を表す文字列の解釈と暦の変換 — 暦に関する統合基盤の構築に向けて

関野 樹
総合地球環境学研究所

山田 太造
東京大学史料編纂所

暦は時間に基づく情報解析や情報連携で必須の要素であり、多様な暦を扱うためには暦同士の変換をはじめとする暦に関する知識を集約した統合基盤が必要である。本研究は、この統合基盤の主要な機能となる日付文字列の解釈と暦変換について、和暦、ユリウス/グレゴリオ暦（西暦）およびユリウス通日を題材に実装を試みた。さらに、これらの機能を利用した暦変換のための Web フォームを構築・公開し、その機能を検証した。その結果、日付文字列を解釈する機能が加わることにより、入力作業の省力化や複数のデータを一度に変換するなどの点で暦変換の操作が大幅に効率化することが確かめられた。今後 API を介してこれらの機能を利用する仕組みを設けることで、ソフトウェアや Web ページに組み込んでの利用やテキスト処理への応用など、暦の統合基盤の利用が広がると考えられる。

Interpretation of character strings expressing date and conversion of calendar – Toward construction of integrated base of calendars

Tatsuki Sekino
Research Institute for
Humanity and Nature

Taizo Yamada
Historiographical Institute,
University of Tokyo

Integrated base to accumulate knowledge and functions about calendars is required in studies using various types of calendar. The main functions of the integrated base are functions to interpret date expression and to convert calendar, and were implemented for Japanese calendar, Julian/Gregorian calendar and Julian Day in the present study. Additionally, a web form using these functions was opened, and the functions were examined using the web form. As a result, the function of interpretation enables to input plurality of data, and makes the calendar conversion more efficient. When API to use these functions will be developed, it will be possible to apply the functions to software, web pages and text processing, and then usage of the integrated base of calendars will be extended.

1. はじめに

暦の変換は、時間に基づく情報解析や情報連携の前提となる処理であり、異なる暦を相互に変換するための多くのサービスが Web 上でも公開されている。しかしながら、その多くは年月日を1つずつテキストボックスに入力してその結果を表示するものであり、一度に多くのデータを変換することができない(図1)。また、元の資料(史料)の記述から年月日を読み取ることも利用者自身の知識に頼ることになり、漢字が読めない海外の研究者には和暦の日付を年月日それぞれのテキストボックスに入力することも容易ではない。もし日付を表す文字列を自動的に解釈することができれば、日付とおぼしき部分をコピー&ペーストすればよいだけなので、入力の負担を減らすことができる。さらに、図1のように複数のデータを一度に入れることもできるので、作業効率も大幅に改善される。

入力だけでなく、暦の変換そのものにも課題がある。異なる暦であればもちろんのこと、同じ暦であっても地域や時代によってその運用が異なることがしばしばある。このため、地域や時代と

結び付けた形で複数の暦に関する知識を集約し、それらに基づいて適切な暦を選択する仕組みが必要である。しかしながら、こういった機能は人文科学の様々な分野で必要であるにも関わらず、誰もが自由に使える十分な環境が整備されているとは言い難い。

年月日を分けて 入力する場合	平成 <input type="text" value="25"/> 年 <input type="text" value="12"/> 月 <input type="text" value="13"/> 日
日付文字列をそのまま 入力する場合	平成 25/12/13 平成二五年一二月一三日 平成廿五年師走癸丑

図1 日付の入力方法の比較。
Figure 1 Comparison of ways to input date expression.

本研究は、この日付文字列の解釈と複数の暦を相互に変換する仕組みを中心に、それぞれの暦に関する知識やこれらの機能を他のソフトウェアや Web ページで利用するためのインタフェース

を備えた暦に関する統合基盤の構築を目指そうとするものである(図2)。本報告では、統合基盤の中心となる2つの機能、つまり、日付を表す文字列を解釈して変換に必要となる年、月、日を読み取る機能と暦を相互に変換する機能について、和暦とユリウス/グレゴリオ暦(西暦)およびユリウス通日を題材に実装および活用例を紹介する。

なお本研究では、暦の統合基盤の実装をMicrosoft社製の.NET Framework環境を用いて

行った。これは、広く利用されている同社製の表計算ソフトExcelで、ユーザ関数やマクロ機能などとして開発した機能を利用することを想定しているためである。また、実装したクラス等の仕様も、基本的に.NET Frameworkが持つDateTime構造体やCalendarクラスなどの構造や動作を踏襲している。これは、既存のプログラムからの移行を容易にするための措置である。

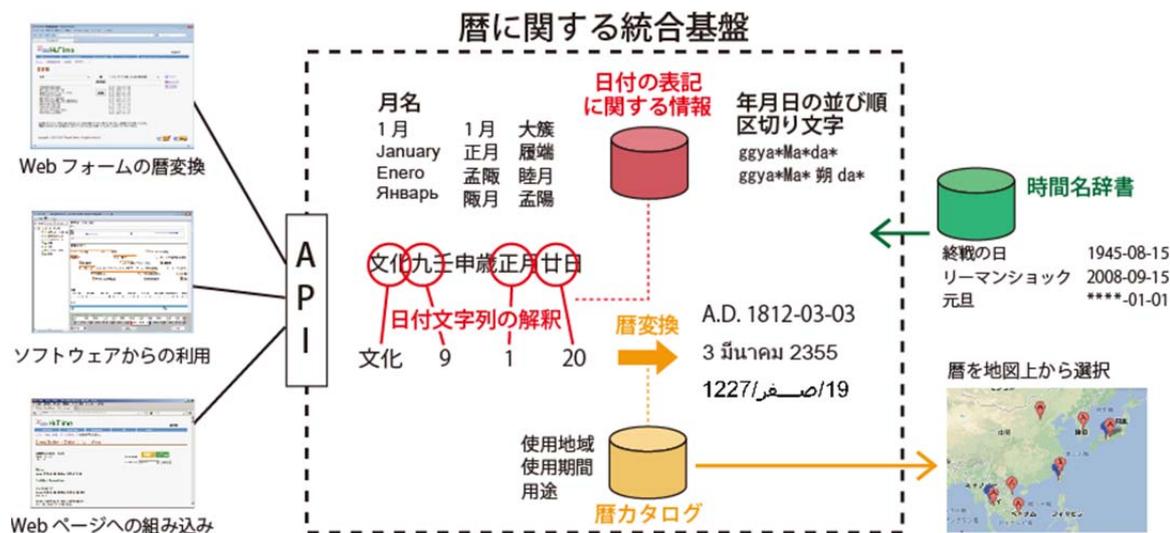


図2 暦に関する統合基盤の概要とその用途。

Figure 2 Outline and usage of the integrated base of calendars.

2. 日付文字列の解釈

入力された日付文字列は、それぞれの暦に応じて解釈され、年号、年、月、日に分けられる。和暦の場合、まず旧字や異体字などの処理が行われる(図3)。漢数字の大字(壺、弍など)もこの過程で一般的な漢数字に置き換えられる。これら旧字、異体字は、東京大学史料編纂所のデータベースで用いられている異体字をまとめた「史料編纂所データベース異体字同定一覧変換一覧」に基づいた[1]。

次に、利用者が指定した書式、もしくは既定の書式に従って、日付が解釈される。書式では、年号、年、月、日が出現する順番、それらの表現形式、区切り文字などのその他の文字が指定されており、入力された日付文字列はその書式に従っているものとして解釈される。日付の書式の指定は、多くのソフトウェアが採用している書式指定子を用いる方法を採用した。たとえば、和暦の平成25年12月13日は“ggyy年MM月dd日”でマッチする。日付表記の標準規格として広く用いられているISO 8601形式に従っている場合、書式はユリウス/グレゴリオ暦で“yyyy-MM-dd”と書式指定をすることになり、2013-12-13がマッチする。このように、yyyy(4ケタの年)やdd(2ケ

タの日)など多くの暦に共通する基本の指定子を設ける一方で、特定の暦だけで用いる指定子が暦ごとに設けられた。和暦での例をあげると、漢数字や干支による表現に対応するための和暦固有の指定子が複数定義されている。たとえば、日ではd:算用数字、dK:漢数字、dE:干支、となる。漢数字はdKで二〇、二十、廿などいくつかのバリエーションをサポートする。

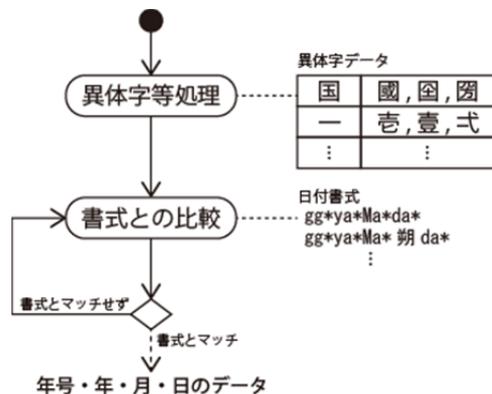


図3 和暦の日付解釈の手順。

Figure 3 Procedure to interpret date expression in Japanese calendar.

さらに、書式の指定にワイルドカードを用いることで、より柔軟な日付文字列の解釈を可能にした。たとえば、和暦の指定子 `da` は、日が算用数字、漢数字、干支のいずれかで表現されていることを示しており、与えられた日付文字列がこれらのいずれかで表現されていれば解釈することができる。つまり、`da` が日を表す文字列として指定された場合、平成 25 年 12 月 13 日の日の部分は、13, 13, 一三, 十三, 癸丑のいずれが入力されても正しく解釈する。

また、ワイルドカードとして、“*” (0 個以上の何らかの文字) や “?” (1 個のなんらの文字) を指定する機能を設け、年月日の区切り文字などについて柔軟な解釈を可能にした。これにより、“`gg*ya*Ma*da*`” と書式が指定されれば、平成 25 年 12 月 13 日、平成 25/12/13、平成 25-12-13 いずれも正しく解釈できる。

これらの指定子や任意の文字を組み合わせて日付文字列の解釈に用いる書式を指定する。書式は候補となるものを複数指定することが可能で、それらを順番に使って日付文字列の解釈を行い、最初に解釈が成功したものが採用される。暦変換には、年号、年、月、日それぞれの指定子にマッチしたデータが用いられる (図 3)。書式を複数指定する仕組みや基本的な書式指定の方法は、互換性の維持や移行を容易にするため、開発環境の .NET Framework 環境の仕様を踏襲している。一方、暦ごとに固有の書式を指定する仕組みや書式指定でのワイルドカードの利用は新たに拡張されたもので、これにより解釈可能な日付文字列の幅が大幅に広がっている。

3. 暦変換機能

複数の暦を相互に変換するには共通の時間軸となる暦を決める必要がある。ここでは、天文学などでも時間軸として用いられるユリウス通日を共通軸とした。ちなみに、ユリウス暦とグレゴリオ暦、いわゆる西暦は、ユリウス暦からグレゴリオ暦へ改暦した際に 10 日以上の日付の不連続が生じること、また、改暦された日が地域によって異なり一貫性が保てないことから共通の軸には適さない。ユリウス通日は、紀元前 4713 年 1 月 1 日正午からの通算日数であり不連続がない。また、時刻は小数として示される。ちなみに、平成 25 年 12 月 13 日 16 時 50 分であれば、2456640.20139 となる (時差を考慮しない場合)。今回は和暦 (神武天皇元年 1 月 1 日から平成 94 年 12 月 31 日、南朝・北朝)、ユリウス/グレゴリオ暦 (紀元前 9999 年 1 月 1 日から 9999 年 12 月 31 日、改暦日は 1582 年 10 月 15 日 (イタリアなど) と 1752 年 9 月 14 日 (イギリスなど) の 2 通り) について、共通のユリウス通日と相互に変換する仕組みを構築した。

暦は計算のみによって理論値を求めることも可能であるが、実際にはさまざまな事情で理論値

とは異なる日付を使うことが少なくない。たとえば和暦では、元日の日食を嫌い、その前の月の日数を故意に操作して元日に日食とならないようにするなどした [2]。史料などを用いる研究の現場ではこうした実際に使われた日付が必要であり、変換結果もこれに対応させる必要がある。このため、変換機能は大きく 2 通りの方法を採用した。1 つは計算によってのみ変換する方法で、ユリウス/グレゴリオ暦とユリウス通日の間の変換はこれによっている [3][4]。なお、ユリウス暦からグレゴリオ暦の改暦日は変換を実行する前に指定する (厳密にはクラスのインスタンス生成時)。

もう 1 つはデータベースを併用する方法で日付の振り方 (閏月の置き方、月内の日数など) が不規則に変わる場合、日付の振り方に人為的な要素が関与する場合、元号がある場合などが相当する。今回は和暦とユリウス通日の間の変換でデータベースを併用する方法を採用した。具体的には、実際に使われた日付に関する研究成果 [2][5] や法令 [6][7] に基づいて、各月の時間範囲とユリウス通日を対応させた表 (月テーブル) を用意し、これを使って日付の変換を行う。このテーブルのデータは、神武天皇元年 1 月 1 日から允恭天皇 33 年 12 月 29 日までは『日本書紀暦日原典』 [5] に、允恭天皇 34 年 1 月 1 日から明治 5 年 12 月 2 日までは『日本暦日原典』 [2] に、明治 6 年 1 月 1 日以降は法令 [6][7] に従っている。月テーブルのレコード数は全体で 33836 件となった。

和暦の場合は年号についても管理が必要となる。各年号の時間範囲とユリウス通日を対応させた表 (年号テーブル) を作成し、これを使って年号を特定する。データは、「神武天皇」から「文武天皇」までは『日本書紀』および『続日本紀』 (ともに『日本国史大系』収録のもの) [8][9] の記述に、「大宝」以降については、原則として『国史大辞典』 [10] の記述に従った。年号が無い期間は天皇 (または皇后) の名称を年号の代わりに用いている。南朝、北朝については、両方の年号を収容し、変換操作の前にどちらが優先されるかを指定することとした。

さらに、和暦については年に一貫した番号が付されていると計算がしやすいことから、神武天皇即位紀元年 (いわゆる皇紀) を上記 2 つのテーブルのそれぞれのレコードに付した。

ユリウス通日から和暦を得る具体的な手順は下記のとおりとなる。

- (1) ユリウス通日 (JD) が与えられると、月テーブルから該当する月の月名 (M)、皇紀年 (Ym) および朔日のユリウス通日 (JDm) を得る。
- (2) 同様に、JD に基づいて年号テーブルから該当する年号 (gg) とその年号が始まった皇紀年 (Ye) を得る。

(3) Y_m と Y_e から、求める日付がその年号の何年目 (Y) かを得る。

$$Y = Y_m - Y_e + 1$$

(4) JD と JD_m から求める日付のその月内の日 (D) を得る。

$$D = JD - JD_m + 1$$

結果、gg Y 年 M 月 D 日 を得る。

反対に、和暦からユリウス通日を得る場合は、下記のとおりとなる。

(1) gg Y 年 M 月 D 日が与えられると、年号テーブルから gg の始まった皇紀年 (Y_e) を得る。

(2) Y と Y_e から、求める日付の皇紀年 (Y_m) を得る。

$$Y_m = Y_e + Y - 1$$

(3) Y_m と M に基づいて、月テーブルから該当する月の朔日のユリウス通日 (JD_m) を得る。

(4) JD_m と D から、求めるユリウス通日 (JD) を得る。

$$JD = JD_m + D - 1$$

この方法では該当する年号の初年からの年数を皇紀年に置き換えて数えるだけで、年号の取りうる範囲はチェックしていない。このため、昭和 70 年などの実在しない年にも対応できる。このような実在しない年は地理的もしくは政治的な要因で改元が正しく行われなかった場合に史料中に出現することがしばしばあり、これらの解釈にも本機能が有用である。

4. 出力される日付文字列の書式指定

変換結果として出力される日付文字列も書式を指定することが可能である。一部利用できない指定子があるものの、指定方法は基本的に入力時と同じである。たとえば、和暦で“ggY (yW) 年 MMM 月 d 日”と指定した場合は、歴史研究等の資料でよく見られる西暦を併記した表現、

元禄 15 (1702) 年 12 月 14 日

といった出力結果を得る。なお、ここで示す西暦は和暦の年に対応させた年で、実際の西暦とは異なる (元禄 15 年 12 月 14 日はグレゴリオ暦では 1703 年 1 月 30 日)。和暦の漢数字については指定子を使い分けることで細かな指定が可能である。例えば 20 の場合、二〇、二十、廿を使い分けることができる。また、1 日を朔に、月の終わりを晦に置き換えることなどもできる。具体的な指定方法として日の場合を例にとると、

dK1 一, ..., 一〇, 一一, ..., 二〇, 二一, ..., 三〇, 三一

dK1b 朔, ..., 一〇, 一一, ..., 二〇, 二一, ..., 晦

dK2 一, ..., 十, 十一, ..., 二十, 二十一, ..., 三〇, 三一

dK3 一, ..., 十, 十一, ..., 廿, 廿一, ..., 卅, 卅一

となる。ユリウス/グレゴリオ暦 (西暦) では、紀元前の表記について実際の利用に即した出力形式を備えた。

y ..., 2, 1, 0, -1, -2, ...

ggY ..., A.D.2, A.D.1, B.C.1, B.C.2, B.C.3, ...

これにより、“-1 年”は紀元前 1 年を示すのか紀元前 2 年を示すのかといった混乱を避けることができる。

5. 実装

日付の演算などの基本機能を担うモジュールとそれぞれの暦に関する処理を行うモジュールが作成された (図 4)。各暦のモジュールには、基準となるユリウス通日と相互に日付を変換する機能、および、その暦で用いられる表記に応じて日付文字列を解釈する機能が実装されている。各暦のモジュールは和暦のようにデータベースを伴っている場合もあるなど仕様はそれぞれ異なるが、基本機能が持つ共通のインタフェースを通じて操作されるため、暦に固有な設定 (和暦の南朝・北朝の別、西暦の改暦日など) が最初に必要である以外は同じ操作で日付文字列の解釈や変換を実施できる。この共通のインタフェースを利用することで新たな暦に対応するモジュールを追加することも容易である。今回は日付文字列の解釈と暦変換の 2 つの基本機能の実装と検証に目的を絞ったため、暦カタログや API は作成しなかった。今後、対応する暦や利用するサービスの種類が増えた場合には実装が必要になると予想される。

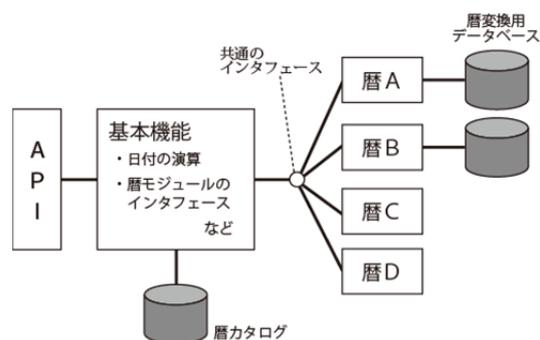


図 4 各機能の実装。

Figure 4 Implementation of each function.

6. Web フォームとしてのサービス提供

構築した統合基盤の機能を検証するとともに、成果を一般に供するため、これらの機能を使った Web フォーム構築した (図 5)。和暦については

日付文字列の解釈のための既定の書式を年号、年、月、日の順番で何らかの表現が並んでいる“gg*ya*Ma*da”と日本書紀などで用いられる日付に対応する“gg*ya*Ma*朔da*”とし、利用者が書式を指定しない場合は、これらの書式で日付を解釈する。以下は既定の書式による変換例である。

天安元年二月丙申 → A.D. 0857-03-27
 文化九壬申歳正月廿日 → A.D. 1812-03-03
 推古天皇元年春正月壬寅朔丙辰
 → A.D. 0593-02-21

利用者が書式を指定した場合は、その書式に従って日付文字列の解釈や結果の出力が行われる。

入力されるデータは改行で区切ることにより複数の日付を一度に変換する仕組みとした。これは、テキストボックスに年月日をそれぞれ入力する従来の方法では実現できなかったものである。

今回構築した Web フォームはすでに公開されており、各暦の指定子などの詳細な情報も当該サイトから入手できる。利用者からは、表計算ソフトから複数行のデータを直接コピー&ペーストで変換するといった使い方や、変換元と変換先に同じ暦を指定することで、バラバラな日付の表記を統一することに使えるなどの新たな利用法も提案されている。

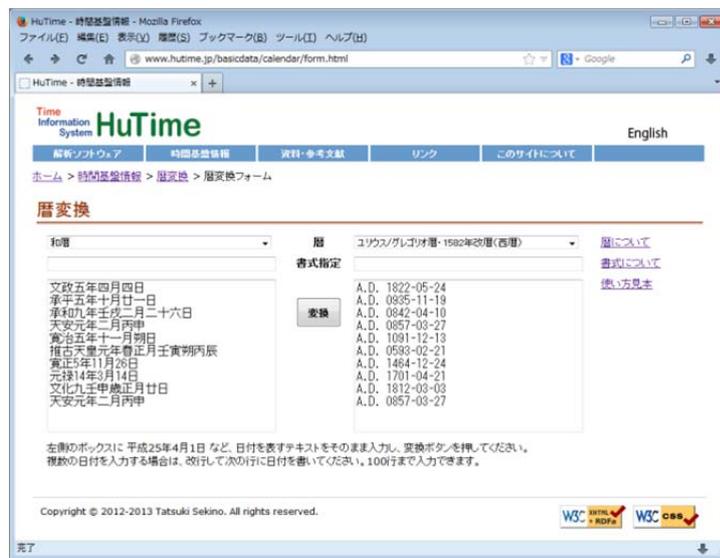


図 5 暦の変換機能の Web フォームへの実装

(<http://www.hutime.jp/>)

Figure 5 The Web form for calendar conversion.

7. 検証

日付文字列の解釈の正確さを検証するため、東京大学史料編纂所のデータベースに含まれる年号、年、月、日の表現を集めてそれらを使った日付文字列を作成し、これを構築した Web フォームに入力することで解釈の成否を確認した。年号については、私年号や省略された表現を除き多くが正しく解釈された (正解率 317/333、以下同じ)。年については、算用数字だけでなく、漢数字、漢数字の大字についても正しく解釈し、解釈できないのは“初年”などの特殊な例だけであった (1165/1167)。月については漢数字、算用数字とも閏月を含めて解釈できるものの、固有の月名 (孟陬=1 月 など) には対応していないため解釈できた割合は下がった (214/395)。これらについては対応を検討中である。日については“朔日”、“晦日”などを含むほとんどのケースで正しく解釈した (363/372)。これらの結果から、少なくとも和暦についてはかなりの数の資料 (史料) で本システムが活用できると考えられる。

なお、今回構築した Web フォームでは、和暦の既定の書式で年月日の区切り文字をワイルドカードで指定していることから、日付以外の文字列が入力されても日付として解釈してしまう可能性がある。たとえば“平成元氣七転八倒”と入力した場合、氣、転、倒はそれぞれワイルドカードで指定されている文字なので、正しい日付 (平成 1 年 7 月 8 日) として解釈してしまう。ただし、暦変換用の Web フォームに利用者が日付以外の文字列をわざわざ入力することは考えにくいいため、実用上このような誤解釈は大きな問題にはならないと考えられる。解釈が正しく行われているかが不確かな場合は、入力と出力の暦を同じにすることで確認することができる。

次に処理速度を確認するため、同じデータ“平成二五年八月二八日”を複数件入力し、ユリウス/グレゴリオ暦への変換を行った (ネットワーク遅延の影響を抑えるため、サーバと同一の LAN で実施)。この結果、3000 件までであれば 1 秒以内に結果が返されることが分かった。ネットワーク環境や入力するデータ、書式指定の複雑さにもよ

るが、現状のシステム構成で十分実用に耐える処理速度であると考えられる。

8. 今後の課題

時間に特化して可視化や解析を行う HuTime [11]のようなソフトウェアが登場し、年表やグラフなどの時間情報を伴うデータを扱う機会が増えてきた。また、デジタルアーカイブの名のもとさまざまな資料（史料）が情報システムの中で扱われるようになり、それら同士、さらには自然科学を含む異なる分野のデータとも連携させる機会が増えてきている。この点でも時間情報は情報資源同士をつなぐ重要な接点であり、暦の統合基盤の役割は大きい。

今回は暦の統合基盤の主要な機能である日付文字列の解釈と暦変換の各機能を、和暦、ユリウス/グレゴリオ暦およびユリウス通日を題材に実装し、実用に耐えるものであるか検証された。今後、これら以外の暦についても需要が生じてくるであろう。また、和暦だけでも私年号や地方で採用された独自の暦への対応にも需要があると見込まれる。実際、タイ仏暦、ヒジュラ暦（イスラム暦）、ユダヤ暦の3つについては試験的な実装を行っており、日付文字列の解釈などについて検討を進めているところである（公開中の Web フォームからも試用可能）。対応すべき暦が増えてくれば、暦変換の機能だけでなく、日付文字列を解釈するための情報、つまり、それらがどのような文字や書式で表現されるのかを検討する必要がある。さらに、暦に関するメタデータ、つまり、ある暦がどの地域でいつの時代に使われたのかを集約した暦のカタログが必要になってくる。こういった情報をどのように収集するかも含めて今後の課題である。

今回は Web フォームから暦の変換を行うサービスとして構築した各機能を直接利用したが、Web API を実装することで、ほかの Web ページやアプリケーションに組み込むことなど、さまざまな形で利用が可能になる。たとえば、Web API を介して HuTime などの時間情報を扱うソフトウェアから直接暦の変換を実施することなどが考えられる。さらに、RDF を出力する機能をサポートすることで、Linked Data の仕組みを介して時間を使った情報資源の連携に寄与することもできると考えられる。

また、構築された日付文字列の解釈機能をテキスト解析などのソフトウェアに応用すれば、文書の中から日付表現だけを抜き出したり、抜き出した日付を別の暦に置き換えたりすることなどが可能になるだろう。実際、地名についてはこのような機能が実現しており、サービスの提供が始ま

っている[12]。テキストへの応用という点では、別途構築を進めている時間名辞書（「終戦の日」と1945年8月15日に関連付ける仕組みで、地名辞書の時間版）[13]と連携することで、歴史上のイベントの名称、さらには「彼岸」や「盆」といった日付を表す一般的な用語を使ってテキストからの日付の抽出を効率よく進めることも期待される。

今回構築された日付文字列の解釈と暦変換を軸に暦に関する統合基盤を発展的に運用することで、時間情報を使った研究がより充実してゆくと期待される。

謝辞 和暦のためのデータ作成には国文学研究資料館の相田満准教授にご協力いただいた。謹んで感謝の意を表す。

参考文献

- 1) 東京大学史料編纂所：史料編纂所データベース異体字同定一覧，入手先
〈http://wwwap.hi.u-tokyo.ac.jp/ships/itaiji_list.jsp〉（参照 2013-10-21）。
- 2) 内田正男：日本暦日原典 第四版，雄山閣（1994）。
- 3) Dershowitz and Reingold: Calendrical Calculations, Cambridge University Press (2007)。
- 4) 暦計算研究会（編）：新こよみ便利帳，恒星社厚生閣（1991）。
- 5) 内田正男：日本書紀暦日原典 新装版，雄山閣（1993）。
- 6) 明治五年太政官布告第三百三十七号（改暦ノ布告）（1872）。
- 7) 明治三十一年勅令第九十号（閏年ニ関スル件）（1898）。
- 8) 黒板勝美，國史大系編修會（編）：日本書紀（前）（後），新訂増補 國史大系 第1巻上，下，吉川弘文館（2000）。
- 9) 黒板勝美，國史大系編修會（編）：続日本紀，新訂増補 國史大系 第2巻，吉川弘文館（2001）。
- 10) 国史大辞典編集委員会（編）：国史大辞典，吉川弘文館（1979-1997）。
- 11) 関野 樹：地域研究における時空間情報の活用，情報処理学会研究報告人文科学とコンピュータ（CH），Vol. 2012-CH95, No.10, pp.1-6（2012）。
- 12) GeoNLP プロジェクト，入手先
〈<http://agora.ex.nii.ac.jp/GeoNLP/>〉（参照 2013-10-21）。
- 13) 関野 樹：HuTime 活用のための時間基盤情報。HuTime/Map を使った研究事例と将来展望，2012年3月20日 H-GIS 研究会 報告書，pp.21-26，入手先
〈http://www.hutime.jp/documents/report_20120320.pdf〉（参照 2013-10-21）。